9th International Workshop on Data Mining in Bioinformatics (BIOKDD 2010)

Held in conjunction with SIGKDD conference, Washington, DC, USA, July 2010

Workshop Chairs

Jun Huan
Jake Y. Chen
Mohammed Zaki

BIOKDD'10 International Workshop on Data Mining in Bioinformatics Washington DC, USA

Held in conjunction with
16th ACM SIGKDD Conference on Knowledge Discovery and Data Mining

Jun (Luke) Huan
Department of Electrical Engineering and Computer Science
University of Kansas
Lawrence, KS, 66047-7621
jhuan@ku.edu

Jake Chen
Indiana University School of Informatics
Indiana University–Purdue University Indianapolis
Indianapolis, IN 46202
jakechen@iupui.edu

Mohammed Zaki
Department of Computer Science
Rensselaer Polytechnic Institute
Troy, NY 12180-3590
zaki@cs.rpi.edu

REMARKS

Bioinformatics is the science of managing, mining, and interpreting information from biological data. Various genome projects have contributed to an exponential growth in DNA and protein sequence databases. Advances in high-throughput technology such as microarrays and mass spectrometry have further created the fields of functional genomics and proteomics, in which one can monitor quantitatively the presence of multiple genes, proteins, metabolites, and compounds in a given biological state. The ongoing influx of these data, the presence of biological answers to data observed despite noises, and the gap between data collection and knowledge curation have collectively created exciting opportunities for data mining researchers.

While tremendous progress has been made over the years, many of the fundamental problems in bioinformatics, such as protein structure prediction, gene-environment interaction, and regulatory pathway mapping, are still open. Data mining will play essential roles in understanding these fundamental problems and development of novel therapeutic/diagnostic solutions in post-genome medicine.

The goal of this workshop is to encourage KDD researchers to take on the numerous challenges that Bioinformatics offers. This year, the workshop will feature the theme of "Mining biocomplexity: from molecular systems to health". Different from analyzing single molecules, complex biological systems consist of components that are in themselves complex and interacting with each other. Understanding how the various components work in concert, using modern high-throughput biology and data mining methods, is crucial to the ultimate goal of genome-based economy such as genome medicine and new agricultural and energy solutions. Applying the study of biological systems, health informatics aims to discover novel and useful patterns in large volumes of health care related data and to explore the links between disease physiology and molecular bio-sciences. It integrates data from heterogeneous multimedia sources, especially those from the new high-throughput technologies, and has a wide range of applications in areas of pharmacy, nursing, clinical care, dentistry, public health and medical research. Knowledge discovery tools are expected to play a central role in helping domain

experts to gain deeper insights and formulize better hypotheses from biological, biomedical, pharmaceutical, and health related data.

We encourage papers that propose novel data mining techniques for post-genome bioinformatics studies in areas such as:

- Phylogenetics and comparative Genomics
- DNA microarray data analysis
- Deep sequencing data analysis
- RNAi and microRNA analysis
- Protein/RNA structure prediction
- Sequence and structural motif finding
- Modeling of biological networks and pathways
- Statistical learning methods in bioinformatics
- Computational proteomics
- Computational biomarker discoveries
- Gene-environment, Gene-drug, drug-drug interaction discoveries
- Computer aided drug discoveries
- Biomedical Text mining
- Biological data management techniques
- Semantic webs and ontology-driven biological data integration methods
- Knowledge discovery in electronic medical records
- Privacy and security issues in mining health databases

PROGRAM

The workshop is a half day event in conjunction with the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Washington DC, USA, July 25, 2010. It is accepted into the full conference program after the SIGKDD conference organization committee reviewed the competitive proposal submitted by the workshop co-chairs. To promote this year's program, we established an Internet web site at http://bio.informatics.iupui.edu/biokdd10/.

This year, we accepted 7 full papers, 9 poster papers out of 29 submissions into the workshop program due to the exceptionally high quality of the submissions. All of the papers are accepted as full presentations each with 20 minutes. Each paper was peer reviewed by at least two members of the program committee and papers with declared conflict of interest were reviewed blindly to ensure impartiality. All papers, whether accepted or rejected, were given detailed review forms as a feedback.

Our specially invited keynote talk speaker for this year's program is Dr. Teresa M. Przytycka, from the National Center for Biotechnology Information, NIH. Her talk title is "Towards Uncovering Pathways Connecting Genotype with Phenotype".

WORKSHOP COCHAIRS

- Jun (Luke) Huan, University of Kansas

- Jake Chen, Indiana University School of Informatics, Indiana University–Purdue University Indianapolis
- Mohammed Zaki, Rensselaer Polytechnic Institute

PROGRAM COMMITTEE

Chris Bailey-Kellogg (Dartmouth College), Xue-wen Chen (University of Kansas), Francisco Couto (Universidade de Lisboa), Jane Gao (University of Texas at Arlington) Miao He, (Sun Yat-sen University, China), Tony Hu (Drexel University), Rui Kuang (Univeristy of Minnesota), Vipin Kumar (Univeristy of Minnesota), Doheon Lee (KAIST, Korea), Stefano Lonardi (University of California, Riverside), Jiao Li (NIH), Jie Liang (University of Illinois, Chicago), Jinze Liu (University of Kentuky), Zoran Obradovic (Temple Univeristy), Srinivasan Parthasarathy (Ohio State University), Jianhua Ruan (University of Texas, San Antonio), Saeed Salem (North Dakota State University), Leming Shi (FDA), Ambuj Singh (University of California at Santa Barbara), Min Song (New Jersey's Science & Technology University), Vincent Tseng (National Cheng Kung University, Tainan), James Wang (Penn State), Jason Wang (New Jersey's Science & Technology University), Wei Wang (University of North Carolina), Dong Xu (University of Missouri, Columbia), Jinbo Xu (Toyota Technological Institute), Aidong Zhang (University at Buffalo, The State University of New York), Shuxing Zhang (MD Anderson), Sheng Zhong (University of Illinois)

ACKNOWLEDGEMENT

We would like to thank all the program committee members, contributing authors, invited speaker, and attendees for contributing to the success of the workshop. Special thanks are also extended to the SIGKDD ’10 conference organizing committee for coordinating with us to put together the excellent workshop program on schedule.

PROGRAM SCHEDUKE

8:25 8:30: **Opening Remarks**

Session I: Systems Biology

8:30-8:45 Discovery of Error-tolerant Biclusters from Noisy Gene Expression Data

8:45-9:00 Systematic Construction and Analysis of Co-expression Networks for Identification of Functional Modules and cis-regulatory Elements

9:00- 9:15 A Fast Markov Blankets Method for Epistatic Interactions Detection in Genome-wide Association Studies

9:15 – 9:30 Combining Active Learning and Semi-supervised Learning Techniques to Extract Protein Interaction Sentences

9:30-10:30: **Keynote Speech**

10:30-11:15 **Coffee Break & Poster Session**

Session II: Proteins and Genes

11:15-11:30 Efficient Motif Finding Algorithms for Large-Alphabet Inputs

11:15 – 11:45 Planning Combinatorial Disulfide Cross-links for Protein Fold Determination

11:45 – 12:00 A New Approach for Detecting Bivariate Interactions in High Dimensional Data using Quadratic Discriminant Analysis

12:00 – 1:00: **Panel Discussion**

Discovery of Error-tolerant Biclusters from Noisy Gene Expression Data

Rohit Gupta
Dept of Comp Sc and Engg
Univ of Minnesota, Twin Cities
Minneapolis, MN USA
rohit@cs.umn.edu

Navneet Rao
Dept of Comp Sc and Engg
Univ of Minnesota, Twin Cities
Minneapolis, MN USA
nrao@cs.umn.edu

Vipin Kumar
Dept of Comp Sc and Engg
Univ of Minnesota, Twin Cities
Minneapolis, MN USA
kumar@cs.umn.edu

ABSTRACT

An important analysis performed on microarray gene-expression data is to discover biclusters, which denote groups of genes that are coherently expressed for a subset of conditions. Various biclustering algorithms have been proposed to find different types of biclusters from these real-valued gene-expression data sets. However, these algorithms suffer from several limitations such as inability to explicitly handle errors/noise in the data; difficulty in discovering small bicliusters due to their top-down approach; inability of some of the approaches to find overlapping biclusters, which is crucial as many genes participate in multiple biological processes. Association pattern mining also produce biclusters as their result and can naturally address some of these limitations. However, traditional association mining only finds exact biclusters, which limits its applicability in real-life data sets where the biclusters may be fragmented due to random noise/errors. Moreover, as they only work with binary or boolean attributes, their application on gene-expression data require transforming real-valued attributes to binary attributes, which often results in loss of information. Many past approaches have tried to address the issue of noise and handling real-valued attributes independently but there is no systematic approach that addresses both of these issues together. In this paper, we first propose a novel error-tolerant biclustering model, '*ET-bicluster*', and then propose a bottom-up heuristic-based mining algorithm to sequentially discover error-tolerant biclusters directly from real-valued gene-expression data. The efficacy of our proposed approach is illustrated in the context of two biological problems: discovery of functional modules and discovery of biomarkers. For the first problem, we used two real-valued *S.Cerevisiae* microarray gene-expression data sets and evaluate the biclusters obtained in terms of their functional coherence as evaluated using the GO-based functional enrichment analysis. The statistical significance of the discovered error-tolerant biclusters as estimated by using two randomization tests, reveal that they are indeed biologically meaningful and statistically significant. For the second problem of biomarker discovery, we used four real-valued *Breast Cancer* microarray gene-expression data sets and evaluate the biomarkers obtained using MSigDB gene sets. We compare our results obtained from both the problems, with a recent approach *RAP* and clearly demonstrate the importance of incorporating noise/errors in discovering coherent groups of genes from gene-expression data.

BIOKDD'10, July 25, 2010, Washington DC, USA.

1. INTRODUCTION

Recent technical advancements in DNA microarray technologies have led to the availability of large-scale gene expression data. These data sets can be represented as a matrix G with genes as rows and different experimental conditions as columns, where G_{ij} denotes the expression value of gene i for an experimental condition j. An important research problem of gene-expression analysis is to discover submatrix patterns or biclusters in G. These biclusters are essentially subsets of genes that show coherent values across a subset of experimental conditions. However, coherence among the data values can be defined in various ways. For instance, Madeira et al [22] classify biclusters into the following four different categories based on the definition of coherence: (i) biclusters with constant values, (ii) biclusters with constant rows or columns, (iii) biclusters with coherent values, and (iv) biclusters with coherent evolutions.

Many approaches ([4, 9, 12, 22, 32, 3, 27]) have been proposed to discover biclusters from gene-expression data. Different biclustering algorithms have been designed to discover different types of biclusters. For instance, coclustering [12] and SAMBA [32] find constant value biclusters, Cheng and Church (CC) [9] find constant row biclusters and OPSM [3] find coherent evolutions biclusters. Though there are differences in biclustering algorithms in terms of the type of bicluster they discover, there are some common issues with these algorithms in general. First critical issue with all of these biclustering algorithms is that they are oblivious to noise/errors in the data and require all values in the discovered bicluster to be coherent. This limits the discovery of valid biclusters that are fragmented due to random noise in the data. Second issue with at least some of the biclustering algorithms is their inability to find overlapping biclusters. For instance, coclustering is designed to only look for disjoint biclusters and Cheng and Church's approach, which masks the identified bicluster with random values in each iteration, also finds it hard to discover overlapping biclusters. Third, most of the algorithms are top-down greedy schemes that start with all rows and columns, and then iteratively eliminate them to optimize the objective function. This generally results in large biclusters, which although are useful, do not provide information about the small biological functional classes. Finally, all the biclustering algorithms employ heuristics and are unable to search the space of all possible biclusters exhaustively.

Association pattern mining can naturally address some of the issues faced by biclustering algorithms i.e, finding over-

lapping biclusters and performing an exhaustive search. However, there are two major drawbacks of traditional association mining algorithms. First, these algorithms use a strict definition of support that requires every item (gene) in a pattern (bicluster) to occur in each supporting transaction (experimental condition). This limits the recovery of patterns from noisy real-life data sets as patterns are fragmented due to random noise and other errors in the data. Second, since traditional association mining was originally developed for market basket data, it only works with binary or boolean attributes. Hence it's application to data sets with continuous or categorical attributes requires transforming them into binary attributes, which can be performed by using discretization [30, 28, 13], binarization [2, 11, 10, 23] or by using rank-based transformation [6]. In each case, there is a loss of information and associations obtained does not reflect relationships among the original real-valued attributes, rather reflect relationships among the binned independent values [16].

Efforts have been made to independently address the two issues mentioned above and to the best of our knowledge, no prior work has addressed both the issues together. For example, various methods [34, 33, 21, 20, 29, 7, 5, 8, 25, 26] have been proposed in the last decade to discover approximate frequent patterns (often called error-tolerant itemsets (ETIs)). These algorithms allow patterns in which a specified fraction of the items can be missing - see [14] for a comprehensive review of many of these algorithms. As the conventional support (i.e the number of transactions supporting the pattern) is not anti-monotonic for error-tolerant patterns, most of these algorithms resort to heuristics to discover these patterns. Moreover, all of these algorithms are developed only for binary data.

Another recent approach [24] addressed the second issue and extended association pattern mining for real-valued data. The extended framework is referred to as *RAP* (Range Support Pattern). A novel *range* and *range support* measures were proposed, which ensure that the values of the items constituting a meaningful pattern are coherent and occurs in a substantial fraction of transactions. This approach reduces the loss of information as incurred by discretization- and binarization-based approaches, as well as enables the exhaustive discovery of patterns. One of the major advantages of using an approach such as *RAP*, which adopts a very different pattern discovery algorithm as compared to more traditional biclustering algorithms such as CC or ISA, is the ability to find smaller or completely novel biclusters. Several examples shown in [24] illustrated that *RAP* can discover some biologically relevant smaller biclusters, which are either completely missed by biclustering approaches such as CC or ISA, or are found embedded in larger biclusters. In either case, they are not able to enrich the smaller functional classes as *RAP* biclusters do. Despite these advantages, *RAP* framework does not directly address the issue of noise and errors in the data.

As it has been independently shown that both issues, handling real-valued atributes and noise, are critical and affect the results of the mining process, it is important to address them together. In this paper, we propose a novel extension of association pattern mining to discover error-tolerant biclusters (or patterns) directly from real-valued gene-expression data. We refer to this approach as '*ET-bicluster*' for error-tolerant bicluster. This is a challenging task because the conventional support measure is not anti-monotonic for the error-tolerant patterns and therefore limits the exhaustive search of all possible patterns. Moreover the set of values constituting the pattern in the real-valued data is different than the binary data case. Therefore, instead of using the traditional support measure, we used the *range* and *RangeSupport* measures as proposed in [24] to ensure the coherence of values and for computing the contribution from supporting transactions. *RangeSupport* is anti-monotonic for both dense and error-tolerant patterns, however, *range* is not anti-monotonic for error-tolerant patterns. Due to this, exhaustive search is not guaranteed, however it is important to note that the proposed *ET-bicluster* framework still, by design, finds more number of patterns (biclusters) than it's counterpart *RAP*. Therefore using *range* as a heuristic measure, we describe a bottom-up pattern mining algorithm, which sequentially generates error-tolerant biclusters that satisfy the user-defined constraints, direcly from the real-valued data.

To demonstrate the efficacy of our proposed *ET-bicluster* approach, we compare it's performance with *RAP* in the context of two biological problems: (a) functional module discovery, and (b) biomarker discovery. Since both *ET-bicluster* and *RAP* use same pattern mining framework, comparing them helps to quantify the impact of noise and errors in the data on the discovery of coherent groups of genes in an unbiased way.

For the first problem of functional module discovery, we used real-valued *S.cerevisiae* microarray gene-expression data sets and discovered biclusters using both *ET-bicluster* and *RAP* algorithm. To illustrate the importance of directly incorporating data noise/errors in biclusters, we compared the error-tolerant biclusters and *RAP* biclusters using gene ontology (GO) based biological processes annotation hierarchy [1] as the base biological knowledge. Specifically, for each {bicluster, GO term} pair, we computed a p-value using a hypergeometric distribution, which denotes the random probability of annotating this bicluster with the given GO term. For the second problem of biomarker discovery, we combined four real-valued case-control Breast Cancer gene-expression data sets, and discovered discriminative biclusters (or biomarkers) from the combined data set using both *ET-bicluster* and *RAP*. Again, to illustrate the importance of explicitly incorporating noise/errors in the data, we compared the biomarkers based on their enrichment scores computed using MSiGDB gene sets [31]. MSigDB gene sets are chosen as the base biological knowledge in this case because they include several manually annotated cancer gene sets. The results obtained for both the functional module discovery and biomarker discovery problem clearly demonstrate that error-tolerant biclusters are not only bigger than *RAP* biclusters but are also biologically meaningful. Using randomization tests, we further demonstrated that error-tolerant biclusters are indeed statistically significant and are neither obtained by random chance nor capture random structures in the data. Overall, the results presented for both the biological problems strongly suggest that our proposed *ET-bicluster* approach is a promising method for the analysis of real-valued gene-expression data sets.

Contributions:

- We proposed a novel association pattern mining based approach to discover error-tolerant biclusters from noisy real-valued gene-expression data.
- Our work highlights the importance of tolerating error(s) in the biclusters in order to capture the true underlying structure in the data. This is demonstrated

using two case studies: functional module discovery and biomarker discovery. Using various real-valued gene expression data sets, we illustrated that our proposed algorithm *ET-bicluster* can discover additional and bigger biologically relevant biclusters as compared to *RAP*.

- We used two randomization techniques to compute the empirical p-value of all the discovered error-tolerant biclusters and demonstrated that they are statistically significant and it is highly unlikely to have obtained them by random chance.

Organization: The rest of the paper is organized as follows. In Section 2, we discuss our proposed algorithm *ET-bicluster*. Section 3 details the experimental methodology for evaluating the error-tolerant biclusters and their comparison with *RAP* biclusters, and the results obtained. We present a summary of the findings in section 4 followed by a discussion on limitations and future work in section 5.

2. ERROR-TOLERANT BICLUSTER MODEL FOR REAL-VALUED DATA

As shown in [22], there can be different types of biclusters one can define on a real-valued data based on different measures of coherence among data values. In this paper, we focus on constant row/column biclusters, as they are well suited for the *ET-bicluster* framework and also considered as one of the promising ways to capture functional coherence from the microarray data sets [9]. However, discovering error-tolerant biclusters directly from real-valued data is a challenging task as several issues arise either due to handling of real-valued attributes or due to relaxing the bicluster requirements to incorporate noise/errors in the data. Specifically, following three issues need to be discussed before we present the algorithm.

(a) **Bicluster Composition:** Unlike the case of binary data where collection of $1s$ was defined as a bicluster, in the case of real-valued data, similar values across a set of rows constitute a bicluster. These values can be any values in the set $\mathbb{R}$ and athough similar across rows, they can be different for different rows. The errors in the biclusters defined on real-valued attributes are introduced in a way similar to the binary case. However, like binary case in which all non-error entries are same ($1s$), in real-valued case, imposing such a requirement would be very harsh. Therefore, a measure is needed to check the coherence among the gene-expression values. For this purpose, we use the *range* measure, which checks for each transaction if the relative range of the gene-expression values in a bicluster, given as $(max_{val} - min_{val})/min_{val}$, is within a pre-specified threshold α. Furthermore, the contribution of each supporting transaction is measured as the minimum of the values taken by any of the genes in the bicluster in that transaction. Overall, to measure the strength of the bicluster, we use the *RangeSupport* (*RS*) measure [24], which sums up the contribution of each supporting transaction. This is similar to the *support* measure that is generally used in association pattern mining for binary data, however unlike binary case, each supporting transaction may not contribute equally for *RangeSupport* of a bicluster in real-valued data. The *range* and *RangeSupport* measures in combination capture the requirement that expression values of the genes in a bicluster are coherent for several transactions, and hence can be used to mine interesting biclusters from the real-valued data. Note that although both measures *range* and *RangeSupport* are anti-monotonic for exact biclusters, *range* is not anti-monotonic for error-tolerant biclusters. Due to this reason, *ET-bicluster* does not exhaustively find all error-tolerant biclusters, but it is noteworthy that it still subsume all biclusters found by *RAP* and can even find biclusters that are fragmented due to noise/errors in the data. One the other hand, as *RAP* is oblivious to errors/noise in the data, it either completely miss these fragmented but valid biclusters or find them as separate parts.

	a	b	c	d	e
1	2	2.1	8	2	2
2	2.1	2.2	2.2	2.2	2.2
3	4	4	9	4	4
4	6.5	6.6	6.5	20	6.5
5	8	20	8.8	8	8
6	9	20	9.1	10	9.1
7	3.2	3	8	20	3.2
8	2	2	2	2	2

Figure 1: A sample error-tolerant bicluster

(b) **Positive/Negative Values:** Unlike binary data, real-valued microarray data has both positive and negative values. In this case, it is important to consider the sign of the value to discover meaningful biclusters. Similar to [24], we address this problem by enforcing that a transaction can only be termed as the supporting transaction of a bicluster if for this transaction, the expression values of all the genes in the bicluster are of the same sign. This also help make biological interpretability easier as the sign enforcement would entail finding only those biclusters in which all the genes are either up-regulated or down-regulated for a given experimental condition. However note that the same genes can be up-regulated for one experimental condition and down-regulated for another.

(c) **Error/Non-error Values:** In binary case, 1 is always a non-error value and 0 an error value. This notion is no more valid for the real-valued data case. For example, consider an error-tolerant bicluster shown in figure 1 with 5 genes (a, b, c, d, e) and 8 experimental conditions ($1 \ldots 8$). For the 1st condition, 8 is an error value, for the 3rd condition 9 is an error value, and for the 5th condition, 20 is an error value. Similarly, non-error values can change for each transaction. Thus, it becomes important to keep track of error and non-error values while mining for biclusters in the real-valued data.

Now, with the understanding of specific challenges and potential ways to address them, we now give the formal definition of error-tolerant biclusters for a real-valued data.

2.1 Definition of Error-tolerant Biclusters

Intuitively, a bicluster B is said to be an error-tolerant bicluster if the following two general conditions are satisfied:

- *RangeSupport* of bicluster B should be more than the user-defined threshold, *RS*.
- All supporting transactions of bicluster B should have mostly non-error values i.e. values should be generally coherent (governed by a user-defined parameter ϵ for maximum number of permissible errors).

Definition 1. Let D be a real-valued gene-expression data, *RS* be the *RangeSupport* threshold, E be a function that

takes a set of real values as input and returns the number of errors in them using *range* criteria, and let error threshold be $\epsilon \in (0, 1]$. A bicluster B (with genes G) is an error-tolerant bicluster *ET-bicluster*(ϵ) in the real-valued attribute domain, if there exists a set of transactions $T \in D$ such that the following two conditions hold:

$$Range\ Support(B) \geq RS \quad (1)$$

$$\forall t \in T, E(D_{t,G}) \leq \epsilon \cdot |G| \quad (2)$$

Thus according to the definition, fraction of errors in each supporting transaction of the bicluster should not exceed ϵ.

2.2 Algorithm to Discover Error-tolerant Biclusters from Real-valued Data

Starting with singletons, the *ET-bicluster* algorithm sequentially generates (k+1)-level biclusters from k-level biclusters. At k = 1, genes that satisfy the *RangeSupport* (computed as the summation of absolute values for all transactions) criterion are valid singletons. Generally speaking, any (k+1)-level bicluster is a valid bicluster if it satisfies the *RangeSupport* criterion and each supporting transaction of the bicluster has at most ϵ fraction of errors.

ET-bicluster algorithm generates (k+1)-level biclusters from k-level biclusters by one of the two steps: error extension or non-error extension. Specifically, if $\lfloor (k+1) \cdot \epsilon \rfloor = \lfloor k \cdot \epsilon \rfloor$, it's a non-error extension step (no more errors values are permitted) or else it will be a error-extension step (one additional error value is permitted). We used two lemmas proved in [20] to efficiently perform these extension steps. In non-error extension step, for each (k+1)-level bicluster, *range* criteria is only checked for the intersection of supporting transactions of all its k-level biclusters. On the other hand, in the error-extension step, *range* criteria is checked for the union of supporting transactions of all its k-level biclusters.

Checking the range criterion to ensure the coherence of values depends on the number of permissible errors at a particular bicluster-level ($k \cdot \epsilon$). For instance, if the permissible number of errors is 1, then *range* criterion for a given transaction is computed as follows. First, for each transaction, all the expression values in a bicluster are sorted and then the range criterion is checked in usual manner by either discarding the minimum value or the maximum value. If the *range* criterion is satisfied in any of the two cases, transaction is classified as the supporting transaction for that bicluster. If for instance, number of permissible errors are 2 at any bicluster-level, we check the *range* criterion for three cases: discarding the 2 minimum values; discarding the 2 maximum values; or discarding 1 minimum value and 1 maximum value. Again, if any of the case satisfies the *range* criterion, transaction is classified as a supporting transaction. Similarly, we exhaustively make all cases when number of permissible errors are more than 2. However, note that with $\epsilon = 0.25$ (value considered in this paper) and itemset size even as big as 12, we only need to make these cases for 3 permissible errors.

2.3 An Example

Considering a sample real-valued data with 5 genes (a, b, c, d, and e) and 8 experimental conditions (1 through 8) as shown in figure 1, below we demonstrate the steps of *ET-bicluster* algorithm. Input parameters: Range Support threshold = 5; $\alpha = 0.5; \epsilon = 0.25$

Step 1: $k = 1$. As range support for each gene is greater than 5, all the genes are returned as valid singletons.

Step 2: $k = 2$. Since $\lfloor k * \epsilon \rfloor = \lfloor k - 1 \rfloor * \epsilon$, this is a non-error extension step. Consider for example bicluster *ab*, for $\alpha = 0.5$, it's supporting transactions are {1,2,3,4,7,8}. To illustrate, while transaction 1 satisfies the range criteria (i.e. $2.1 - 2 \leq 0.5 * 2$) and hence is valid, transaction 5 is not valid since $20 - 8 > 0.5 * 8$. Now, *RangeSupport* of bicluster *ab* is given as the sum of the contributions from each supporting transaction i.e. $RS(ab) = 2 + 2.1 + 4 + 6.5 + 3 + 2 = 19.6$. Since, $RS(ab) > 5$, *ab* is a valid bicluster. Similarly, biclusters *ac*, *ad*, *ae*, *bc*, *bd*, *be*, *cd*, *ce*, *de* are all valid biclusters.

Step 3: $k = 3$. Again since $\lfloor k * \epsilon \rfloor = \lfloor k - 1 \rfloor * \epsilon$, this is a non-error extension step. Consider for example, bicluster *abc*, *range* criterion is checked for intersection of supporting transactions of biclusters *ab*, *bc* and *ac* and hence supporting transactions are identified as {2,4,8}. Now, since $RS(abc) = 10.6$, which is greater than the thereshold 5, *abc* is a valid bicluster. Similarly, *abd*, *abe*, *bce*, *bde* and *cde* are all valid biclusters.

Step 4: $k = 4$. In this case, since $\lfloor k * \epsilon \rfloor \neq \lfloor k - 1 \rfloor * \epsilon$, this is an error extension step. The number of permissible errors at this level is $k * \epsilon_r = 4 * 0.25 = 1$. Consider for example, bicluster *abcd*, *range* criterion is checked for the union of supporting transactions of all its level-3 biclusters subsets. Hence, we get {1,2,3,4,5,6,8} as the set of supporting transactions. For illustration, take an example of transaction 1. As only one error value is permitted, range criterion is checked as follows: $(((2^{nd} max - min)/min) = (2.1 - 2)/2 = 0.05 < \alpha(0.5))$. Therefore, this is a supporting transaction. On the other hand, transaction 7, even after discarding one error value does not satisfy the range criterion for bicluster *abcd*. Also $RS(abcd) = 33.6$, hence *abcd* is a valid bicluster. Similarly, *abce* is also a valid bicluster.

Step 5: $k = 5$. Since, $\lfloor k * \epsilon \rfloor = \lfloor k - 1 \rfloor * \epsilon$, this is a non-error extension step. A bicluster *abcde* will be generated with set of supporting transactions as {1,2,3,4,5,6,8}. Now since $RS(abcde) = 33.6$, abcde is a valid bicluster.

It is important to note that since *RAP* does not explicitly handle errors/noise in the data, it cannot discover the bicluster *abcde*, which is fragmented due to errors.

3. EXPERIMENTAL RESULTS AND DISCUSSION

We implemented our proposed association pattern mining approach '*ET-bicluster*' in C++. In this paper, we only compare our proposed approach with *RAP*, as *RAP* has already been shown to outperform biclustering approaches such as ISA and Cheng and Church, especially for finding small biclusters. Also, as mentioned in [24], transformation of data from real-valued attributes to binary attributes leads to loss of distinction between various types of biclusters (or patterns). Therefore, as the focus of this study is to discover constant row biclusters, binarization of real-valued gene-expression data is not meaningful. For this reason, we only show results on real-valued data sets. Further, in order to compare the performance of '*ET-bicluster*' and *RAP* in discovering coherent groups of genes, we considered two biological problems: discovery of functional modules (finding coherent gene groups) and discovery of biomarkers (finding coherent gene groups that are discriminative of the two classes of patients: cases and controls).

Selecting Top Biclusters As association mining based approach generally produces a large number of biclusters that often have substantial overlap with each other, this redundancy in biclusters may bias the evaluation. Hence, we used a commonly adopted selection methodology similar to the

one proposed by [27] to select upto 500 top biclusters. However, because error-tolerant biclusters generally have a large set of supporting experimental conditions, even biclusters with high overlap in gene dimension may get selected in the top 500 biclusters. To avoid this situation, we computed the size of a bicluster by the number of genes ($|genes|$) in it, not by $|genes| \times |conditions|$ in it. Therefore, starting with the largest bicluster (only in terms of the number of genes in it), we greedily select upto 500 biclusters such that the overlap among any of the selected biclusters is not more than 25%. In case of a tie between the size of biclusters, bicluster with lower Mean Square Error (MSE) value [9] is selected. Please note that MSE of a bicluster is computed by discarding the error values in it, since *ET-bicluster* is meant to look for error-tolerant patterns.

3.1 Case Study 1 - Discovery of Functional Modules

We used the following two real-valued *S.cerevisiae* microarray gene-expression data sets for the discovery of functional modules:

- Hughes et al's data set [18]: This data set contains a compendium of expression profiles corresponding to 300 diverse mutations and chemical treatments in S. cerevisiae and was compiled to study the functions of yeast genes on a large scale. The overall dimensions of this data set are 6316 genes x 300 conditions, with values (log_{10} ratio of expression values observed for experimental condition and control condition) in the range [-2,2].
- Mega Yeast data set [19]: This data set contains 501 yeast microarray experiments, including stress responses, cell cycle, sporulation, etc. The overall dimensions of this data set are 6447 genes x 501 conditions, with values in the range [-12,12].

Functional Enrichment Analysis Since the discovered biclusters represent groups of genes that are expected to co-express with each other, we evaluated all the biclusters discovered in terms of their functional coherence using the biological processes annotation hierarchy of Gene Ontology [1]. A p-value using a hypergeometric probability distribution is computed for each combination of bicluster and biological process GO term to determine if the discovered biclusters are statistically significant. The p-value computed for a pair of bicluster (denoted by b) and GO term (denoted by t) denotes the random probability of annotating a bicluster of size same as b with the same GO term t.
To compare error-tolerant biclusters and *RAP* biclusters in an unbiased fashion, we used the same 2652 biological processes GO terms (or classes), all of which contain at least 1 and at most 500 genes from *S.cerevisiae*. Furthermore, as only 4684 genes are annotated with either one or more of these 2652 classes, we restricted our analysis to a subset of data sets comprising of 4684 *genes* $\times$ 501 *conditions* and 4684 *genes* $\times$ 300 *conditions* for mega yeast and Hughes's et al's gene-expression data sets respectively.

3.1.1 Quantitative Analysis of Biclusters

Table 1 provides a general overview of all the biclusters obtained by *ET-bicluster* and *RAP* algorithm on mega yeast and Hughes et al's real-valued gene-expression data sets using various parameter settings. Parameter controlling error-tolerance (ϵ) was set to 0.25 in all the runs for *ET-bicluster*. It is important to note that number of error-tolerant biclusters is substantially larger than the number of *RAP* biclusters. Therefore, for a specific $range(\alpha)$ value and user-defined *RangeSupport* threshold, if *ET-bicluster* algorithm was not able to finish in a reasonable amount of time and memory with $\epsilon = 0.25$, we first obtain exact biclusters (no error-tolerance) by setting ϵ to 0 and then increase the *RangeSupport* to obtain error-tolerant biclusters by setting ϵ to 0.25. The final resulting set of biclusters is obtained by merging these exact and error-tolerant biclusters. Following are some of the general observations:

Number of Biclusters: It can be clearly seen from table 1 that introducing an error-tolerance of 25% substantially increased the total number of biclusters. For example, number of total error-tolerant biclusters obtained on mega yeast data is approximately 5-times (for $\alpha = 0.5$) and 3-times (for $\alpha = 0.3$) the number of *RAP* biclusters for corresponding α values. Similarly, for Hughes et al's data set, number of error-tolerant biclusters is approximately 3-times the number of *RAP* biclusters for both the α values considered ($\alpha = 0.8$ and $\alpha = 0.5$).

Size of Biclusters: Another important observation one can make from the results shown in table 1 is that the size of error-tolerant biclusters is more than *RAP* biclusters. This is expected as *RAP* can only find exact biclusters (with no error-tolerance) and hence valid biclusters that are fragmented due to random noise and errors in the data, are either found as separate biclusters or completely missed. On the other hand, because *ET-bicluster* algorithm explicitly handles noise and errors in the data, it can potentially find larger biclusters by stitching together the fragmented parts or can even find new biclusters that were missed by *RAP*. This might have a significant impact on the functional enrichment analysis as *ET-bicluster* algorithm can potentially discover biclusters that have higher overlap with the considered GO biological processes classes. We discuss this further in the next section.

Coverage of Genes and Relationships Among Them: As can be noted from table 1, the number of genes covered by *ET-bicluster* and *RAP* algorithm is same at least if we consider all biclusters. This is because the starting set of genes ('singletons') are same for both the algorithms. In fact, if the error-tolerance, ϵ is 0.25 for example, then singletons, pairs (level-2 bicluster) and even triplets (level-3 bicluster) will be identical for *ET-bicluster* and *RAP*. However note that the number of level-4 biclusters generated by *ET-bicluster* is more than those generated by *RAP*. This is due to the fact that *ET-bicluster* algorithm, owing to its relaxed error-tolerance criterion, can generate more combinations of genes than *RAP*. Therefore in other words, even if the total genes covered by both the algorithms are same, *ET-bicluster* algorithm can find more relationships among them.

As mentioned above and shown in table 1, since *ET-bicluster* algorithm, as compared *RAP*, can potentially find newer and larger biclusters and hence more relationships among genes, an important question to address is: whether these larger and new biclusters are biologically meaningful? One promising way to answer this question is through functional enrichment analysis and below we discuss these results.

3.1.2 Functional Enrichment using GO Biological Processes

As mentioned earlier, a p-value for each of the (bicluster, GO term) pair is computed for the selected top 500 biclusters using the 2652 biological processes GO terms considered in this study. To demonstrate how well error-tolerant and *RAP* biclusters are enriched by GO terms, we show the dis-

Run ID	Parameter Settings	# total biclusters	# genes covered[1]	# top biclusters	# genes covered[2]	Size distribution[2] # of genes: # of biclusters	Time taken (seconds)
Error-tolerant Biclusters on Mega Yeast Data Set							
$ET\text{-}bicluster_{M1}$	$\alpha = 0.5$, $\epsilon = 0\ for\ RS \in [120\ 150)$, $\epsilon = 0.25\ for\ RS \geq 150$	153,960	361	500	295	2:128, 3:235, 4:8, 5:76, 6:39, 7:7, 8:2, 9:1, 10:2, 11:1, 13:1	10,560
$ET\text{-}bicluster_{M2}$	$\alpha = 0.3$, $\epsilon = 0\ for\ RS \in [60\ 90)$, $\epsilon = 0.25\ for\ RS \geq 90$	271,101	792	500	233	3:203, 4:28, 5:177, 6:80, 7:5, 8:3, 9:3, 10:1	33,000
RAP Biclusters on Mega Yeast Data Set							
RAP_{M1}	$\alpha = 0.5,\ RS \geq 120$	33,330	361	500	247	2:68, 3:379, 4:33, 5:16, 6:4	642
RAP_{M2}	$\alpha = 0.3,\ RS \geq 60$	94,806	792	500	241	3:384, 4:68, 5:43, 6:5	7,580
Error-tolerant Biclusters on Hughes et. al's Data Set							
$ET\text{-}bicluster_{H1}$	$\alpha = 0.8$, $\epsilon = 0\ for\ RS \in [10\ 15)$, $\epsilon = 0.25\ for\ RS \geq 15$	150,372	506	496	437	2:210, 3:187, 4:12, 5:66, 6:14, 7:3, 8:1, 10:1, 11:1, 13:1	8,360
$ET\text{-}bicluster_{H2}$	$\alpha = 0.5$, $\epsilon = 0\ for\ RS \in [6\ 10)$, $\epsilon = 0.25\ for\ RS \geq 10$	234,761	1135	500	443	2:115, 3:258, 4:22, 5:69, 6:24, 7:6, 8:1, 9:2, 11:1, 13:1, 14:1	21,745
RAP Biclusters on Hughes et. al's Data Set							
RAP_{H1}	$\alpha = 0.8,\ RS \geq 10$	56,009	506	495	438	2:212, 3:207, 4:25, 5:40, 6:5, 7:3, 8:2, 11:1	2,835
RAP_{H2}	$\alpha = 0.5,\ RS \geq 6$	80,335	1135	500	405	2:96, 3:303, 4:18, 5:75, 6:2, 7:2, 8:3, 12:1	1,505

Table 1: Statistics of biclusters obtained using '*ET-bicluster*' and '*RAP*' from Mega Yeast and Hughes et al's microarray gene-expression data sets. ([1]all biclusters, [2]top biclusters)

tribution of $-log_{10}(pvalue)$ and size of the biclusters. While figures 2 (a) and (b) show this distribution for mega yeast data set corresponding to two α values of 0.5 and 0.3, figures 2 (c) and (d) show this distribution for Hughes et al's data set corresponding to α values of 0.8 and 0.5 considered in this study. It can be seen from these plots that *ET-bicluster* algorithm not only generates bigger biclusters (in terms of number of genes in them) as discussed before, but also these biclusters have high $-log_{10}(pvalue)$ (or low p-value), which means it is highly unlikely to have discovered these error-tolerant biclusters by random chance. Consider mega yeast data for example, while *ET-bicluster* algorithm can discover biclusters of sizes as big as 13 (for $\alpha = 0.5$) and 10 (for $\alpha = 0.3$), *RAP* algorithm can only discover biclusters of maximum size 6. Moreover, enrichment scores of these larger error-tolerant biclusters (computed using the minimum p-value estimated for these biclusters for 2652 classes) are reasonably high. Therefore, even if the number of unique genes covered and number of enriched GO terms are comparable for *ET-bicluster* and *RAP* algorithm, the degree to which error-tolerant biclusters enrich the GO terms is certainly higher. In other words, *ET-bicluster* algorithm can find more relationships among the genes covered and as shown by functional enrichment analysis, these relationships indeed seem to be biologically relevant and not spurious.

Further, considering various p-value thresholds (from loose – 5×10^{-2} to strict – 1×10^{-5}), we collected two more statistics. First, the fraction of biclusters that are enriched by at least one GO term, and second, the fraction of GO terms that enriched at least one bicluster. To illustrate the efficacy of *ET-bicluster* in capturing the functional coherence among genes, and comparing it with *RAP*, the above two statistics are collected for all the runs shown in table 1. For instance, if we compare these statistics for mega yeast data, while 83% of the top 500 error-tolerant biclusters (corresponding to Run ID $ET\text{-}bicluster_{M2}$) were enriched, only 76% of the top 500 RAP biclusters (corresponding to Run ID RAP_{M2}) were enriched by at least one GO term at a reasonable p-value threshold of 1×10^{-3}, a gain of 7%. At even more strict p-value threshold of 1×10^{-5}, the gain is 11%. Similarly, for Hughes et al's data set, though the gain is not significant, biclusters obtained from *ET-bicluster* still outperform those obtained by *RAP* in terms of the fraction of biclusters enriched. As far as the second statistics is concerned i.e. the number of GO terms that enriched at least one bicluster, performance of *ET-bicluster* and *RAP* is comparable, however, as shown in $-log_{10}(pvalue)$ vs. $size$ distribution plots, enrichment scores for error-tolerant biclusters are generally higher than *RAP* biclusters.

3.1.3 Statistical Significance of Error-tolerant Biclusters Using Randomization Tests

Motivated by the discussion of randomizaton tests and their importance in validating the results from any data mining approach [17], we further estimate the statistical significance of the error-tolerant biclusters using a data centric randomization approach. More specifically, an empirical p-value is computed for all the error-tolerant biclusters using the two randomization tests.

In the first randomization test, conserving the size of the top 500 error-tolerant biclusters, we generated 1000 random sets of 500 biclusters each and evaluated them by the same functional enrichment analysis using GO biological processes. So effectively, for each actual error-tolerant bicluster, we generated 1000 random biclusters of the same size (in terms of number of genes). The empirical p-value for each actual error-tolerant bicluster is then computed as the fraction of random biclusters (out of total 1000) whose enrichment score ($-log_{10}(pvalue)$) exceeds the enrichment score of the actual error-tolerant bicluster. For instance, if for a error-tolerant bicluster, only 1 out of 1000 random biclusters has higher enrichment score than it's actual value, empirical p-value of this error-tolerant bicluster is given as '1 *in* 1000' or 10^{-3}. Figure 3 shows the ($-log_{10}(empirical\ p\text{-}values)$) for all the error-tolerant biclusters that were shown in figure 2. To plot these values at the same scale, an empirical p-value of '0 *in* 1000' is set to 10^{-5} to ensure that they stand out from the rest. Therefore, all the biclusters showing

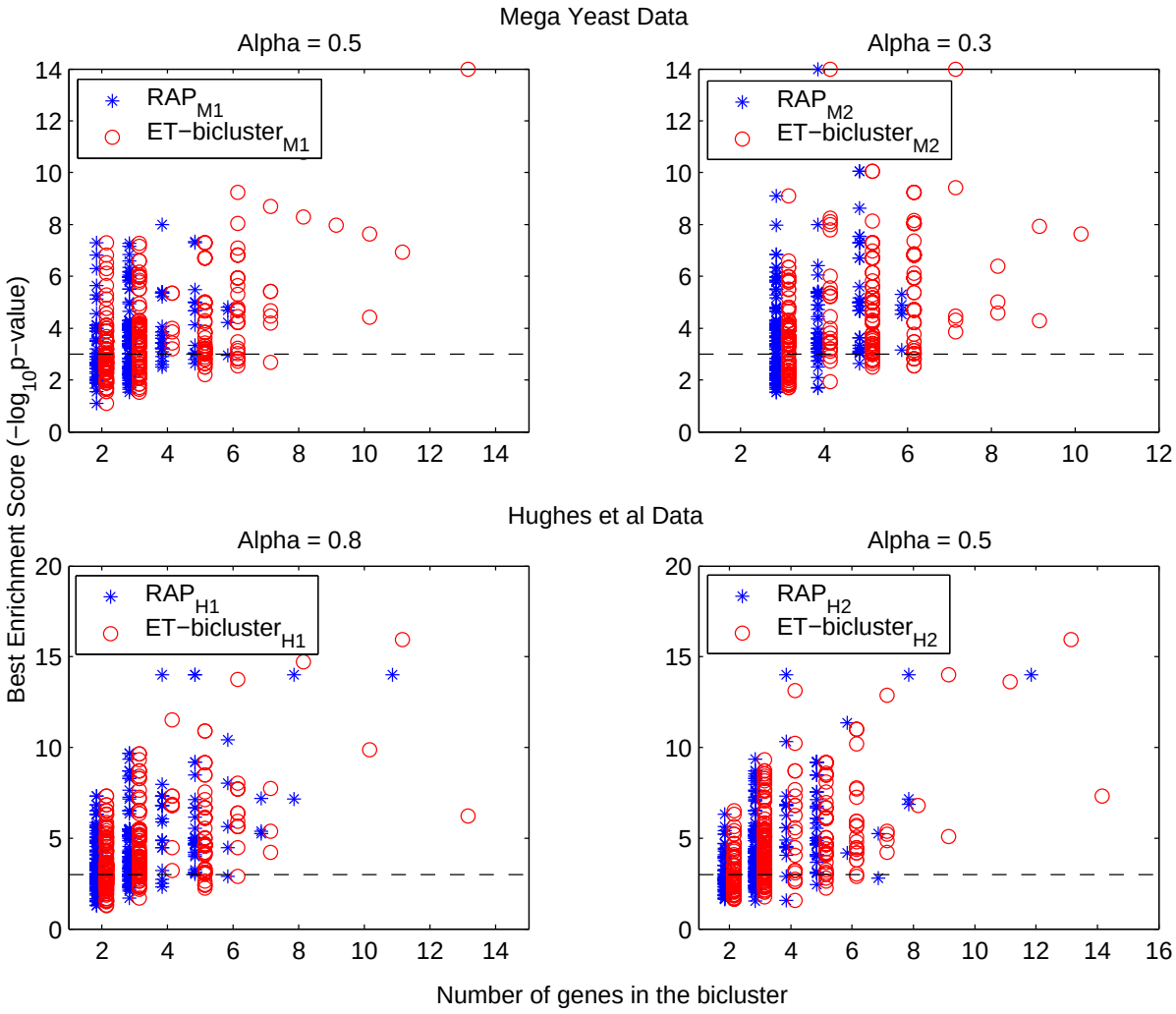


Figure 2: Bicluster Size vs Enrichment Scores (Computed using Biological Processes) for Mega Yeast and Hughes et al's data sets

($-log_{10}(empirical\ p\text{-}values)$) as 5 in figure 3 correspond to empirical p-value of '0 *in* 1000'. It can be clearly seen from figure 3 that error-tolerant biclusters that were assigned high enrichment scores from the GO-based evaluation also have high ($-log_{10}(empirical\ p\text{-}values)$). This means higher the enrichment score of a bicluster, less likely it is to obtain this by random chance, which further illustrates that the bigger error-tolerant biclusters discovered by only *ET-bicluster* algorithm but not by *RAP* algorithm are indeed statistically significant.

We also showed in table 2, the summary statistics of the evaluation results on 1000 randomly generated sets of biclusters. More specifically, for a given p-value threshold, we first compute for each of the 1000 random runs, the fraction of biclusters that have a p-value better than the given threshold and then we report how many times it exceeds the same fraction computed for the actual set of biclusters. It can be clearly seen from the table that specially for a stricter p-value threshold, none of the randomly generated biclusters are better than the actual biclusters. For instance, while 83% of the actual 500 biclusters on mega yeast data ('Run ID: $ET\text{-}bicluster_{M2}$') had $-log_{10}(pvalue)$ higher than 3, this percentage for 1000 random runs was substantially lower with mean of around 36% and a maximum of only 42%. The results were very similar for Hughes et al's data set. Both these set of results further confirms the statistical significance of biclusters obtained from *ET-bicluster* algorithm.

In the second randomization test, we randomized the data itself by shuffling the data values among the conditions for each gene. By doing this, we conserved the distribution of each gene profile but broke the correlation among them. We ran our proposed *ET-bicluster* algorithm on randomized mega yeast data set for example, and obtained only 42 biclusters, all of which were pairs. In contrast, application of *ET-bicluster* algorithm on actual non-randomized mega yeast data generated many more biclusters and of size as big as 10.

Both of the above randomization tests suggest that the error-tolerant biclusters obtained from the real-valued gene-expression data sets were indeed biologically meaningful and are neither obtained by random chance nor capture any random structure in the data.

3.2 Case Study 2 - Discovery of Biomarkers

We used four real-valued *Breast Cancer* gene-expression data sets, all of which were taken from Affymetrix platform HGU133A and normalized using RMA-normalization ap-

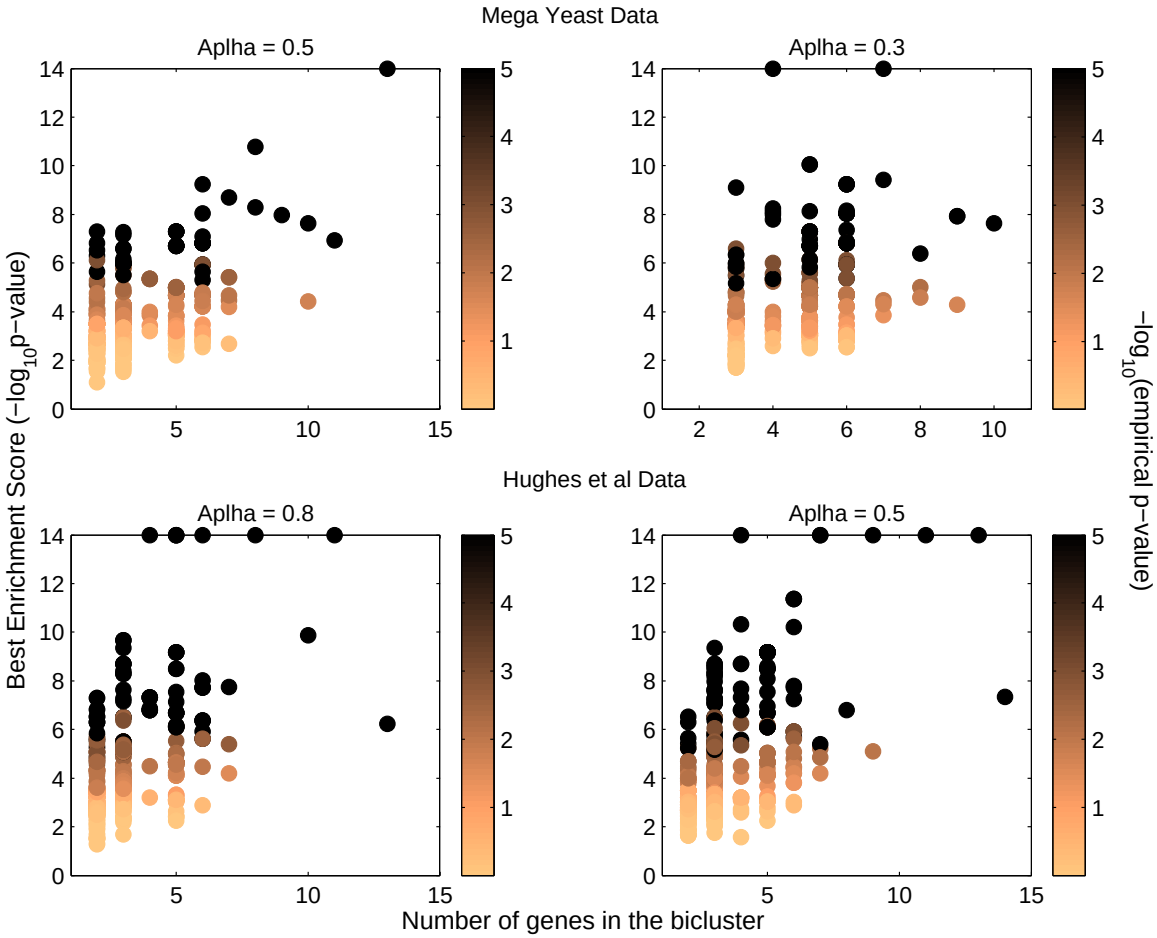


Figure 3: Biological and Empirical p-value (using 1000 random runs) of the Biclusters Obtained from *ET-bicluster* Algorithm [figure best viewed in color].

Run ID	# of random runs out of 1000 in which fraction of biclusters enriched exceeds the fraction for the true run				
	pval ≤ 0.05	pval ≤ 0.01	pval ≤ 0.005	pval ≤ 0.001	pval ≤ 0.00001
$ET\text{-}bicluster_{M1}$	660	33	0	0	0
$ET\text{-}bicluster_{M2}$	660	76	4	0	0
$ET\text{-}bicluster_{H1}$	797	0	0	0	0
$ET\text{-}bicluster_{H2}$	886	0	0	0	0

Table 2: Statistical Significance of Biclusters Obtained from *ET-bicluster*

proach. Please note that these gene-expression data sets are different than those considered for functional module discovery problem, in the sense that experimental conditions are replaced by two groups of patients. All the four breast cancer data sets were downloaded from GEO website: Desmedt (GSE7390), Loi (GSE6532), Miller (GSE3494) and Pawitan (GSE1456). The patients in the four data sets are classified as cases and controls based on their metastasis state. The patients who developed metastasis within 5 years of prognosis were considered as metastasis cases. The patients who were free of metastasis longer than 8 years of survival and follow-up time were considered as controls. The case-control ratio for Desmedt, Loi, Miller and Pawitan data set was 35:136, 51:112, 37:150 and 35:35 respectively. To increase the samle size, we combined these four data sets and used it for the discovery of biomarkers. This combined data set comprises of 8,920 genes and a case-control ratio of 158:433.

We discovered biclusters on combined Breast Cancer gene-expression data set using *ET-bicluster* with parameters, $\alpha = 0.5$, $RS = 80$, and $\epsilon = 0.25$.

Selecting disriminative biclusters First we select top biclusters using the approach defined earlier and then amongst the top biclusters, only those are selected as biomarkers that are discriminative of the two groups of patients, cases and controls. To measure the discriminative power, we used two measures, odds ratio and p-value. While odds ratio quantifies how different are cases and controls for a specific bicluster, p-value quantifies the significance of the difference reflected by odds ratio. Only those biclusters are selected that have a p-value of less than 0.05 and odds ratio of more than 2.0 (biclusters more represented in cases) or less than 0.5 (biclusters more represented in controls).

Functional Enrichment Analysis We evaluated all the identified biomarkers in terms of their enrichment scores using the MSigDB gene sets [31]. A p-value using a hypergeometric probability distribution, which denotes the random probability of annotating a biomarker with the gene set considered, is computed for all pair combinations of biomarkers and 5452 gene sets from MSigDB database. Enrichment score of each biomarker is then computed as $-log_{10}(p\text{-}value_{min})$ and used as a metric to compare the biomarkers obtained using *ET-bicluster* and *RAP*.

3.2.1 Enrichment Analysis Using MSigDB Gene Sets

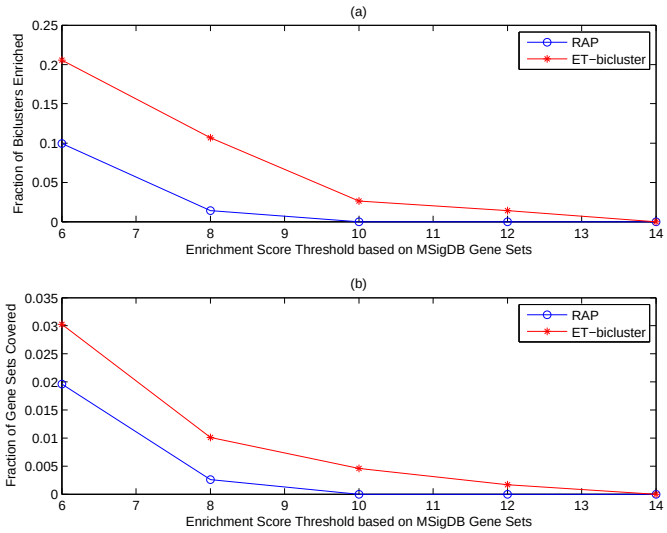


Figure 4: (a) Fraction of Biomarkers Enriched by at least One Gene Set, (b) Fraction of Gene Sets Enriched by at least One Biomarker

Considering various p-value thresholds (from 10^{-6} to 10^{-14}), figure 4 shows two statistics: (a) fraction of biomarkers enriched by at least one gene set, and (b) fraction of gene sets that enriched at least one biomarker. These two statistics are collected both for biomarkers obtained from *ET-bicluster* and *RAP* algorithm at various p-value thresholds. As mentioned earlier, biomarkers obtained by *ET-bicluster* are not only bigger than those obtained by *RAP*, as illustrated in figure 4(a), even a higher fraction of them is enriched by at least one gene set. Consider for instance, a strict p-value threshold of 10^{-8} (corresponding to $-log_{10}(p\text{-}value)$ of 8 as shown on the x-axis), while 10.5% of the error-tolerant biomarkers are enriched, only 1.5% of the RAP biomarkers are enriched.

Now refer to figure 4(b), gene sets covered by *ET-bicluster* biomarkers are more than those covered by *RAP* biomarkers. The fraction of gene sets covered by biomarkers obtained from both the algorithms seems very low but this is expected because first a large number of gene sets are considered for the analysis and second, these biomarkers are only reflective of breast cancer metastasis. An important point to note however is that even a small change in fraction of gene sets covered would mean covering substantially large number of gene sets. For instance, consider a p-value threshold of 10^{-6} (corresponding to $-log_{10}(p\text{-}value)$ of 6 as shown on the x-axis), *ET-bicluster* and *RAP* biomarkers cover 3.03% (165 gene sets) and 1.96% (107 gene sets) respectively. These numbers for a even stricter p-value threshold of 10^{-8} are 1.01% (55 gene sets) 0.26% (14 gene sets) respectively.

It is clear that the biomarkers obtained from *ET-bicluster* algorithm are indeed biologically meaningful and because *RAP* algorithm does not explicitly handle noise in the data, it either completely miss some of these biologically relevant biomarkers or find fragmented parts of these, which eventually affect their enrichment score.

4. CONCLUSIONS

We proposed a novel error-tolerant biclustering model and presented an heuristic-based algorithm '*ET-bicluster*' to sequentially generate error-tolerant biclusters from real-valued gene-expression data in a bottom-up fashion.
We presented two biological case studies, functional module discovery and biomarker discovery, to demonstrate the importance of incorporating noise and errors in the data for discovering coherent groups of genes. In both the case studies, we found that the biclusters discovered using our proposed *ET-bicluster* algorithm are not only bigger than those obtained by *RAP* algorithm, they were also assigned a higher functional enrichment score using the biological processes GO terms and MSigDB gene sets. These results suggest that the discovered error-tolerant biclusters, not only capture the functional coherence among the genes, it is unlikely to have obtained them by random chance. We further demonstrated that the statistical significance of error-tolerant biclusters is high by computing their empirical p-value using the two randomization tests. The results from both randomization tests (one randomly selects the biclusters and other randomizes the input data itself) suggest the robustness of our proposed approach and clearly illustrate that discovered biclusters were indeed biologically and statistically meaningful and neither obtained by random chance nor capturing any random structure in the data.

5. LIMITATIONS AND FUTURE WORK

The work presented in this study has several limitations and can be extended in various ways. Below we discuss some of the limitations of the *ET-bicluster* algorithm and possible ideas to address them.

- Since the *range* criterion that is used to check the coherence of expression values is not anti-monotonic, the proposed *ET-bicluster* approach does not exhaustively search for all error-tolerant biclusters. Therefore, a promising idea is to define a new anti-monotonic measure that measures the coherence among the expression values and enable exhaustive search for error-tolerant biclusters.
- The current implementation of *ET-bicluster* algorithm only impose error-tolerance constraints in the bicluster row. This means that it is possible for a gene in a discovered bicluster to have all error values. To avoid this situation, one can use additional column constraint and find a subset of supporting transactions for which each column in the pattern has no more than some user-defined fraction of errors. For binary data case, this kind of additional column constraint has been used in [20], however, a heuristic-based approach is used to check the column constraint. One potential way to address this is to develop a pattern mining algorithm that checks both the row and column error-tolerance constraints, and exhaustively search for all the error-tolerant biclusters.
- As the error-tolerant pattern mining is computationally more challenging, more efficient data structures and memory management techniques can be used. This would enhance the scalability of the algorithm and enable the discovery of biclusters on a wider range of parameter settings.

We only presented comparison of *ET-bicluster* and *RAP* since comparison with other biclustering approaches such as *CC* and *ISA* is not well suited for quantifying the affect of noise/errors. Moreover *CC* and *ISA* approaches generally finds larger biclusters and follow a different approach based on optimizing an objective function. Nevertheless, it will still be interesting in future to compare *ET-bicluster* with *CC* and *ISA* for potential complementarity among them.

It is also important to note that gene-expression data provides useful but limited view of the genome and therefore biclusters obtained from gene-expression data alone may not elucidate the complete underlying biological mechanism. Hence another promising research direction is to integrate multiple biological data sources for complex problems like discovery of functional modules or biomarkers. For example, protein-protein interaction data can be used as a prior knowledge to guide the discovery of biclusters from the gene-expression data. The biclusters identified in this way are potentially more reliable and biologically plausible than those obtained from individual data sources. We are currently developing error-tolerant pattern mining based approaches for integrated analysis of gene-expression and protein-protein interaction data. One such application for discovering sub-network based biomarkers for Breast cancer metastasis has been shown in [15], however, these approaches are primitive at this stage and further work is needed in this area.

6. ACKNOWLEDGEMENTS

The authors would like to thank the anonymous reviewers for their constructive and thoughtful comments. This work was supported by NSF grants IIS-0916439, CRI-0551551 and a University of Minnesota Rochester Biomedical Informatics and Computational Biology (BICB) Program Traineeship Award (Rohit Gupta). Access to computing facilities was provided by the Minnesota Supercomputing Institute.

7. REFERENCES

[1] M. Ashburner, C.A. Ball, J.A. Blake, D. Botstein, H. Butler, J.M. Cherry, A.P. Davis, K. Dolinski, S.S. Dwight, J.T. Eppig, et al. Gene Ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29, 2000.

[2] C. Becquet, S. Blachon, B. Jeudy, J.F. Boulicaut, and O. Gandrillon. Strong-association-rule mining for large-scale gene-expression data analysis: a case study on human SAGE data. *Genome Biology*, 3(12), 2002.

[3] A. Ben-Dor, B. Chor, R. Karp, and Z. Yakhini. Discovering local structure in gene expression data: the order-preserving submatrix problem. *Journal of Computational Biology*, 10(3-4):373–384, 2003.

[4] S. Bergmann, J. Ihmels, and N. Barkai. Iterative signature algorithm for the analysis of large-scale gene expression data. *Physical review E*, 67(3):31902, 2003.

[5] J. Besson, C. Robardet, and J.F. Boulicaut. Mining a new fault-tolerant pattern type as an alternative to formal concept discovery. *LNCS*, 4068:144, 2006.

[6] T. Calders, B. Goethals, and S. Jaroszewicz. Mining rank-correlated sets of numerical attributes. In *ACM SIGKDD*, page 105. ACM, 2006.

[7] H. Cheng, PS Yu, and J. Han. Ac-close: Efficiently mining approximate closed itemsets by core pattern recovery. In *ICDM*, pages 839–844, 2006.

[8] H. Cheng, P.S. Yu, and J. Han. Approximate frequent itemset mining in the presence of random noise. *Soft Computing for Knowledge Discovery and Data Mining*, page 363, 2007.

[9] Y. Cheng and GM Church. Biclustering of gene expression data. In *ISMB 2000*, pages 93–103, 2000.

[10] G. Cong, K.L. Tan, A.K.H. Tung, and F. Pan. Mining frequent closed patterns in microarray data. *ace*, 125:123.

[11] C. Creighton and S. Hanash. Mining gene expression databases for association rules, 2003.

[12] I.S. Dhillon, S. Mallela, and D.S. Modha. Information-theoretic co-clustering. In *ACM SIGKDD*, pages 89–98. ACM New York, NY, USA, 2003.

[13] T. Fukuda, Y. Morimoto, S. Morishita, and T. Tokuyama. Mining optimized association rules for numeric attributes. *Journal of Computer and System Sciences*, 58(1):1–12, 1999.

[14] R. Gupta, G. Fang, B. Field, M. Steinbach, and V. Kumar. Quantitative evaluation of approximate frequent pattern mining algorithms. In *ACM SIGKDD*, pages 301–309. ACM, 2008.

[15] Rohit Gupta, Smita Agrawal, Navneet Rao, Ze Tian, Rui Kuang, and Vipin Kumar. Integrative Biomarker Discovery for Breast Cancer Metastasis from Gene Expression and Protein Interaction Data Using Error-tolerant Pattern Mining. In *Proc of the International Conference on Bioinformatics and Computational Biology (BICoB)*, 2010.

[16] A. Gyenesei, R. Schlapbach, E. Stolte, and U. Wagner. Frequent pattern discovery without binarization: Mining attribute profiles. *LNCS*, 4213:528, 2006.

[17] S. Hanhijärvi, M. Ojala, N. Vuokko, K. Puolamäki, N. Tatti, and H. Mannila. Tell me something I don't know: randomization strategies for iterative data mining. In *ACM SIGKDD*, pages 379–388. ACM New York, NY, USA, 2009.

[18] T.R. Hughes, M.J. Marton, A.R. Jones, C.J. Roberts, R. Stoughton, C.D. Armour, H.A. Bennett, E. Coffey, H. Dai, Y.D. He, et al. Functional discovery via a compendium of expression profiles. *Cell*, 102(1):109–126, 2000.

[19] The Gasch Lab. http://gasch.genetics.wisc.edu/datasets.html.

[20] J. Liu, S. Paulsen, X. Sun, W. Wang, A. Nobel, and J. Prins. Mining approximate frequent itemsets in the presence of noise: Algorithm and analysis. In *SDM*, pages 405–416, 2006.

[21] J. Liu, S. Paulsen, W. Wang, A. Nobel, and J. Prins. Mining approximate frequent itemsets from noisy data. In *IEEE ICDM*, page 4, 2005.

[22] S.C. Madeira and A.L. Oliveira. Biclustering algorithms for biological data analysis: a survey. *IEEE Transactions on computational Biology and Bioinformatics*, pages 24–45, 2004.

[23] T. McIntosh and S. Chawla. High-Confidence Rule Mining for Microarray Analysis. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, pages 611–623, 2007.

[24] G. Pandey, G. Atluri, M. Steinbach, C.L. Myers, and V. Kumar. An association analysis approach to biclustering. In *ACM SIGKDD*, pages 677–686. ACM New York, NY, USA, 2009.

[25] A.K. Poernomo and V. Gopalkrishnan. Mining statistical information of frequent fault-tolerant patterns in transactional databases. In *ICDM*, pages 272–281. IEEE Computer Society Washington, DC, USA, 2007.

[26] A.K. Poernomo and V. Gopalkrishnan. Towards efficient mining of proportional fault-tolerant frequent itemsets. In *ACM SIGKDD*, pages 697–706. ACM New York, NY, USA, 2009.

[27] A. Prelic, S. Bleuler, P. Zimmermann, A. Wille, P. B "uhlmann, W. Gruissem, L. Hennig, L. Thiele, and E. Zitzler. A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics*, 22(9):1122–1129, 2006.

[28] R. Rastogi and K. Shim. Mining optimized association rules with categorical and numeric attributes. *IEEE TKDE*, pages 29–50, 2002.

[29] J.K. Seppänen and H. Mannila. Dense itemsets. In *ACM SIGKDD*, pages 683–688. ACM New York, NY, USA, 2004.

[30] R. Srikant and R. Agrawal. Mining quantitative association rules in large relational tables. *ACM SIGMOD Record*, 25(2):12, 1996.

[31] A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, and J. P. Mesirov. Gene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. *Proc Natl Acad Sci U S A*, 102(43):15545–50, 2005.

[32] A. Tanay, R. Sharan, and R. Shamir. Discovering statistically significant biclusters in gene expression data, 2002.

[33] C. Yang, U. Fayyad, and P.S. Bradley. Efficient discovery of error-tolerant frequent itemsets in high dimensions. In *ACM SIGKDD*, pages 194–203. ACM New York, NY, USA, 2001.

[34] M. Zhang, W. Wang, and J. Liu. Mining approximate order preserving clusters in the presence of noise. In *Proc. ICDE*, volume 8, pages 160–168. Citeseer, 2008.

Systematic construction and analysis of co-expression networks for identification of functional modules and cis-regulatory elements

Jianhua Ruan[1,*], Joseph Perez[1,2], Brian Hernandez[2], Garry Sunter[2] and Valerie M. Sponsel[2]
[1]Department of Computer Science and [2]Department of Biology
University of Texas at San Antonio
One UTSA Circle, San Antonio, TX 78249

*Corresponding author email: jruan@cs.utsa.edu

ABSTRACT

Gene co-expression networks have been used successfully for discovering biological relationships among genes on a whole-genome scale, such as predicting gene functional modules and cis-regulatory elements. However, those networks are often constructed in an ad hoc manner, and various methods for network construction and analysis have not been fully evaluated and compared. In this study, we propose a method for constructing gene co-expression networks based on mutual k-nearest-neighbor graphs (mKNN), and compare it with two widely used approaches: threshold-based approach and asymmetric k-nearest-neighbor graph approach (aKNN). We show that mKNN is more robust with respect to the presence of experimental noise and scatter genes, and is less sensitive to parameter variations. Furthermore, we propose a topology-based criterion to guide the selection of the optimal parameter for mKNN, and combine the method with a modularity-based community discovery algorithm to predict functional modules. We evaluate the method on both synthetic and real microarray data. On synthetic data, our method, which does not require any user-tuned parameters, is superior to several popular methods in recovering the embedded modules. Using the yeast stress-response microarray data, we show that the overall functional coherence of the modules predicted by our method using the automatically determined parameters is close to optimal. Finally, we apply the method to study a large collection of gene expression microarray data in *Arabidopsis thaliana*. Remarkably, with our simple method, we have found many functional modules that are much more significant than those reported by previous studies on the same data set. In addition, we are able to predict cis-regulatory elements for the majority of the functional modules, and the association between the cis-regulatory elements and the functional modules can often be confirmed by existing knowledge.

BIOKDD '10 Arlington, VA USA

Categories and Subject Descriptors

J.3 [**Life and Medical Sciences**]: Biology and Genetics; H.2.8 [**Database applications**]: Data Mining

General Terms

Algorithms, Experimentation, Performance

Keywords

Bioinformatics, microarray, co-expression network, functional module, cis-regulatory element

1. INTRODUCTION

Gene co-expression networks has been shown as an important and useful technique in discovering knowledge from gene expression microarray data, with many interesting results being reported [1, 3, 4, 6, 10, 11, 13, 16, 17, 22, 26, 27, 29, 30, 31]. Surprisingly, however, although many methods have been developed to analyze such networks, the problem on how to construct the network in the first place has not been well studied. Most approaches connect two genes whenever the similarity (or some transformation of similarity) between their expression levels is above some threshold [1, 3, 6, 10, 11, 13, 12, 16, 26, 27, 29, 30, 31]. This threshold is usually dataset dependent, although a few ideas have been proposed to help in the automatic selection of the threshold [3, 11]. A problem with threshold-based approach is that different biological processes may show different levels of co-expression. Therefore, it is unlikely a single threshold can be used to define all co-expression links.

Recently, we proposed a k-nearest-neighbor approach to construct gene co-expression networks [18, 17]. Basically, for each gene g, we connect it to k other genes whose similarity to g is ranked the top k among all the genes. In the final network, each gene may have more than k connections, since the rank of similarities is asymmetric: gene A may list gene B as a top-k friend, but gene B may not list gene A as a top-k friend - in this case, we still treat A and B as connected. The advantage of this approach is that two genes sharing only weak expression similarity may be linked. We showed that a small k is needed to keep the whole network connected, and partitioning the network can result in higher module prediction accuracy than conventional clustering algorithms [18]. A problem with this approach, however, is that the microarray data needs to be preprocessed so that

genes unrelated to the process of interest are removed before the construction of the network, to prevent them from being accidentally included in the network.

In this study, we propose a mutual k-nearest neighbor approach, which solves the problem of unspecific connections in the asymmetric KNN graphs, and is robust to random noise and scatter genes. We also propose a topology-based criterion to automatically determine the optimal k for constructing gene co-expression networks. We then apply a modularity-based community discovery algorithm to partition the network into relatively dense subnetworks as candidates of functional modules. To evaluate our method, we test the method on a large collection of synthetic microarray data. Our method, which does not require any user-tuned parameter, has achieved much higher clustering accuracy than several previous methods, even if we give the other methods the advantage of knowing the correct number of clusters or the optimal parameters. We also evaluate our method using the yeast stress-response microarray data, where our algorithm has identified a large number of significant functional modules. Using an objective evaluation metric, we believe that our approach has found an "optimal" clustering for this data set, using the automatically determined network parameter.

Finally, we apply our method to construct a whole-genome gene co-expression network for *Arabidopsis thaliana* using more than one thousand microarray experiments. From the network we have identified many interesting clusters that are functionally coherent and potentially co-regulated. Remarkably, the functional modules we predicted are statistically much more significant than those reported by previous studies on the same data set. In addition, we have predicted cis-regulatory elements for many of the functional modules, and the association between the cis-regulatory elements and the functional modules can often be confirmed by existing knowledge.

The remainder of the paper is organized as follows. In Section 2, we describe the methods for network construction and analysis and the evaluation metrics. In Section 3, we present evaluation results on synthetic microarray data. In Section 4, we show results on two real microarray data - yeast data as an evaluation and Arabidopsis data as a real application. We conclude and discuss some future directions in Section 5.

2. METHODS

2.1 Network construction

Given a set of genes, each of which is associated with a d-dimensional vector, we are interested in constructing a sparse network that can capture the inter-gene relationships. Sparseness is often preferred when one is dealing with networks of thousands of vertices. It is usually expected that each gene can only interact with a limited number of other genes. Therefore, a sparse network is usually sufficient to capture most of the inter-gene relationships, while enabling efficient storage and analysis. It also enables visualization of the relationships easily. Furthermore, many topology-based network analyses can only be applied to un-weighted graphs, which is usually also sparse.

All network construction approaches depend on some metric that measures the similarity between two genes. Several metrics are popular choices, such as Pearson correlation coefficient, Euclidean distance, cosine similarity, mutual information etc, and different metrics may be useful under difference circumstances. Here we assume that an appropriate metric has been chosen, based on which a similarity score has been computed for each pair of genes. Let s_{ij} be the similarity between gene i and gene j.

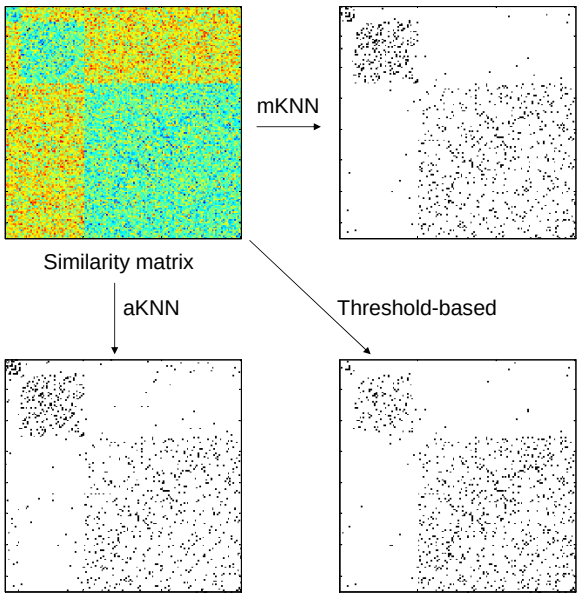


Figure 1: Illustration of three co-expression network construction methods.

We define a network as $G = \{V, E\}$, where V is the set of entities and E is the set of edges. Alternatively, we represent a network by its adjacency matrix, $W = (w_{ij})$, where $w_{ij} = 1$ if there is an edge between v_i and v_j, and 0 otherwise. We consider three approaches for constructing a sparse network.

(1) Threshold-based: given a similarity matrix $S = (s_{ij})$, and a similarity threshold c, the network can be obtained by letting $w_{ij} = 1$ if $s_{ij} \geq c$ or 0 otherwise.

(2) Asymmetric k-nearest neighbors: two genes are connected if one is within the top-k most similar genes of the other. Formally, we let $w_{ij} = 1$ if $s_{ij} \geq \min\{s_{ii_k}, s_{jj_k}\}$ or 0 otherwise, where i_k is the index of the gene whose similarity to gene i is smaller than exactly $k-1$ other genes. In other words, $|x, x \neq i \text{ and } s_{ix} > s_{ii_k}| = k-1$.

(3) Mutual k-nearest neighbors: two genes are connected if they are within the top-k most similar genes of each other. Formally, we let $w_{ij} = 1$ if $s_{ij} \geq \max\{s_{ii_k}, s_{jj_k}\}$ or 0 otherwise, where i_k is the same as above.

The advantage of the mKNN methods compared to the other two methods can be explained by a small example in Figure 1, which shows a similarity matrix containing three clusters of different sizes (10, 40, and 100, respectively) and three networks constructed by the above methods. We have chosen parameters to make the three networks to have approximately the same density. Assume that the diagonal blocks (within-cluster gene pairs) in the similarity matrix are generated from the same distribution, and that the similarity scores in the off-diagonal regions (inter-cluster gene pairs) are generated from a different distribution. In the threshold-based method, all entries in the diagonal blocks have the same probability to be selected as edges. As a

result, the expected edge density in different clusters will be the same. This, however, creates a huge disadvantage for the vertices in the smaller clusters, as they will have a smaller number of within-cluster edges compared to those in larger clusters. Even worse, they may by chance have more inter-cluster edges. In the two KNN-based networks, in contrast, the smaller clusters usually have higher within-cluster edge densities, because the networks were constructed by connecting each vertex to the same number of neighboring vertices. (This does not mean all vertices have the same number of edges, however.) Indeed, as shown in Figure 1, the smallest cluster represented by the upper left diagonal block has much higher edge densities in the two KNN-based networks than in the threshold-based network. The mKNN-based network also has fewer inter-cluster edges in the off-diagonal regions than the aKNN-based network.

2.2 Topology-based parameter selection

Each of the above methods depends on a single parameter. The threshold-based approach needs a cutoff value. When Pearson correlation coefficient is used, one usually chooses a fixed value for all genes, or equivalently, one can use a p-value cutoff to control the significance of the correlation coefficients. The cutoff correlation coefficient value has been chosen from as high as 0.95 to as low as 0.65, depending on the type of the application. For other types of similarity metric, it is typically difficult to decide a fixed parameter in advance. One usually tests a range of values and chooses one by experience. For the KNN-based method, the parameter k is usually chosen arbitrarily. It is highly desirable that these parameters can be chosen automatically.

It is known that many real networks, including biological networks and the Internet, share common topological properties that are different from random networks [15]. For example, real networks often have a long-tail degree distribution, the small-world property, and high clustering coefficient [15]. Therefore, it is often suggested that these properties may be used to distinguish real networks from their random counterparts [15, 3].

The topology-based parameter selection method works as follows. Given a co-expression network construction method and a topological measure Γ, we first decide a set of possible values for the parameter (e.g., Pearson-correlation coefficient for the threshold-based method and k for a KNN-based method). We then construct a co-expression network using each parameter value, and compute the Γ value of the resulting network. At the same time, we also generate a random network by applying the same network construction method to a randomly permutated copy of the original expression data, and compute the corresponding Γ value of the random network. We then choose the network parameter that maximizes the difference between Γ_{true} and Γ_{random}. Formally, let $G(A, v)$ be the co-expression network generated on data set A using parameter v, and let A^r be the permutated data, the optimal network G^* is constructed as follows:

$$G^* = G(A, v^*), \text{ where } v^* = \underset{v}{\operatorname{argmax}} \Gamma_{G(A,v)} - \Gamma_{G(A^r,v)}. \quad (1)$$

Here we consider two types of topological measures. The first is the clustering coefficient, defined by the following formula: $C = \frac{1}{N}\sum_i 2n_i/k_i(k_i-1)$, where N is the number of vertices in the network, k_i is the degree of vertex i, and n_i is the number of connections between the neighbors of vertex i. In a recent study, Elo and colleagues recommended using clustering coefficient to choose the optimal network parameter [3]. Their experimental results were based exclusively on threshold-based networks. Furthermore, a subtle but significant difference is that in their method, the random network was generated by randomly rewiring the true network. In contrast, in our method, the random network was generated by applying the same network construction method to a randomly permutated data set. As a randomly rewired network has no clustering structure at all, its clustering coefficient is close to zero, when the network is sufficiently sparse. In contrast, the clustering coefficient of a network constructed from a random data set is non-negligible. In addition, our method searches for the parameter that corresponds to a global maximum value of $\Gamma_{true} - \Gamma_{random}$, while their method searches for the parameter that corresponds to the first local maximum of $\Gamma_{true} - \Gamma_{random}$. As a result, our method is less prone to noises than their method.

The second type of topological measure we propose is a novel measurement specific for the mKNN method. Assume that we choose parameter k in the mKNN method, and the average vertex degree of the resulting network is n_k. We define the *normalized degree* of the network as n_k/k. The normalized degree for any mKNN network is between 0 and 1. We use the normalized degree as the topological measure, and apply Equation (1) to choose a k that maximizes the difference between the normalized degree of the true network and that of its random counterpart. The rationale is as follows. In the mKNN network, the normalized degree is related to the conditional probability $p(s_{ij} \geq s_{ii_k} | s_{ij} \geq s_{jj_k})$. Consider a similarity matrix where the similarity scores are completely random, which means $p(s_{ij} \geq s_{ii_k})$ and $p(s_{ij} \geq s_{jj_k})$ are independent. When each vertex chooses k neighbors, the probability for each of the k neighbors to also rank the current vertex as a top-k neighbor is exactly k/N, where N is the size of the network. The expected degree is therefore k^2/N and the expected normalized degree would be $k^2/N/k = k/N$. In a non-random similarity matrix that has clustering structures, when k is small (or more precisely, smaller than a typical cluster size), the k nearest neighbors of most vertices are members of their clusters, and therefore the expected degree for each vertex would be k^2/n, where n is the size of the cluster that the vertex is in. The average degree of the network would be proportional to ck^2/N where c is the number of clusters. Consequently, the normalized degree would be proportional to ck/N and the difference between the normalized degree of the true network and that of the random network would grown as k grows, until k is about the same size of a typical cluster. After that, when k increases, new neighbors for most vertices would be chosen primarily from outside of their clusters, randomly. The probably $p(s_{ij} \geq s_{ii_k} | s_{ij} \geq s_{jj_k})$ now drops to k/N from k/n and as a result, the difference between the normalized degree of the true and random networks would decrease when k increases.

2.3 Module detection and annotation

Many module detection algorithms have been developed, most of which rely on some graph partitioning routines. We recently developed two graph partitioning algorithms within the framework of community discovery, which aims to identify the most interesting "natural" communities (i.e., relatively dense subnetworks) without user-tuned parameters [19]. The first algorithm, called *Qcut*, partitions a net-

work by optimizing a well-known modularity function [19]. The second algorithm, called *HQcut*, solves the intrinsic resolution limit problem of the modularity function by iteratively calling *Qcut* to identify communities that does not contain any statistically significant sub-communities [19]. Here we employ the *HQcut* algorithm to the co-expression networks and treat the identified communities as candidates of functional modules. *HQcut* does not use any user-tunable parameters, except an optional statistical significance cutoff. We used a fixed cutoff (p-value = 0.05) throughout the paper. Previously we have shown that in general the results of *HQcut* are not sensitive to this cutoff value [18].

We use enrichment of Gene Ontology terms to evaluate the significance of functional modules [24]. Specifically, given a gene subnetwork s and a Gene Ontology term t, the p-value for the enrichment of t in s is estimated by the cumulative hypergeometric test:

$$p(t,s) = \sum_{k=a}^{\min(m,n)} \frac{\binom{m}{k}\binom{N-m}{n-k}}{\binom{N}{n}}, \quad (2)$$

where N is the number of genes in the genome, m is the size of the subnetwork, n is the number of genes in the genome with function t, and a is the number of genes in s with function t.

To evaluate the overall functional coherence of all predicted functional modules and compare the results of different algorithms, we propose the following functional coherence score:

$$S = \sum_t \max_s(-\log_{10} p(t,s)), \quad (3)$$

where t iterates over all gene ontology terms and s iterates over all functional modules.

This is essentially Fisher's combined probability test, treating the enrichment of each functional term as a hypothesis. Intuitively, for each gene ontology term t, we test whether it is significantly enriched in each cluster using the cumulative hyper-geometric test, and take the most significant p-value across all clusters. We then multiply the p-values for all functional terms, and take the negative logarithm of the product as the final functional coherence score. A higher score means the algorithm has discovered more functionally significant GO terms and better overall quality. Consider an intuitive alternative definition, $S' = \sum_s \max_t(-\log_{10} p(t,s))$, which treats the enrichment of some functional terms within each subnetwork as a hypothesis. As different algorithms may have identified different numbers of clusters, S' would be biased towards an algorithm that returns more clusters. In contrast, S is not biased by the number of clusters because the number of hypothesis tests is the same, independent of the number of clusters.

2.4 Discovery of cis-regulatory elements

To establish the connection between co-expression and co-regulation in real microarray data (specifically, Arabidopsis microarray data in this work), we explore the known transcription factor binding sites to find cis-regulatory elements (motifs) within each functional module. To do this, the promoter region (1000 bp upstream from the transcription start site) of each gene in a module is scanned with over 500 known motifs curated in the PLACE database, represented as consensus sequences [7]. The idea is that if a motif is found to be enriched in the genes' promoters in a module, then perhaps those genes are regulated by that motif.

The motif scanning was done by our own program. To account for motif degeneracy, we allow a certain number of mismatches during the motif scanning. For long consensus sequences, this is necessary because many transcription factor binding sites are different from their canonical consensus sequences. How to determine the number of mismatches to be allowed, however, is not trivial. We propose a simple strategy to search the optimal number of mismatches (for each motif) that can result in the most significant enrichment of the motif in a particular cluster. The sequence occurrence of the motif (with up to l mismatches) within a module is compared to that in the entire genome and the enrichment of the motif in the module is computed using the cumulative hyper-geometric test similarly as for testing the enrichment of Gene Ontology terms. We vary l between 0 and L, where L is proportional to the length of the motif, and choose the optimal l that gives the most significant enrichment of the motif.

3. RESULTS ON SYNTHETIC DATA

3.1 Synthetic microarray data sets

In order to compare different approaches for constructing gene co-expression networks and identifying co-expressed modules, we used a large collection of synthetic microarray expression data sets that was originally proposed in [23]. Each data set contains about 600 genes that belong to one of fifteen clusters, plus zero or more scatter genes that do not belong to any cluster. The expression levels of the genes in the same cluster were generated according to a common log normal distribution. Each data set can be characterized by two parameters: Standard Deviation (SD), which defines the level of Gaussian noise added to the expression levels and thus the difficulty to separate different clusters, and R, which represents the ratio of the number of scatter genes to the number of clustered genes. In this experiment, R = 0, 1, and 2, representing the cases of no scatter genes, 1x scatter genes, and 2x scatter genes, respectively. Values of SD are 0.05, 0.1, 0.2, 0.4, 0.8, and 1.2. Furthermore, 100 data sets were generated for each combination of R and SD, giving a total of 3 x 6 x 100 = 1800 data files. In our experiments, unless otherwise mentioned, we assume no prior knowledge of the number of clusters, number of scatter genes, and amount of random noises in each data set.

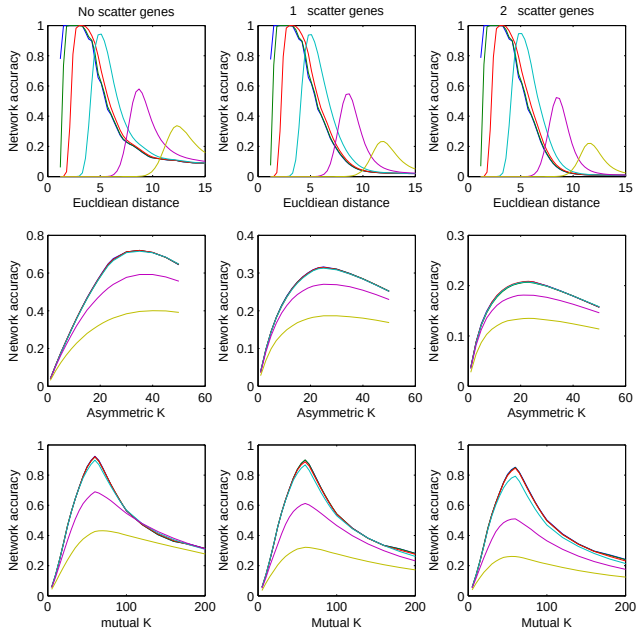
3.2 Network reconstruction accuracy

The reconstruction accuracy of a network is measured by

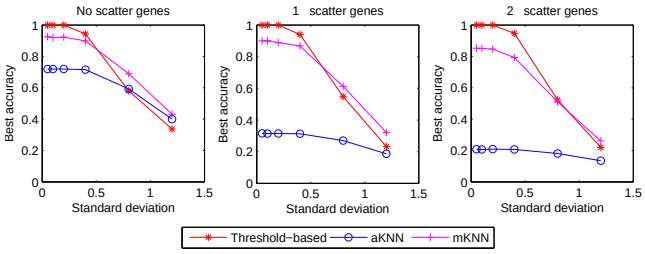
$$accuracy = \frac{|E_t \cap E_p|}{|E_t \cup E_p|},$$

where E_t and E_p are the sets of edges in the "true" co-expression network and the constructed network, respectively. To obtain the "true" co-expression network, we assume that genes in the same cluster are fully connected, and there is no connections between genes in different clusters.

We tested the network reconstruction accuracy of three methods: threshold-based method, aKNN, and mKNN. Figure 2(a) shows the accuracy of each method as a function of the network reconstruction parameter: distance cutoff for the threshold-based method, and k for the two KNN-based approaches. In Figure 2 (a), each row represents a different construction method (threshold-based, aKNN, and

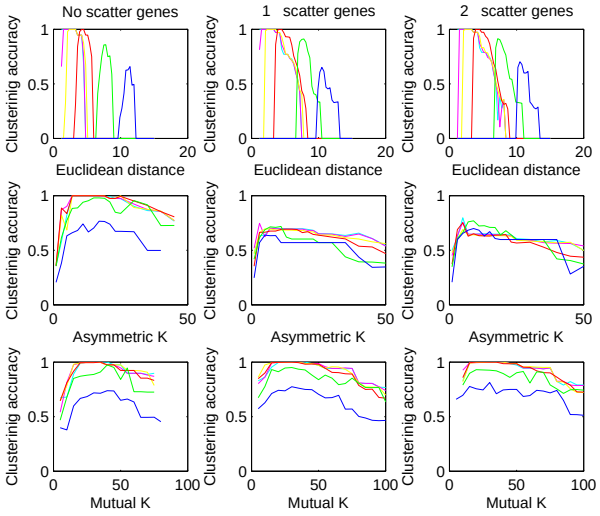


(a) Accuracy as a function of network parameters

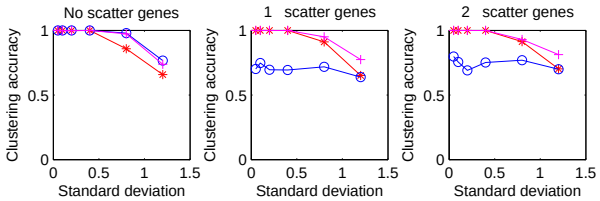


(b) Best accuracy

Figure 2: Network reconstruction accuracy.



(a) Accuracy as a function of network parameters



(b) Best accuracy

Figure 3: Network clustering accuracy (see Figure 2 and text for legend).

mKNN, respectively) and the three columns represent the data sets with 0×, 1×, and 2× scatter genes, respectively. Each subplot contains six curves, corresponding to the data sets with different random errors: SD = 0.05 (blue), 0.1 (green), 0.2 (red), 0.4 (cyan), 0.8 (magenta), and 1.2 (yellow). Figure 2(b) shows the best reconstruction accuracy of the three approaches on each data set assuming that optimal parameters were chosen.

The threshold method achieves the best accuracy (> 0.95) among the three methods when SD ≤ 0.4, but its accuracy drops significantly when $SD \geq 0.8$. The mKNN method, in contrast, has slightly lower accuracy (≈ 0.9) for SD ≤ 0.4, but is better than the threshold method for SD $>=$ 0.8 with or without scatter genes. For example, when R=0, the accuracy of mKNN is 0.63 for SD = 0.8 and 0.39 for SD = 1.2, while the accuracy of the threshold method is 0.54 and 0.30, for SD = 0.8 and 1.2, respectively.

The aKNN method performs reasonable well when no scatter genes are included (accuracy = 0.76, 0.58 and 0.37 for SD = 0.4, 0.8, and 1.2, respectively), but its accuracy decreases dramatically when scatter genes are present. In contrast, the accuracy of mKNN and that of threshold-based method are not affected significantly by the addition of scatter genes. In an aKNN network, every gene has at least k edges; therefore, many spurious edges involving the scatter genes will be created in aKNN. On the other hand, for the data set with SD = 1.2, aKNN has similar accuracy as the threshold method, even with scatter genes. As most real microarray data are noisy, this suggests that in real situations aKNN may not be an entirely bad option.

As shown in Figure 2(a), the optimal parameter for the threshold-based method varies when SD increases. In contrast, the optimal parameters for the two KNN methods are relatively invariant with respect to SD values. We have also observed that the optimal parameters for the KNN methods depend on the logarithm of the number of genes in the data set (data not shown). For most modest-sized data sets, therefore, the optimal parameters for the KNN methods can be relatively easily selected. In addition, we show later that choosing the optimal k can be guided by several network topological properties.

Finally, we argue that the network reconstruction accuracy measurement is flawed, if our ultimate objective is to identify functional modules, or gene clusters. In measuring the network reconstruction accuracy, we have required that a perfect co-expression network should have all the genes in the same cluster completely connected, and have no edges between genes in different clusters. As cluster sizes vary, this measurement favors methods that emphasize more on the larger cluster, as the number of within-cluster edges is a quadratic function of cluster size. For network to be correctly clustered, however, one does not need the genes within each cluster to be completely connected. In fact, in the ex-

treme case, as long as there is no connection between clusters, and there is a path connecting any pair of genes in the same cluster, clusters can be easily identified by finding the independent components. Assuming that the connections within a cluster is essentially random, the number of connections required to ensure all genes can be connected to a single component with a high probability is in fact incredibly small (≈2-3 for aKNN and $\log(k)$ for mKNN, where k is the size of a cluster). In general, as long as the genes within the same cluster are relatively well-connected compared to the genes between clusters, clusters can be identified with high accuracy. Both KNN methods have the advantage of focusing on the relatively smaller / weaker clusters. For the threshold-based method, it achieves high reconstruction accuracy by connecting the larger clusters all together while ignoring smaller clusters.

3.3 Network clustering accuracy

For networks with no scatter genes, we measured the clustering accuracy by the adjusted Rand Index [14]. Given a set of objects $S = s_1, s_2, \ldots, s_n$, let $X = \{X_1, X_2, \ldots, X_M\}$ and $Y = \{Y_1, Y_2, \ldots, Y_N\}$ represent the true and predicted partitions of the objects, where each object appears in X and Y exactly once. Let n_{ij} be the number of common objects between X_i and Y_j. Also let $n_{i\bullet} = \sum_j n_{ij} = |X_i|$ be the size of X_i, and $n_{\bullet j} = \sum_i n_{ij} = |Y_j|$ be the size of Y_j. The adjusted Rand Index can be computed by:

$$R(X,Y) = \frac{\sum_{i,j}\binom{n_{ij}}{2} - \sum_i \binom{n_{i\bullet}}{2}\sum_j \binom{n_{\bullet j}}{2}/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{n_{i\bullet}}{2} + \sum_j \binom{n_{\bullet j}}{2}] - \sum_i \binom{n_{i\bullet}}{2}\sum_j \binom{n_{\bullet j}}{2}/\binom{n}{2}}$$

For networks with scatter genes, we follow the strategy of [23]. Let X_M and Y_N be the cluster that contains all the scatter genes in X and Y, respectively. Although we can use $R(X,Y)$ to estimate the accuracy, it gives equal weight to scatter genes and clustered genes. Therefore, we remove the scatter genes from both partitions, and obtain two new partitions $X' = \{X_1 \setminus Y_N, X_2 \setminus Y_N, \ldots, X_{M-1} \setminus Y_N\}$ and $Y' = \{Y_1 \setminus X_M, Y_2 \setminus X_M, \ldots, Y_{N-1} \setminus X_M\}$. The clustering accuracy is then measured by

$$R'(X,Y) = \frac{R(X,Y) + R(X',Y')}{2}.$$

Figure 3(a) shows the clustering accuracy of the *HQcut* algorithm on the networks constructed with the threshold-based method and the two KNN-based methods, as a function of network parameters. Figure 3(b) shows the optimal clustering accuracy when the best network construction parameters[1] are used.

Similar to network reconstruction accuracy, the clustering accuracy of *HQcut* on the threshold-based networks and the mKNN networks are relatively robust to the presence of scatter genes, while the accuracy of *HQcut* on aKNN networks degrade significantly in presence of scatter genes. In order to obtain the best accuracy on the threshold-based networks, different threshold parameters need to be used for different data sets, and the performance may be significantly worse if the parameter is not optimized. In contrast, the best accuracy on the aKNN and mKNN networks can be achieved with an almost uniform choice of network parameter for all data sets, and the performance degradation is much mild even when a sub-optimal network parameter is chosen.

[1]These parameters may not be identical to the parameters that give the best network accuracy.

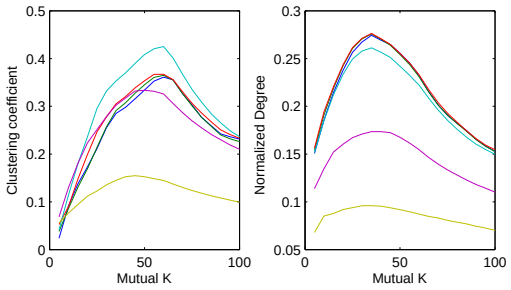


Figure 4: Topology of mKNN networks (no scatter genes)

When the level of Gaussian noises is low (SD $\leq$ 0.4) and no scatter genes are present, all methods have resulted in almost perfect clustering accuracy, indicating that genes in different clusters are well separated in these data sets. With the addition of scatter genes, however, the aKNN-based network has a much lower accuracy than the other methods on these well-separated data sets. When SD $\geq$ 0.8, mKNN has better accuracy than the threshold-based method for all values of R. Interestingly, aKNN also shows better accuracy than threshold-based accuracy for SD = 1.2, even when scatter genes are present.

Comparing Figure 3 with Figure 2, it is clear that the network reconstruction accuracy does not predict the network clustering accuracy very well. For example, although mKNN and aKNN have lower network reconstruction accuracy than the threshold method for SD $\leq$ 0.4 and R = 0, all three types of networks have resulted in almost identical clustering accuracies. The clustering accuracy on the aKNN networks are disproportionally high compared to their reconstruction accuracy. Furthermore, the optimal network parameters that resulted in the best clustering accuracy are usually different from the optimal network parameters that resulted in the best network reconstruction accuracy.

In order to provide an automated method for deciding the most appropriate k for constructing an mKNN network, we looked at several network topological measurements. Since it is known that real-world networks often have topological characteristics that are absent in random networks, we hypothesize that when we vary the network parameters, these measures may peak at the point that best separates a real-world network from its random counterpart. Therefore, we computed clustering coefficients and normalized degrees for mKNN network constructed with different values of k (Section 2.2). Interestingly, as shown in Figure 4, the clustering coefficient measure peaked at the k that gives the best network reconstruction accuracy, while the normalized degree measure peaked almost perfectly at the k that gives the best clustering accuracy. Very similar results were obtained for the data sets with 1x or 2x scatter genes. Therefore, in the remainder of the paper, we use the normalized degree to automatically determine k for constructing mKNN networks, unless otherwise mentioned.

3.4 Comparison of mKNN-based clustering methods with other methods

We compared the best clustering accuracy of *HQcut* on mKNN networks by searching all possible values of k (mKNN-HQcut-opt), the clustering accuracy of *HQcut* on mKNN

networks where values of k are determined automatically by optimizing normalized degree (mKNN-HQcut-auto), and the clustering accuracy of three other methods: Markov clustering algorithm (MCL) [2], k-means, and tight clustering [23]. MCL is one of the most popular graph partitioning algorithms in Bioinformatics literatures [2]. It has only a single parameter to tune, inflation. We applied MCL to the mKNN networks that have given *HQcut* the best clustering accuracy, and searched from a large range of values for the inflation parameter to achieve the best clustering accuracy.

As shown in Figure 5, the clustering accuracy of mKNN-HQcut-auto is very close to that of mKNN-HQcut-opt, even for the cases with a larger number of scatter genes and/or high Gaussian noises, indicating that our method is robust to both types of noises. The MCL algorithm has similar accuracy as mKNN-HQcut-auto, except for the data set with 2X scatter genes, for which MCL is slightly better. It is worth noting that MCL achieved this accuracy by searching over a broad range of values for the inflation parameter. We have found that no fixed value of inflation can be used to obtain a near optimal clustering accuracy for different data sets. For example, when SD = 1.2 and R = 2, the optimal inflation is 1.8. For SD = 0.8 and R = 2, the optimal inflation is 1.5 (accuracy = 0.93); the accuracy drops to 0.2 at I = 1.8. Finally, we found that the best accuracy of MCL is achieved in mKNN networks rather than in aKNN or threshold-based networks (data not shown), suggesting that the benefit of mKNN networks is not only for HQcut, but other graph-based clustering algorithm as well.

For the data sets *without* scatter genes, we also clustered each data set directly (without co-expression networks) using k-means implemented in Matlab 7.5. As k-means is stochastic, we used 10 random restart for each run of k-means to obtain the best results. We first run k-means with the correct number of clusters given explicitly, and then varied the number of clusters from 5 to 25, and used the gap statistic measure [25] to automatically determine the number of clusters. As shown in Figure 5(a), even with the number of clusters given explicitly, k-Means performed worse than our algorithm, and the gap statistic performed poorly in suggesting a correct number of clusters.

Finally, for the microarray data *with* scatter genes, k-mean performed very poorly (data not shown), as the scatter genes were often grouped together with genes in other clusters. Also, simply increasing the number of clusters did not improve the results. We therefore used the tight clustering, which have previously been shown to have achieved the best clustering accuracy on exactly the same data set [23]. Both mKNN-HQcut and mKNN-MCL have achieved much higher clustering accuracy than tight clustering (Figure 5(b) and (c)).

4. RESULTS ON REAL MICROARRAY DATA

4.1 Evaluation using yeast stress response data

We applied the method to cluster a large number of yeast genes based on their expression levels in response to a variety of stress treatments. This data set contains 173 dimensions, and as in most previous studies, we selected the top 3000 genes with the highest variances [5]. After quantile normalized the expression data, our algorithm constructed an optimal kNN network with $k = 120$, and identified 78 modules as well as about 150 singletons. As a comparison,

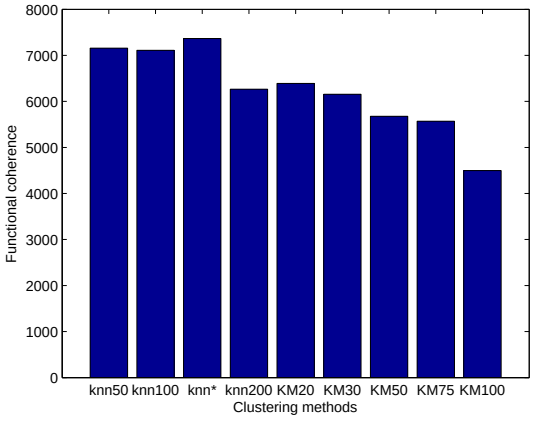


Figure 6: Results on yeast gene networks

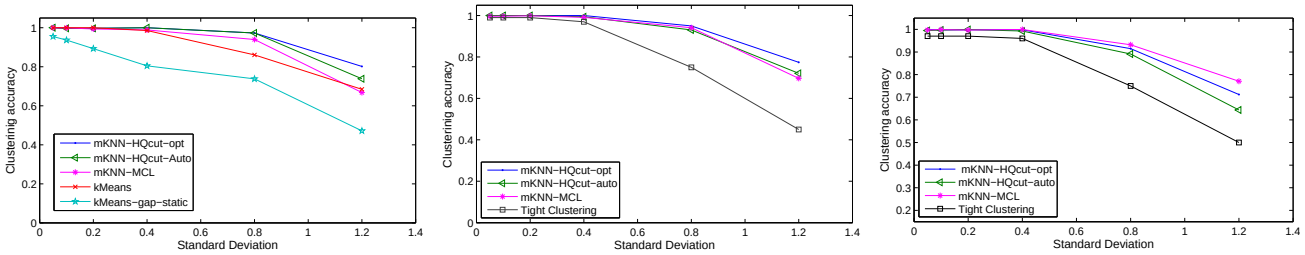
we constructed three additional mKNN networks using k = 50, 100, and 200, and applied *HQcut* to each of them. Note that a larger k will create a denser network, which will result in fewer clusters in general. Furthermore, we applied the k-means algorithm to the expression data directly, with the number of clusters setting to 20, 30, 50, 75, and 100. We used the k-means implemented in Matlab 7.5, and used 10 random restart for each run of k-means.

To compare the clustering results obtained by different methods or parameters, we computed a functional coherence score (FC) for each set of clustering result, based on Gene Ontology functional terms [24] and the Fisher's combined probability test (see Section 2.3). This test treats each GO term, rather than each cluster, as a hypothesis, and therefore is not biased by the number of clusters.

Figure 6 shows the FC scores for the clustering results obtained by mKNN-HQcut with k determined automatically (knn*), mKNN-HQcut with various k (knnxxx, xxx = 50, 100, and 200), and k-means with different number of clusters (KMxxx, xxx = 20, 30, 50, 75 and 100). Interestingly, the highest FC is achieved with the automatically determined k (knn*). Using $k = 50$ or 100, *HQcut* also resulted in comparable FC scores (knn50 and knn100, respectively), confirming that the algorithm is relatively robust to different values of k. Interestingly, k-means achieved its best FC score with only 20 clusters, and its FC scores for 75 clusters and 100 clusters are much worse. Further investigation discovered that the sizes of clusters obtained by our algorithm are more diverse than the sizes of k-means clusters (data not shown). In other words, our algorithm can identify both large and small clusters, while the sizes of k-means clusters are relatively uniform. Therefore, in k-means, if the number of clusters is set too high, large functional modules are likely forced to split, which reduces their functional significance, while if the number of clusters is set to too low, small functional modules are likely merged with others, preventing them from being detected.

4.2 Application to Arabidopsis gene expression data

Finally, as a real application, we applied mKNN-HQcut to a large collection of Arabidopsis gene expression microarray data from AtGenExpress, which includes more than 1000



(a) No scatter genes (b) 1x scatter genes (c) 2x scatter genes

Figure 5: Comparison of mKNN-HQcut with other methods

microarrays for various growth conditions, developmental stages, and tissues of Arabidopsis [20, 9]. A co-expression network is constructed with $k = 100$, determined by the normalized degree. Similarity is measured using Pearson correlation coefficient. The mean and median degree of the network is 33 and 26, respectively. The network contains a large connected component with 20389 genes, and 2202 genes in singletons or components of size 2-3. Using *HQ-cut*, we find 1474 modules from the largest component of the network, with sizes from 2 to 175.

Gene Ontology analysis revealed that many of the modules have enriched functions. For example, among the ≈800 (300) clusters whose sizes are at least 10 (20), 81.1% (88.0%) of them have at least one enriched function, with bonferroni corrected p-value ≤ 0.05. We also annotated each module with a list of known cis-regulatory elements (motifs) from the PLACE database that are over-represented in the promoter sequences of the genes in the module (see Section 2.4). Overall, 66.7% of the clusters with size ≥ 20 have at least one over-represented cis-regulatory element with nominal p-value < 0.001.

Table 1 lists top 15 clusters with the most significant Gene Ontology enrichment (top half), and 11 selected clusters with significantly enriched cis-regulatory elements (bottom half). The functional enrichment is extremely significant for some clusters. For example, we have found several clusters where the majority of genes are involved in the same specific functions (c_1402, ribosome, p < 1E-300; c_1473, photosynthetic membrane, p = 1.3E-137; c_1051, proteasome complex, p < 5.9E-126).

Statistical significance of the over-representation of cis-regulatory elements in the clusters is much weaker than that of GO terms; this is because cis-regulatory elements are usually short and degenerate, and as a result may appear in many promoter sequences just by chance. Nevertheless, based on the information from the PLACE database [7], we find that many of the associations between the functional modules and the enriched cis-regulatory elements can be explained (see motifs highlighted in Table 1). For example, c_453 is enriched with heat response genes, while the most significant motif in the cluster is heat shock element (HSE). Cluster c_992 contains nucleosome assembly genes and is enriched with OCETYPEINTHISTONE, a motif known to regulate a histone gene. Cluster c_992 has function in DNA replication, and is enriched with the binding sites of the E2F transcription factor, which plays a major role in regulating cell cycles. The most significant motif in c_1257, UPRMOTIFIIAT, is a cis-acting element regulating the unfolded protein response, which is activated in response to an accumulation of unfolded or misfolded proteins in endoplasmic reticulum [21]. Another cell-cycle related cluster, c_973, is enriched with MYBCOREATCYCB1, a core cis-regulatory element for the Arabidopsis cyclin B1:1 gene. Cluster c_701 is involved in aromatic compound metabolic process and is enriched with L1DCPAL1, which is a cis-regulatory element in a phenylalanine ammonia-lyase gene. c_294 contains water responsive genes and is enriched with a dehydration-responsive element. The EVENINGAT element found in c_778 is known as an important cis-element for circadian regulation. The OCSELEMENTAT motif enriched in c_302 was found in Arabidopsis glutathione S-transferase gene. Finally, a few clusters contain genes having functions in abiotic stress response or embryonic development (c_711, c_488, c_489, c_316, c_302), while the corresponding cis-regulatory elements are either the well-known ABA responsive elements (ABREs) or the ubiquitous CGCG-box, which is known to be involved in multiple signaling pathways in plants. Interestingly, ABRE, UP1/2ATMSD, SITEIIATCYTC, and several other motifs have occurred in multiple clusters, indicating that they may be involved in regulating multiple processes.

4.2.1 *Comparison with previous studies*

Several previous studies have also attempted to predict functional modules in Arabidopsis, using the same microarray data compendium, based on co-expression networks or clustering methods [8, 13, 28, 11]. It is worth noting that the previous co-expression networks were all constructed by some variants of the threshold-based methods. Remarkably, the enrichment of GO terms in our functional modules is much stronger than in all the previous studies to our knowledge. For example, Horan et. al. applied hierarchical clustering directly to the microarray data and obtained 916 clusters [8]. The most significant GO terms in their clusters are photosynthesis (p < 1.3E-89), ribosome (p < 5.3E-65), and proteosome complex (p < 1E-28). Mao et. al. constructed a co-expression network using a Pearson correlation coefficient cutoff 0.75 [13]. Using the Markov clustering algorithm (MCL) [2], they identified 527 clusters. The five most significant clusters contain genes in photosynthesis (p < 1.4E-52), protein biosynthesis (p < 5.7E-52), DNA metabolism (p < 9.1E-52), starch metabolism (p < 3.2E-19), and response to heat (p < 1.7E-17). Ma et. al. [11] and Vandepoele et. al. [28] have also used co-expression networks for predicting functional modules, but the overall goals/strategies of their studies are different ours. Ma et. al. attempts to find co-expressed neighbors of known guide genes. The five most significant GO terms found by Ma et. al. are response to

Table 1: Most significant clusters according to function or motif

C_ID	Size	Enriched Function	P-value	f_c	f_g	Enriched Motif	P-value
c_1402	174	structural constituent of ribosome	<1E-300	0.81	0.013	UP1ATMSD	<1E-16
c_1473	110	photosynthetic membrane	1.3E-137	0.55	0.012	ACGTROOT1.1	4.0E-15
c_1051	70	proteasome complex	5.9E-126	0.52	0.002	SITEIIATCYTC	6.0E-06
c_1474	154	plastid	3.7E-93	0.63	0.082	UP1ATMSD	1.4E-08
c_453	91	response to heat	1.2E-54	0.27	0.003	**HSE**	3.8E-11
c_992	47	nucleosome assembly	1.2E-50	0.43	0.002	**OCETYPEINTHISTONE**	2.3E-14
c_619	47	mitochondrion	5.4E-50	0.68	0.033	–	–
c_1434	53	plastid	5.1E-41	0.67	0.082	UP1ATMSD	5.6E-05
c_1463	56	chloroplast thylakoid	5.6E-38	0.41	0.010	–	–
c_620	65	mitochondrion	1.0E-32	0.46	0.033	SITEIIATCYTC	1.6E-07
c_1090	30	RNA splicing	1.7E-31	0.31	0.003	–	–
c_991	45	DNA metabolic process	2.2E-31	0.39	0.010	**E2FAT**	1.1E-06
c_148	112	nutrient reservoir activity	6.6E-31	0.14	0.002	RYREPEATBNNAPA	4.5E-12
c_1257	55	endoplasmic reticulum	7.3E-30	0.31	0.010	**UPRMOTIFIIAT**	3.7E-11
c_973	134	microtubule motor activity	8.2E-30	0.12	0.002	**MYBCOREATCYCB1**	6.1E-11
c_701	17	aromatic compound metabolic process	8.0E-24	0.54	0.009	**L1DCPAL1**	5.9E-08
c_294	26	response to water	2.6E-22	0.35	0.004	**DRECRTCOREAT**	5.1E-06
c_778	99	circadian rhythm	2.0E-17	0.08	0.002	**EVENINGAT**	2.2E-16
c_711	18	response to auxin stimulus	6.7E-15	0.37	0.008	**MYCATRD22**	4.9E-05
c_488	72	defense response	7.1E-15	0.19	0.021	**CGCGBOXAT**	2.3E-10
c_1369	59	ribonucleoprotein complex biogenesis and assembly	8.0E-15	0.18	0.009	UP2ATMSD	<1E-16
c_489	81	response to abiotic stimulus	9.3E-09	0.17	0.028	**CGCGBOXAT**	<1E-16
c_493	25	glutathione transferase activity	7.0E-07	0.08	0.002	**OCSELEMENTAT.4**	7.8E-16
c_316	9	abscisic acid mediated signaling	1.9E-06	0.23	0.002	**ABREATRD22**	1.0E-06
c_140	36	embryonic development ending in seed dormancy	1.6E-05	0.15	0.014	**ABRERATCAL**	4.0E-10
c_302	14	response to abscisic acid stimulus	1.9E-04	0.18	0.007	**ABRE3HVA1**	9.2E-07

f_c: Frequency in cluster. f_g: Frequency in genome.

heat (p < 9.4E-55), chromatin (p < 7.5E-48), response to auxin (p < 3.6E-41), proteasome complex (p < 6.7E-29), and starch metabolism (p < 6.5E-18). The work of Vandepoele et. al. combines co-expression with sequence-level conservation between Arabidopsis and poplar. The most significant GO terms they found are photosynthesis (p < 2.2E-87), ribosome biogenesis and assembly (6.1E-68), and DNA replication (p < 8.9E-26). In addition, it is worth noting that our network (mean vertex degree = 26) is much sparser than the network of Mao et. al. (mean vertex degree = 165), and that of Vandepoele et. al. (mean vertex degree = 717), our network is more sparse, making it easier for analysis and visualization. At the same time, our network covers about 95% of the Arabidopsis genes, whereas the networks by Ma et. al. and Mao et. al. only cover about 30% of Arabidopsis genes. As a result, we are able to identify more functional modules than in these previous studies.

5. CONCLUSIONS

In this study, we have proposed a mutual k nearest neighbor-based method for constructing gene co-expression networks, and compared its performance with two other methods, in both network reconstruction accuracy and network clustering accuracy. We also proposed a novel topological measure to guide the selection of the optimal network parameter. Combining the mKNN-based networks with a community identification algorithm, we find we can significantly improve the prediction accuracy of functional modules, in both synthetic and real microarray data. Our application to Arabidopsis leads to the discovery of the largest number of Arabidopsis functional modules in the literature; for many modules, we are able to annotate them with functional terms and cis-regulatory elements. Together, the high statistical significance of Gene Ontology enrichment and the agreement between cis-regulatory and functional annotations of these genes modules in Arabidopsis show that our Arabidopsis gene modules are excellent candidates of functional modules. Therefore, we believe that the results can be utilized to predict the functions of unknown genes in Arabidopsis, and to understand the regulatory mechanisms of many genes. In the near future we plan to apply de novo motif finding tools to identify novel motifs from the functional modules, and construct a database of co-expressed and co-regulated genes.

6. ACKNOWLEDGMENTS

This work is supported in part by a National Institutes of Health grant SC3GM086305 to JR and a National Science Foundation grant IOS-0848135 to VMS. This work is supported in part by a NIH grant SC3GM086305 to JR and a NSF grant IOS-0848135 to VMS. BH and JP are in the Undergraduate Mathematics and Biology Scholars program supported by NSF award UBM0634588 (Dr. David Senseman, PI).

7. REFERENCES

[1] S. Carter, C. Brechbuhler, M. Griffin, and A. Bond. Gene co-expression network topology provides a framework for molecular characterization of cellular state. *Bioinformatics*, 20:2242–50, 2004.

[2] S. V. Dongen. Graph clustering via a discrete uncoupling process. *SIAM J. Matrix Anal. Appl.*, 30:121–141, 2008.

[3] L. Elo, H. Jarvenpaa, M. Oresic, R. Lahesmaa, and T. Aittokallio. Systematic construction of gene coexpression networks with applications to human t

helper cell differentiation process. *Bioinformatics*, 23:2096–103, 2007.

[4] J. J. Faith, B. Hayete, J. T. Thaden, I. Mogno, J. Wierzbowski, G. Cottarel, S. Kasif, J. J. Collins, and T. S. Gardner. Large-scale mapping and validation of Escherichia coli transcriptional regulation from a compendium of expression profiles. *PLoS Biol*, 5:e8, 2007.

[5] A. Gasch, P. Spellman, C. Kao, O. Carmel-Harel, M. Eisen, G. Storz, D. Botstein, and P. Brown. Genomic expression programs in the response of yeast cells to environmental changes. *Mol Biol Cell*, 11:4241–4257, 2000.

[6] A. Ghazalpour, S. Doss, B. Zhang, S. Wang, C. Plaisier, R. Castellanos, A. Brozell, E. Schadt, T. Drake, A. Lusis, and S. Horvath. Integrating genetic and network analysis to characterize genes related to mouse weight. *PLoS Genet*, 2:e130, 2006.

[7] K. Higo, Y. Ugawa, M. Iwamoto, and T. Korenaga. Plant cis-acting regulatory DNA elements (PLACE) database:1999. *Nucleic Acids Res*, 27:297–300, 1999.

[8] K. Horan, C. Jang, J. Bailey-Serres, R. Mittler, C. Shelton, J. Harper, J. Zhu, J. Cushman, M. Gollery, and T. Girke. Annotating genes of known and unknown function by large-scale coexpression analysis. *Plant Physiol.*, 147:41–57, 2008.

[9] J. Kilian, D. Whitehead, J. Horak, D. Wanke, S. Weinl, O. Batistic, D'Angelo, E. Bornberg-Bauer, J. Kudla, and K. Harter. The AtGenExpress global stress expression data set: protocols, evaluation and model data analysis of uv-b light, drought and cold stress responses. *Plant J.*, 50:347–363, 2007.

[10] H. Lee, A. Hsu, J. Sajdak, J. Qin, and P. Pavlidis. Coexpression analysis of human genes across many microarray data sets. *Genome Res*, 14:1085–94, 2004.

[11] S. Ma, Q. Gong, and H. Bohnert. An Arabidopsis gene network based on the graphical Gaussian model. *Genome Res*, 17:1614–1625, 2007.

[12] P. Magwene and J. Kim. Estimating genomic coexpression networks using first-order conditional independence. *Genome Biol*, 5:R100, 2004.

[13] L. Mao, J. L. V. Hemert, S. Dash, and J. A. Dickerson. Arabidopsis gene co-expression network and its functional modules. *BMC Bioinformatics*, 10:346, 2009.

[14] M. Meila. Comparing clusterings: an axiomatic view. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 577–584, New York, NY, USA, 2005. ACM Press.

[15] M. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.

[16] M. Oldham, S. Horvath, and D. Geschwind. Conservation and evolution of gene coexpression networks in human and chimpanzee brains. *Proc Natl Acad Sci U S A*, 103:17973–8, 2006.

[17] M. Ray, J. Ruan, and W. Zhang. Variations in the transcriptome of alzheimer's disease reveal modular networks involved in cardiovascular diseases. *Genome Biol.*, 9:R148, 2008.

[18] J. Ruan, A. Dean, and W. Zhang. A general co-expression network-based approach to gene expression analysis: Comparison and applications. *BMC Syst. Biol.*, 4:8, 2010.

[19] J. Ruan and W. Zhang. Identifying network community structures with a high resolution. *Phys Rev E*, 77:016104, 2008.

[20] M. Schmid, T. S. Davison, S. R. Henz, U. J. Pape, M. Demar, M. Vingron, B. Schölkopf, D. Weigel, and J. Lohmann. A gene expression map of Arabidopsis development. *Nat. Genet.*, 37:501–506, 2005.

[21] M. Schröder and R. Kaufman. The mammalian unfolded protein response. *Annu Rev Biochem*, 74:739–789, 2005.

[22] J. Stuart, E. Segal, D. Koller, and S. Kim. A gene-coexpression network for global discovery of conserved genetic modules. *Science*, 302:249–55, 2003.

[23] A. Thalamuthu, I. Mukhopadhyay, X. Zheng, and G. Tseng. Evaluation and comparison of gene clustering methods in microarray analysis. *Bioinformatics*, 22(19):2405–12, 2006.

[24] The Gene Ontology Consortium. The gene ontology (GO) database and informatics resource. *Nucleic Acids Res*, 32, 2004.

[25] R. Tibshirani, G. Walther, and T. Hastie. Estimating the number of clusters in a data set via the gap statistic. *J. Roy. Statist. Soc. Ser. B*, 63:411–23, 2001.

[26] P. Tsaparas, L. Marino-Ramirez, O. Bodenreider, E. Koonin, and I. Jordan. Global similarity and local divergence in human and mouse gene co-expression networks. *BMC Evol Biol*, 6:70, 2006.

[27] V. van Noort, B. Snel, and M. Huynen. The yeast coexpression network has a small-world, scale-free architecture and can be explained by a simple model. *EMBO Rep*, 5:280–4, 2004.

[28] K. Vandepoele, M. Quimbaya, T. Casneuf, L. D. Veylder, and Y. V. de Peer. Unraveling transcriptional control in Arabidopsis using cis-regulatory elements and coexpression networks. *Plant Physiol.*, 150:535–546, 2009.

[29] D. Weston, L. Gunter, A. Rogers, and S. Wullschleger. Connecting genes, coexpression modules, and molecular signatures to environmental stress phenotypes in plants. *BMC Syst Biol*, 2:16, 2008.

[30] X. Zhou, M. Kao, and W. Wong. Transitive functional annotation by shortest-path analysis of gene expression data. *Proc Natl Acad Sci U S A*, 99:12783–8, 2002.

[31] D. Zhu, A. Hero, H. Cheng, R. Khanna, and A. Swaroop. Network constrained clustering for gene microarray data. *Bioinformatics*, 21:4014–20, 2005.

A Fast Markov Blankets Method for Epistatic Interactions Detection in Genome-wide Association Studies

Bing Han
Department of Electrical Engineering and Computer Science
University of Kansas
Lawrence, KS 66045, USA
hanbing@ittc.ku.edu

Xue-wen Chen
Department of Electrical Engineering and Computer Science
University of Kansas
Lawrence, KS 66045, USA
xwchen@ittc.ku.edu

Zohreh Talebizadeh
Section of Genetics, Children's Mercy Hospital and
University of Missouri-Kansas City
Kansas City, MO 64108, USA
ztalebi@cmh.edu

ABSTRACT

Understanding epistatic interactions among multiple genetic factors can help to improve pathogenesis, prevention, diagnosis and treatment of complex human diseases such as cardiovascular disease, cancer, and diabetes. Although the development of large genome-wide association studies provides us with an extraordinary opportunity to identify potential epistatic interactions that cause disease susceptibility, the sheer size of the genotyped data and the large amount of combinations of all the possible genetic factors present a significant challenge, both mathematically and computationally, to data mining society in developing powerful and time-efficient methods for epistatic interactions detection. Currently, most existing computational detection methods are based on the classification capacity of SNP sets, which often fail to identify SNP sets that are strongly associated with the diseases and tend to introduce more false positives. In addition, most methods are not suitable for genome-wide scale studies due to their computational complexity. To address these issues, we propose a new and fast Markov Blanket-based method, FEPI-MB, for epistatic interactions detection. Experimenal results on both simulated data sets and a real data set demonstrate that FEPI-MB significantly outperforms other existing methods and is capable of finding SNPs that have a strong association with common diseases. Moreover, we also show that FEPI-MB is time-efficient and can achieve a better performance comparing to other Markov Blanket learning methods.

Categories and Subject Descriptors

I.2.8 [**ARTIFICIAL INTELLIGENCE**]: Problem Solving, Control Methods, and Search –*Heuristic methods.*

General Terms

Algorithms, Performance, Experimentation.

Keywords

Biological data mining, Markov blankets, gene-gene interaction.

1. INTRODUCTION

Identifying genetic variants that confer an increased risk to particular diseases in human populations is very important in human genetics. Epistasis, referring to 'interactions between chromosomal regions' such as gene-gene or SNP-SNP (single-nucleotide polymorphism) interactions [1], plays a critical role in the genetic bases of complex diseases. It has long been observed that the combined effects of multiple genetic factors are more significant than that of a single locus in common (or complex) diseases such as cardiovascular disease, cancer, and diabetes [2]. The detection of epistatic interactions therefore can help to improve pathogenesis, prevention, diagnosis and treatment of complex human diseases.

While the recent development of genome-wide association studies (GWAS) [3] and the International Hapmap project [4-5] has made it possible to identify common genetic variation or heritable risk factors in diseases from population-based data [6-8], the size of the genotyped data is typically very large and the number of combinations of all the genetic factors to be checked for the interactions are enormous, which cannot be exhaustively detected by experimental methods. Therefore, it is crucial to detect causal interacting genes or SNPs by heuristic computational methods.

Commonly-used methods for epistatic interactions detection can be roughly grouped into two categories: parametric statistical methods and machine learning methods. Parametric statistical methods include logistic regression [9], multifactor dimensionality reduction (MDR) methods [10-13], a stepwise-penalized logistic regression (stepPLR) [14], and a Bayesian epistasis association mapping (BEAM) method [15]. The most commonly used parametric statistical method for detecting epistasis is logistic regression [9]. However, logistic regression is inappropriate due to its overfitting problem since the number of parameters will be very large when the number of interacting genes of interest increases. Therefore, many other statistical methods have been developed to avoid this shortcoming. The

most famous method to identify epistatic interactions for binary outcomes is MDR (multifactor dimensionality reduction) [10-13]. MDR utilizes the ratio of the number of cases to the number of controls to reduce the dimensionality to one dimension and selects SNP combinations that have the highest prediction performance [14]. Recently, Park and Hastie proposed the stepwise-penalized logistic regression (stepPLR) to overcome the drawbacks of logistic regression and MDR [15]. They used quadratic penalization to avoid increasing the number of parameters to be estimated and a forward stepwise method to reduce the time complexity for detecting gene interactions. BEAM is a Bayesian marker partition model using Markov Chain Monte Carlo to reach an optimal marker partition and a new B statistic to check each marker or set of markers for significant associations [15]. Despite their success to some degrees, statistical methods can only be applied to small-scale analysis due to their computational complexity. For instance, MDR employs an exhaustive searching strategy to avoid local optima, which makes it impractical for large-scale datasets. StepPLR is also very time-consuming if the number of SNPs is larger than 100. Support vector machine-based approach [16] and random forest-based approach [17] are two commonly-used machine learning methods for epistatic interactions detection. In [16], Chen et al. proposed a support vector machine approach for detecting epistatic interactions based on RFE (recursive feature elimination), RFA (recursive feature addition) and GA (genetic algorithm) feature selection method. Jiang et al. adopted random forests, which is an ensemble learning technique, to the detection of epistatic interactions in case-control studies [17]. They first ranked SNPs based on gini importance of each SNP from random forests and then performed a greedy search for a small subset of SNPs that could minimize the classification error by a Sliding Window Sequential Forward feature Selection (SWSFS) algorithm. The common limitation of machine learning-based methods is that they might identify a SNP set that produces the highest classification accuracy, but not necessarily has the strongest association with the diseases. As a result, machine learning-based approaches tend to introduce many false positives, since the including of more SNPs increases classification accuracies.

In this paper, we address these problems by proposing a new and fast Markov Blanket method, FEPI-MB (Fast EPistatic Interactions detection using Markov Blanket), to detect epistatic interactions. The Markov Blanket is a minimal set of variables, which can completely shield the target variable from all other variables. Thus we can guarantee that the SNP set detected by Markov Blanket method has a strong association with diseases and contains fewest false positives. Furthermore, Markov Blanket method performs a heuristic search by calculating the association between variables to avoid the time-consuming training process as in SVMs and Random Forests. Some Markov Blanket methods take a divide-and-conquer approach that breaks the problem of identifying Markov Blanket of variable T (MB (T)) into two subproblems: First, identifying parents and children of T (PC (T)) and, second, identifying the parents of the children of T (spouse). The goal of epistatic interactions detection is to identify causal interacting genes or SNPs for some certain diseases and therefore it is a special application of Markov Blanket method because we only need to detect the parents of the target variable T (disease status labels). Our new Markov Blanket method makes some simplifications to adapt to this special condition.

We apply the FEPI-MB algorithm to simulated datasets based on four disease models and a real dataset (the Age-related Macular Degeneration (AMD) dataset). We demonstrate that the proposed method significantly outperforms other commonly-used methods and is capable of finding SNPs strongly associated with diseases. Comparing to other Markov Blanket learning methods, our method is faster and can still achieve a better performance.

The rest of our paper is organized as follows. In Section 2, we provide a brief introduction for Markov Blanket and several Markov Blanket learning methods. We also introduce an important method to test independence and conditional independence: G^2 test. In section 3, we describe the proposed new Markov Blanket method, FEPI-MB. In section 4, we present the results comparing the performance of FEPI-MB with other existing methods for epistatic interaction detection (BEAM, MDR and SVM) and one best Markov Blanket learning method (interIAMBnPC). Finally, we give our conclusion in section 5.

2. MARKOV BLANKET METHODS

2.1 Markov Blankets

Bayesian networks represent a joint probability distribution J over a set of random variables by a directed acyclic graph (DAG) G and encode the Markov condition property: each variable is conditionally independent of its nondescendants, given its parents in G [18]. In a Bayesian network, if the probability distribution of X conditioned on both Y and Z is equal to the probability distribution of X conditioned only on Y, i.e., $P(X \mid Y, Z) = P(X \mid Y)$, X is conditionally independent of Z given Y. This conditional independence is represented as $(X \perp Z \mid Y)$.

Definition 1 (Faithfulness). *A Bayesian network N and a joint probability distribution J are faithful to each other if and only if every conditional independence entailed by the DAG of N and the Markov Condition is also present in J[19] .*

Theorem 1. *If a Bayesian network N is faithful to a joint probability distribution J, then: (1) nodes X and Y are adjacent in N if and only if X and Y are conditionally dependent given any other set of nodes. (2) for the triplet of nodes X, Y , and Z in N, X and Z are adjacent to Y , but Z is not adjacent to X, $X \rightarrow Y \leftarrow Z$ is a subgraph of N if and only if X and Z are dependent conditioned on every other set of nodes that contains Y .*

We can define the Markov Blanket of a variable T, MB (T), as a minimal set for which $(X \perp T \mid MB(T))$, for all $X \in V - \{T\} - MB(T)$ where V is the variable set. The Markov Blanket of a variable T is a minimal set of variables, which can completely shield variable T from all other variables. All other variables are probabilistically independent of the variable T conditioned on the Markov Blanket of variable T.

Theorem 2. *If Bayesian network N is faithful to its corresponding joint probability distribution J, then for every variable T, MB(T) is unique and is the set of parents, children, and spouses of T.*

Theorem 1 and **Theorem 2** are proven in [20-21], separately. We show an example of the Markov Blanket in the well-known Asia network in Figure 1. The MB(T) of the node 'TBorCancer' is the set of gray-filled nodes.

Given the definition of a Markov Blanket, the probability distribution of T is completely determined by the values of variables in MB(T). Therefore, the detection of Markov Blanket can be applied for optimal variable selection. In addition, the Markov Blanket can be used for causal discovery because MB(T) contains direct cause variables (parents), direct effect variables (children), and direct cause variables (spouse) of direct effect variables of T.

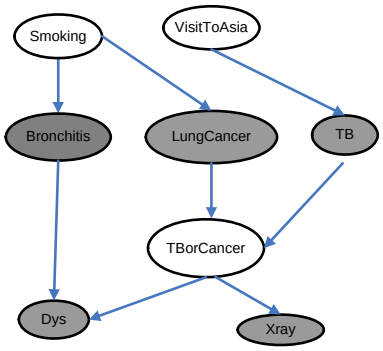


Figure 1. The Aisa Network. The gray-filled nodes are the MB(T) of node 'TBorCancer'.

As shown in Figure 2, genome-wide association studies try to identify the *k*-way interaction among disease SNPs: SNP1, SNP2,…,SNPk and exclude all other unrelated normal SNPs(SNPk+1,…,SNPn). Thus, the Markov Blanket learning method is suitable for detection of epistatic interactions in genome-wide case-control studies, e.g., to identify a minimal set of SNPs which may cause the disease and require further experiments. Meanwhile this detected minimal set of causal SNPs can shield the disease from all other normal SNPs to decrease the false positive rate and reduce the cost of future validation experiments.

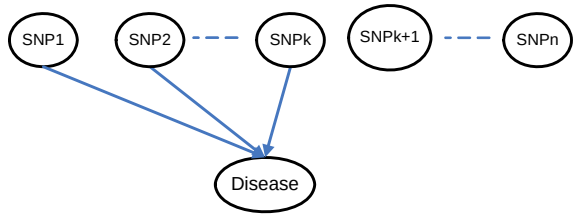


Figure 2. Example of genome-wide association studies (GWAS). The goal of genome-wide association studies is to identify the *k*-way interaction among disease SNPs: SNP1, SNP2,…,SNPk.

2.2 Markov Blankets Learning Methods

There are several Markov Blanket learning methods such as: Koller-Sahami (KS) algorithm [22], Grow-Shrink (GS) algorithm [23], Incremental association Markov Blanket (IAMB) algorithm [24], Max-Min Markov Blanket (MMMB) algorithm [25], HITON_MB [26] and PCMB [27].

Koller-Sahami (KS) algorithm is the first algorithm to employ Markov Blanket for feature selection. However, there is no theoretical guarantee for Koller-Sahami (KS) algorithm to find optimal MB set [22]. The GS algorithm [23] and IAMB methods [24] are two similar algorithms with two search procedures, forward and backward. In the forward phase, the nodes of MB(T) are admitted into MB, while in the backward phase false positives are removed from MB. Under the assumptions of faithfulness and correct independence test, both the GS algorithm and IAMB are proved correct [24]. Comparing to GS algorithm, IAMB might achieve a better performance with fewer false positives admitted during the forward phase. A common limitation for GS algorithm and IAMB is that both methods require a very large number of samples to perform well. IAMB can be revised in two ways: (1) After each admission step in forward phase, perform a backward conditioning phase to remove false positives to keep the size of MB(T) as small as possible. (2) Substitute the backward conditioning phase with the PC algorithm instead [19]. In other words, the backward phase will perform the independence test conditioned on all subsets of the current Markov Blanket. Tsamardinos et al. proposed three IAMB variants: interIAMB, IAMBnPC and InterIAMBnPC [24]. They also proved the correctness of InterIAMBnPC. The time complexity of IAMB is O(|MB|×N) where |MB| is the size of MB and N is number of variables.

To overcome the data inefficient problem of IAMB and its variants, Max-Min Markov Blanket (MMMB) algorithm [25], HITON_MB [26] and PCMB [27] are proposed. All these three algorithms take a divide-and-conquer method that breaks down the problem of identifying Markov Blanket of variable T into two subproblems: First, identifying parents and children of T (PC(T)) and, second, identifying the spouses of T. Meanwhile, they have the same two assumptions as IAMB (i.e. faithfulness and correct independence test) and take into account the graph topology to improve data efficiency. However, results from MMPC/MB and HITON-PC/MB are not always correct since some descendants of T other than its children will enter PC(T) during the first step of identifying parents and children of T [27]. PCMB can be proved correct in [27]. In every loop, PCMB first remove unrelated variables, then PCMB use IAMBnPC method to admit one feature and remove false positives. The problem of PCMB is that the PC algorithm performs an exhaustive conditional independence test, which is very time consuming. The reason that PC algorithm was used in PCMB and interIAMBnPC is that PC algorithm is a more sample-efficient method and is sound under the assumption of faithfulness [24]. In fact if the size of Markov Blanket is large, PC algorithm still needs a lot of samples to guarantee its performance. There is no theoretical proof and guarantee that the PC algorithm admits less false positives than other methods.

2.3 FEPI-MB: Method Description

Detecting epistatic interaction is a special application of Markov Blanket learning method because we only need detect the parents of the target variable T and don't need to design a complex algorithm to detect spouses of T. Here target variable T is the disease status labels and the parents of T are those disease SNPs. MB(T) only contains the parents of T.

In our FEPI-MB method, the G^2 test is used to test independence and conditional independence between two variables for discrete data [19, 28]. The null hypothesis for G^2 test is that two variables are independent.

Assume that we have a contingency table to record and analyze the joint distribution of two variables. The count in a particular cell in the contingency table, x_{ij} , is the value of a random variable from N samples with a multinomial distribution. Let $x_{i\bullet}$ represent the sum of elements in all cells along a row, and $x_{\bullet j}$ denote the sum of the counts in all cells along a column. The expected value of the random variable x_{ij} is:

$$E(x_{ij}) = \frac{x_{i\bullet} x_{\bullet j}}{N} \quad (1)$$

under the null hypothesis.

We can compute the conditional independence from appropriate marginal distributions in a similar way. For instance, to determine whether the first variable is independent of the second conditioned on the third, we can calculate the expected value of a cell x_{ijk} as

$$E(x_{ijk}) = \frac{x_{i\bullet k} x_{\bullet jk}}{x_{\bullet\bullet k}} \quad (2)$$

For n cells in a contingency table, assume that the observed numbers are denoted by O_1, O_2, …, O_n and the corresponding expected numbers by E_1, E_2, …, E_n, then, the G^2 is given by

$$G^2 = 2\sum_i^n O_i \ln(\frac{O_i}{E_i}) \quad (3)$$

which has an asymptotical distribution as chi-square (χ^2) with appropriate degrees of freedom. The degrees of freedom (df) for the G^2 test between two variables A and B can be calculated as:

$$df = (Cat(A)-1)\times(Cat(B)-1) \quad (4)$$

and the degrees of freedom (df) for the G^2 test between A and B conditional on the third variable C can be calculated as:

$$df = (Cat(A)-1)\times(Cat(B)-1)\times\prod_{i=1}^{n} Cat(C_i) \quad (5)$$

where Cat(X) is the number of categories of the variable X and n is the number of variables in C. P-values from G^2 test reflect the statistical significance of association/dependence between variables.

The proposed FEPI-MB uses G^2 to test the association and independence between SNPs and disease status. The detail of FEPI-MB is shown in Fig. 2.

/*Initialization*/
V : set of all variables; T: Target variables;
MB(T)= ϕ ;
canMB=V-{T};
/* our algorithm*/
Begin procedure
Repeat
Remove-MB;
Forward-MB;
Backward-MB;
Until MB(T) has not changed;
End procedure
/* Remove phase */
Begin *Remove-MB*
For all $x_i \in$ canMB;
$g(x_i) = G^2(x_i : T \mid MB(T))$;
If $(x_i \perp T \mid MB(T))$
$canMB = canMB - x_i$;
End If
End For
End
/* Forward phase */
Begin *Forward-MB*
X = argmax ($g(x_i)$) $x_i \in$ canMB ;
If $(X \perp T \mid MB(T))$
$MB(T) = MB(T) \cup \{X\}$;
canMB=canMB-X;
End If
End
/Backward phase/
Begin *Backward-MB*
For all $Y \in MB(T)$
If $(Y \perp T \mid MB(T) - Y)$
$MB(T) = MB(T) - \{Y\}$;
End If
End For
End

Figure 3. FEPI-MB algorithm

The FEPI-MB consists of three phases: *Remove*-MB, *Forward*-MB and *Backward*-MB. During the phase of *Remove*-MB, unrelated variables are removed from the candidate set for Markov Blanket (canMB) based on the conditional independence test. This will reduce the searching space after each iteration and can help to decrease the computational complexity. After the phase of *Remove*-MB, the variable which has the maximal G^2 score and is associated with the target variable T in canMB enters MB(T) in the phase of *Forward*-MB, where false positives are removed during the phase of *Backward*-MB. Comparing to PCMB, we get rid of the time-consuming PC algorithm and use the maximal subset of current MB(T) to perform the conditional independence test in the phase of *Backward*-MB. The time complexity of FEPI-MB is less than the O(|MB|×N) of IAMB because in each iteration after the first iteration the number of conditional independence tests performed in the phase of *Remove*-MB is less than N. The optimal time complexity of FEPI-MB is O(N). Like IAMB and PCMB, the soundness of FEPI-MB is based on the assumptions of DAG-faithfulness and correct independence test.

3. EXPERIMENTAL RESULTS

3.1 Results on Simulated Data

We first evaluate the proposed FEPI-MB on simulated data sets, which are generated from three commonly used two-locus epistatic models in [9, 17] and one three-locus epistatic model developed in [17]. Table 1 lists the disease odds for these four epistatic models, where α is the baseline effect and θ is the genotypic effect. Assume an individual has genotype g_A at locus A and genotype g_B at locus B in a two-locus epistatic model, then the disease odds are defined as

$$p(D \mid g_A, g_B) / p(\overline{D} \mid g_A, g_B) \qquad (6)$$

where $p(D \mid g_A, g_B)$ is the probability that an individual has the disease given genotype (g_A, g_B) and $p(\overline{D} \mid g_A, g_B)$ is the probability that an individual does not have the disease given genotype (g_A, g_B).

In Model1 the odds of disease increase in a multiplicative mode both within and between two loci. For example, an individual with *Aa* at locus A has larger odds which are $1+\theta$ times relative to those of an individual who is homozygous *AA*; the *aa* homozygote has further increased disease odds by $(1+\theta)^2$. We can also find similar effects on locus B. Finally the odds of disease for each combination of genotypes at loci A and B can be obtained by the product of the two within-locus effects. Model2 demonstrates two-locus interaction multiplicative effects because at least one disease-associated allele must be present at each locus to increase the odds beyond the baseline level. Moreover the increment of the disease-associated allele at loci A or B can further increase the disease odds by the multiplicative factor $1+\theta$. Model3 specifies two-locus interaction threshold effects. Like Model 2, Model3 also requires at least one copy of the disease-associated alleles at both loci A and B. However the increment of the disease-associated allele does not increase the risk further. We call this as disease threshold effect. It means a single copy of the disease-associated allele at each locus is required to increase odds of disease and this is the disease threshold. But after the disease threshold has already been met, having both copies of the disease-associated allele at either locus has no additional influence on disease odds. There are three disease loci in model 4. Some certain genotype combinations can increase disease risk and there are almost no marginal effects for each disease locus. Model 4 is more complex than Model 1, 2 and 3. All these four models are non-additive models and they differ in the way that the number of disease-associated allele increases the odds of disease.

Table 1. Disease odds for four epistatic models

Model 1	**AA**	**Aa**	**aa**
BB	α	$\alpha(1+\theta)$	$\alpha(1+\theta)^2$
Bb	$\alpha(1+\theta)$	$\alpha(1+\theta)^2$	$\alpha(1+\theta)^3$
bb	$\alpha(1+\theta)^2$	$\alpha(1+\theta)^3$	$\alpha(1+\theta)^4$
Model 2	**AA**	**Aa**	**aa**
BB	α	α	α
Bb	α	$\alpha(1+\theta)$	$\alpha(1+\theta)^2$
bb	α	$\alpha(1+\theta)^2$	$\alpha(1+\theta)^4$
Model 3	**AA**	**Aa**	**aa**
BB	α	α	α
Bb	α	$\alpha(1+\theta)$	$\alpha(1+\theta)$
bb	α	$\alpha(1+\theta)$	$\alpha(1+\theta)$
Model4	**AA**		
	BB	**Bb**	**bb**
CC	α	α	α
Cc	α	α	$\alpha(1+\theta)$
cc	α	$\alpha(1+\theta)$	α
	Aa		
	BB	Bb	bb
CC	α	α	$\alpha(1+\theta)$
Cc	α	$\alpha(1+\theta)$	α
cc	$\alpha(1+\theta)$	α	α
	aa		
	BB	Bb	bb
CC	α	$\alpha(1+\theta)$	α
Cc	$\alpha(1+\theta)$	α	α
cc	α	α	α

The prevalence of a disease is the proportion the total number of cases of the disease in the population and we assume that the disease prevalence is 0.1 for all these four disease models [9].

To generate data, we need to determine three parameters associated with each model: the marginal effect of each disease locus (λ), the minor allele frequencies (MAF) of all disease loci, and the strength of linkage disequilibrium (LD) between the unobserved disease locus and a genotyped locus [9]. LD is a nonrandom association of alleles at different loci and is quantified by the squared correlation coefficient r^2 calculated from allele frequencies [9]. In this paper, we set λ equal to 0.3, 0.3, and 0.6 for models 1, 2, and 3, respectively. For model 4, we set $\theta = 7$ arbitrarily because there are almost no marginal effects in model 4.We let MAF take four values (0.05, 0.1, 0.2, and 0.5) and let r^2 take two values (0.7, 1.0) for each model. For each non-disease marker we randomly chose its MAF from a uniform distribution in [0.0. 0.5]. We first generate 50 datasets and each contains 100 markers genotyped for 1,000 cases and 1,000 controls based on each parameter setting for each model. To test the scalability of FEPI-MB, we also generate 50 larger datasets and each contains 500 markers genotyped for 2,000 cases and 2,000 controls using the same parameter setting for each model.

We compare the FEPI-MB algorithm with three commonly-used methods: BEAM, Support Vector Machine and MDR on the four simulated disease models. To measure the performance of each method, we use "power" as the criterion function. Power is calculated as the fraction of 50 simulated datasets in which two disease associated markers are identified and demonstrate statistically significant associations (G^2 test p-values below a threshold) with the disease [9-17].

BEAM uses a Bayesian marker partition model to partition SNPs into three groups: group 0 contains markers unlinked to the disease, group 1 contains markers contributing independently to the disease, and group 2 contains markers jointly influence the disease. After the partition step by MCMC, candidate SNPs or groups of SNPs are further filtered by the B statistic [15]. The BEAM software is downloaded from http://www.fas.harv-ard.edu/~junliu/BEAM.

For support vector machines, we use LIBSVM with a RBF kernel to detect gene-gene interactions [29]. A grid search is used for selecting optimal parameters. Instead of using the exhaustive greedy search strategy for SNPs as in [16], which is very time-consuming and infeasible to large-scale datasets, we turn to a search strategy used in [17]. First we rank SNPs based on the mutual information between SNPs and disease status label which is 0 for the control and 1 for the case. Then, we use a sliding window sequential forward feature selection (SWSFS) algorithm in [17] based on SNPs rank. The window size in SWSFS algorithm determines how robust the algorithm could be and we set it to 20.

Since MDR algorithm can not be applied to a large dataset directly, we first reduce the number of SNPs to 10 by ReliefF [30], a commonly-used feature selection algorithm, and then MDR performs an exhaustive search for a model consisting of no more than four SNPs that can maximize cross-validation consistency and prediction accuracy. When one model has the maximal cross-validation consistency and another model has the maximal prediction accuracy, MDR follows statistical parsimony (selects the model with fewer SNPs).

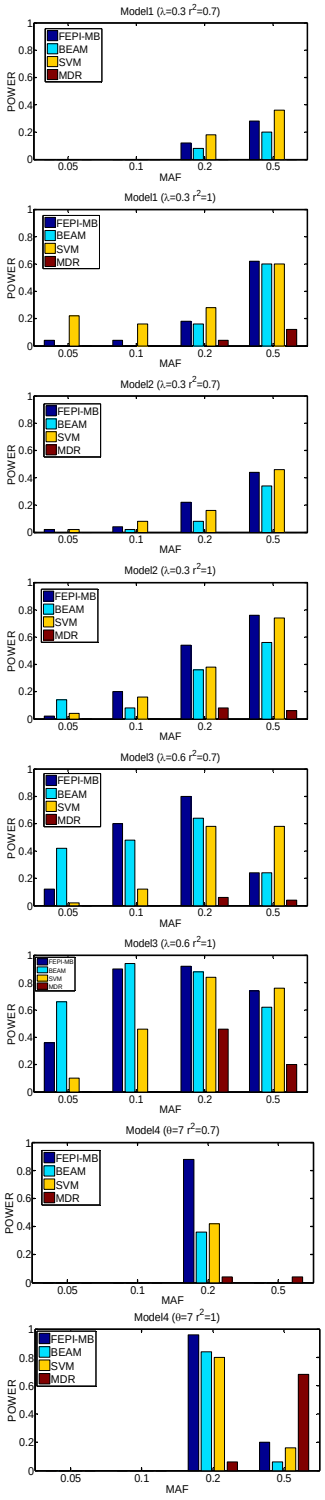


Figure 4. Performance comparison for small datasets containing 100 markers genotyped from 1000 cases and 1000 controls.

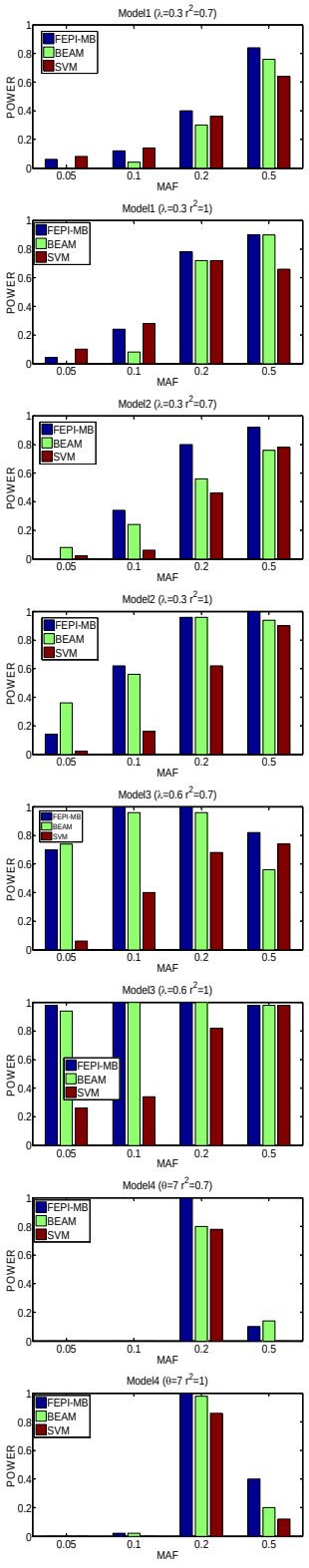


Figure 5. Performance comparison for large datasets containing 500 markers genotyped from 2000 cases and 2000 controls.

For the large datasets containing 500 markers genotyped for 2,000 cases and 2,000 controls, we only compare the performance of FEPI-MB, BEAM and SVM because ReliefF [30] in MDR can not work for large datasets of this scale.

The results on the simulated data are shown in Figure 4 and Figure 5. As can be seen, among the four methods, the FEPI-MB algorithm performs the best. BEAM is the second best. Interestingly, BEAM prefers to assign the two disease-associated markers to group 1 for model1, model2 and model3, which means that BEAM considers that the two disease SNPs affect the disease independently. In most cases, the powers of MDR are much smaller than these of the FEPI-MB and BEAM algorithms. For the MDR algorithm, the poor performance may be due to the use of ReliefF to reduce SNPs from a very large dimensionality. In some cases, SVM can achieve a comparable or even better performance than FEPI-MB and BEAM, however, at the cost of introducing more false positives. Figure 5 also demonstrates the scalability of FEPI-MB on large datasets.

An important issue for epistatic interaction detection in genome-wide association studies is the number of samples. Typically, the size of samples is limited and consequently, computational model behaves differently. We explore the effect of the number of samples on the performance of BEAM and FEPI-MB (SVMs will always introduce a large number of false positives and thus, is not compared here). We generate synthetic datasets containing 40 markers genotyped for different number of cases and controls for model 3 with $\lambda = 0.6$, $r^2 = 1$ and MAF=0.5. The result is shown in Figure 6 and we find that FEPI-MB can achieve a higher power than BEAM when the number of samples is the same. On the other hand, FEPI-MB needs fewer samples to reach the perfect power comparing to BEAM. So we can conclude that FEPI-MB is more sample-efficient than BEAM.

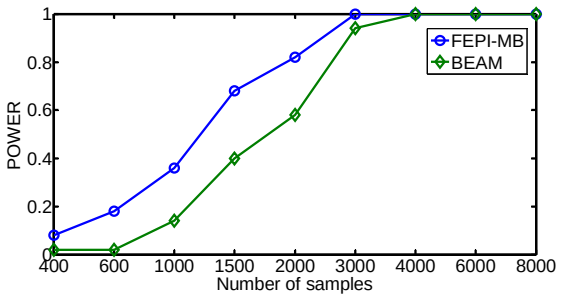


Figure 6. Effect of number of samples on the performance of FEPI-MB and BEAM

We also compare the performance of FEPI-MB with interIAMBnPC based on the large dataset from model 1 to show the time efficiency of FEPI-MB. Among the three variants of IAMB, interIAMBnPC can achieve the best performance [24]. Both FEPI-MB and interIAMBnPC are written in MATLAB and all the experiments are run on an Intel Core 2 Duo T6600 2.20 GHz, 4GB RAM and Windows Vista. The results are shown in Table 2. As seen, FEPI-MB runs more than ten times faster than interIAMBnPC.

Table 2. Comparison of performance of FEPI-MB and interIAMBnPC for the large datasets of Model1.

Model	λ	r^2	MAF	Algorithm	Power	Average Time(s)
1	0.3	0.7	0.05	FEPI-MB	3	0.4574
				interIAMBnPC	3	7.5505
			0.1	FEPI-MB	6	0.4437
				interIAMBnPC	5	9.2449
			0.2	FEPI-MB	20	0.4436
				interIAMBnPC	20	9.4295
			0.5	FEPI-MB	42	0.4449
				interIAMBnPC	42	8.2823
		1	0.05	FEPI-MB	2	0.4393
				interIAMBnPC	2	7.3610
			0.1	FEPI-MB	12	0.4421
				interIAMBnPC	12	9.7156
			0.2	FEPI-MB	39	0.4431
				interIAMBnPC	38	9.6498
			0.5	FEPI-MB	45	0.4449
				interIAMBnPC	43	9.1229

3.2 Results on Real Data

From the results on simulated data with 100 SNPs or 500 SNPs, FEPI-MB demonstrates a better performance than three other methods. Notice that a real genome-wide case-control association study may require genotyping of 30,000–1,000,000 common SNPs. In this section, we show that FEPI-MB algorithm can also handle large-scale datasets in real genome-wide case-control studies. We consider an Age-related Macular Degeneration (AMD) dataset, which contains 116,204 SNPs genotyped with 96 cases and 50 controls [8]. AMD (OMIM 603075) [31] is a common genetic disease related to the progressive visual dysfunction in age over 70 in the developed country. A GWA study was successfully conducted on this disease finding two associated SNPs, rs380390 and rs1329428 ('rs': assigned reference SNP ID by dbSNP [32]) in non-coding region of the gene for complement factor H (*CFH*), which is located on chromosome 1 in a region linked to AMD [8].

In the phase of preprocessing data, we remove non-polymorphic SNPs and those that significantly deviated from Hardy-Weinberg Equilibrium (HWE) [8]. We also remove all SNPs that have more than five missing genotypes. For the remaining SNPs with less than five missing genotypes, we estimate every single SNP's missing genotyping data from that SNP's observed genotyping distribution. After filtering, there are 97,327 SNPs lying in 22 autosomal chromosomes remained.

The searching time of FEPI-MB for AMD-related SNPs on an Intel Core 2 Duo T6600 2.20 GHz, 4GB RAM and Windows Vista is 96.4s and FEPI-MB detects two associated SNPs: rs380390 and rs2402053, which have a G^2 test p-value of $5.36*10^{-10}$. The first SNP, rs380390, is already found in [8] with a significant association with AMD. The other SNP detected by the FEPI-MB algorithm is SNP rs2402053, which is integenic between TFEC and TES in chromosome 7q31. TES, which is also called as TESTIN, is reported at OMIM (OMIM 606085) and increasing expression of TES can reduce growth potential profoundly [33]. Although no evidences were reported with this gene related to AMD in the literature, it may be a plausible candidate gene associated with AMD.

4. CONCLUSION

While many computational methods were used for identification of epistatic interactions, most existing computational methods do not consider the complexity of genetic mechanisms causing common diseases and only focus on the selection of SNP sets, which show the best classification capacity. This will introduce many false positives inevitably. Furthermore, most existing methods cannot directly handle genome-wide scale problems. In this paper, we introduce a new and fast Markov Blanket-based method, FEPI-MB, to identify epistatic interactions. We compared FEPI-MB with three other methods, BEAM, Support Vector Machine and MDR, over both simulated datasets and a real dataset. Our results show that the FEPI-MB algorithm outperforms other methods in terms of the power and sample-efficiency. It can identify a minimal set of SNPs associated with diseases, which contains less false positives. This is critical in saving the potential costs of biological experiments. Moreover, we compare FEPI-MB with one of the best Markov Blanket learning method, interIAMBnPC. The performance of FEPI-MB is a little better than interIAMBnPC while it is more than ten times faster than interIAMBnPC.

5. ACKNOWLEDGMENTS

This work is supported by the US National Science Foundation Award IIS-0644366.

6. REFERENCES

[1] McKinney, B. A., Reif, D. M., Ritchie, M. D. and Moore, J. H. Machine learning for detecting gene-gene interactions: a review. Applied bioinformatics, 5, 2 2006), 77-88.

[2] Moore, J. H. The ubiquitous nature of epistasis in determining susceptibility to common human diseases. Human heredity, 56, 1-3 2003), 73-82.

[3] Hirschhorn, J. N. and Daly, M. J. Genome-wide association studies for common diseases and complex traits. Nat Rev Genet, 6, 2 (Feb 2005), 95-108.

[4] The International HapMap Project. Nature, 426, 6968 (Dec 18 2003), 789-796.

[5] A haplotype map of the human genome. Nature, 437, 7063 (Oct 27 2005), 1299-1320.

[6] Easton, D. F., Pooley, K. A., Dunning, A. M., Pharoah, P. D., Thompson, D., Ballinger, D. G., Struewing, J. P., Morrison, J., Field, H., Luben, R., Wareham, N., Ahmed, S., Healey, C. S., Bowman, R., Meyer, K. B., Haiman, C. A., Kolonel, L. K., Henderson, B. E., Le Marchand, L., Brennan, P., Sangrajrang, S., Gaborieau, V., Odefrey, F., Shen, C. Y., Wu, P. E., Wang, H. C., Eccles, D., Evans, D. G., Peto, J., Fletcher, O., Johnson, N., Seal, S., Stratton, M. R., Rahman, N., Chenevix-Trench, G., Bojesen, S. E., Nordestgaard, B. G., Axelsson, C. K., Garcia-Closas, M., Brinton, L., Chanock, S., Lissowska, J., Peplonska, B.,

Nevanlinna, H., Fagerholm, R., Eerola, H., Kang, D., Yoo, K. Y., Noh, D. Y., Ahn, S. H., Hunter, D. J., Hankinson, S. E., Cox, D. G., Hall, P., Wedren, S., Liu, J., Low, Y. L., Bogdanova, N., Schurmann, P., Dork, T., Tollenaar, R. A., Jacobi, C. E., Devilee, P., Klijn, J. G., Sigurdson, A. J., Doody, M. M., Alexander, B. H., Zhang, J., Cox, A., Brock, I. W., MacPherson, G., Reed, M. W., Couch, F. J., Goode, E. L., Olson, J. E., Meijers-Heijboer, H., van den Ouweland, A., Uitterlinden, A., Rivadeneira, F., Milne, R. L., Ribas, G., Gonzalez-Neira, A., Benitez, J., Hopper, J. L., McCredie, M., Southey, M., Giles, G. G., Schroen, C., Justenhoven, C., Brauch, H., Hamann, U., Ko, Y. D., Spurdle, A. B., Beesley, J., Chen, X., Mannermaa, A., Kosma, V. M., Kataja, V., Hartikainen, J., Day, N. E., Cox, D. R. and Ponder, B. A. Genome-wide association study identifies novel breast cancer susceptibility loci. Nature, 447, 7148 (Jun 28 2007), 1087-1093.

[7] Fellay, J., Shianna, K. V., Ge, D., Colombo, S., Ledergerber, B., Weale, M., Zhang, K., Gumbs, C., Castagna, A., Cossarizza, A., Cozzi-Lepri, A., De Luca, A., Easterbrook, P., Francioli, P., Mallal, S., Martinez-Picado, J., Miro, J. M., Obel, N., Smith, J. P., Wyniger, J., Descombes, P., Antonarakis, S. E., Letvin, N. L., McMichael, A. J., Haynes, B. F., Telenti, A. and Goldstein, D. B. A whole-genome association study of major determinants for host control of HIV-1. Science (New York, N.Y, 317, 5840 (Aug 17 2007), 944-947.

[8] Klein, R. J., Zeiss, C., Chew, E. Y., Tsai, J. Y., Sackler, R. S., Haynes, C., Henning, A. K., SanGiovanni, J. P., Mane, S. M., Mayne, S. T., Bracken, M. B., Ferris, F. L., Ott, J., Barnstable, C. and Hoh, J. Complement factor H polymorphism in age-related macular degeneration. Science (New York, N.Y, 308, 5720 (Apr 15 2005), 385-389.

[9] Marchini, J., Donnelly, P. and Cardon, L. R. Genome-wide strategies for detecting multiple loci that influence complex diseases. Nature genetics, 37, 4 (Apr 2005), 413-417.

[10] Hahn, L. W., Ritchie, M. D. and Moore, J. H. Multifactor dimensionality reduction software for detecting gene-gene and gene-environment interactions. Bioinformatics (Oxford, England), 19, 3 (Feb 12 2003), 376-382.

[11] Moore, J. H., Gilbert, J. C., Tsai, C. T., Chiang, F. T., Holden, T., Barney, N. and White, B. C. A flexible computational framework for detecting, characterizing, and interpreting statistical patterns of epistasis in genetic studies of human disease susceptibility. Journal of theoretical biology, 241, 2 (Jul 21 2006), 252-261.

[12] Ritchie, M. D., Hahn, L. W. and Moore, J. H. Power of multifactor dimensionality reduction for detecting gene-gene interactions in the presence of genotyping error, missing data, phenocopy, and genetic heterogeneity. Genetic epidemiology, 24, 2 (Feb 2003), 150-157.

[13] Ritchie, M. D., Hahn, L. W., Roodi, N., Bailey, L. R., Dupont, W. D., Parl, F. F. and Moore, J. H. Multifactor-dimensionality reduction reveals high-order interactions among estrogen-metabolism genes in sporadic breast cancer. American journal of human genetics, 69, 1 (Jul 2001), 138-147.

[14] Park, M. Y. and Hastie, T. Penalized logistic regression for detecting gene interactions. Biostatistics (Oxford, England), 9, 1 (Jan 2008), 30-50.

[15] Zhang, Y. and Liu, J. S. Bayesian inference of epistatic interactions in case-control studies. Nature genetics, 39, 9 (Sep 2007), 1167-1173.

[16] Chen, S. H., Sun, J., Dimitrov, L., Turner, A. R., Adams, T. S., Meyers, D. A., Chang, B. L., Zheng, S. L., Gronberg, H., Xu, J. and Hsu, F. C. A support vector machine approach for detecting gene-gene interaction. Genetic epidemiology, 32, 2 (Feb 2008), 152-167.

[17] Jiang, R., Tang, W., Wu, X. and Fu, W. A random forest approach to the detection of epistatic interactions in case-control studies. BMC bioinformatics, 10 Suppl 12009), S65.

[18] Chen, X.-W., Anantha, G. and Lin, X. Improving Bayesian Network Structure Learning with Mutual Information-Based Node Ordering in the K2 Algorithm. IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, 20, 5 (May 2008), 628-640.

[19] Spirtes, P., Glymour, C. N. and Scheines, R. Causation, prediction, and search. MIT Press, Cambridge, Mass., 2000.

[20] Pearl, J. Probabilistic reasoning in intelligent systems : networks of plausible inference. Morgan Kaufmann Publishers, San Mateo, Calif., 1988.

[21] Tsamardinos, I. and Aliferis, C. Towards Principled Feature Selection: Relevancy, Filters and Wrappers. In Proceedings of the Ninth International Workshop on Artificial Intelligence and Statistics (Key West, Florida, January 3-6, 2003). Morgan Kaufmann, San Fransisco, CA.

[22] Koller, D. and Sahami, M. Toward Optimal Feature Selection. In Proceedings of the 13th conference on machine learning (Bari, Italy, July 3-6th, 1996). Morgan Kaufmann, San Fransisco, CA, 284-292.

[23] Margaritis, D. and Thrun, S. Bayesian Network Induction via Local Neighborhoods. In Proceedings of the Neural Information Processing Systems 12 (Denver, Colorado, USA, November 29 - December 4, 1999). MIT Press, Cambridge, MA, 505-511.

[24] Tsamardinos, I., Aliferis, C., Statnikov, A. and Statnikov, E. Algorithms for Large Scale Markov Blanket Discovery. In Proceedings of the The 16th International FLAIRS Conference (St. Augustine, FL, May 11-15, 2003). AAAI Press, Menlo Park, CA, 376-380.

[25] Tsamardinos, I., Aliferis, C. and Statnikov, A. Time and Sample Efficient Discovery of Markov Blankets And Direct Causal Relations. In Proceedings of the The ninth ACM SIGKDD international conference on Knowledge discovery and data mining (Washington, D.C., August 24-27, 2003). ACM, New York, NY, 673-678.

[26] Aliferis, C. F., Tsamardinos, I. and Statnikov, A. HITON: a novel Markov Blanket algorithm for optimal variable selection. AMIA ... Annual Symposium proceedings / AMIA Symposium2003), 21-25.

[27] Peña, J. M., Nilsson, R., Björkegren, J. and Tegnér, J. Towards scalable and data efficient learning of Markov boundaries. International Journal of Approximate Reasoning 45, 2 2006), 211-232.

[28] Sokal, R. R. and Rohlf, F. J. Biometry : the principles and practice of statistics in biological research. Freeman, New York, 1995.

[29] Chang, C.-c. and Lin, C.-J. LIBSVM: A library for support vector machines. City, 2001.

[30] Robnik-Šikonja, M. and Kononenko, I. Theoretical and empirical analysis of ReliefF and RReliefF. Machine learning, 53, 1 2003), 23-69.

[31] Hamosh, A., Scott, A. F., Amberger, J., Bocchini, C., Valle, D. and McKusick, V. A. Online Mendelian Inheritance in Man (OMIM), a knowledgebase of human genes and genetic disorders. Nucleic Acids Res, 30, 1 (Jan 1 2002), 52-55.

[32] Sherry, S. T., Ward, M. H., Kholodov, M., Baker, J., Phan, L., Smigielski, E. M. and Sirotkin, K. dbSNP: the NCBI database of genetic variation. Nucleic Acids Res, 29, 1 (Jan 1 2001), 308-311.
[33] Tobias, E. S., Hurlstone, A. F., MacKenzie, E., McFarlane, R. and Black, D. M. The TES gene at 7q31.1 is methylated in tumours and encodes a novel growth-suppressing LIM domain protein. Oncogene, 20, 22 (May 17 2001), 2844-2853.

Combining Active Learning and Semi-supervised Learning Techniques to Extract Protein Interaction Sentences

Min Song
Information Systems, New Jersey Institute of Technology
973-596-5291
Min.song@njit.edu

Hwanjo Yu
CSE Department, POSTECH
Pohang, South Korea
hwanjoyu@postech.ac.kr

Wook-Shin Han
CE Department, Kyungpook National University
Daegu, South Korea
wshan@knu.ac.kr

ABSTRACT

Protein-protein interaction (PPI) extraction has been a focal point of many biomedical research and database curation tools. Both Active Learning and Semi-supervised SVMs have recently been applied to extract PPI automatically. In this paper, we explore integrating active learning approaches to semi-supervised SVMs with a NLP-driven feature selection technique. Our contributions in this paper are as follows: (a) We proposed a novel PPI extraction technique called PPISpotter by combining an active learning technique with semi-supervised SVMs to extract protein-protein interaction. (b) We extracted a comprehensive set of features from MEDLINE records by Natural Language Processing (NLP) techniques for SVM classifiers. (c) We conducted experiments with three different PPI corpora and showed that PPISpotter is superior to four other comparison techniques in terms of precision, recall, and F-measure.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]:Data mining; H.3.3 [**Information Systems**]:Information Search and Retrieval—*Information filtering, Query formulation, Retrieval models, Search process*; I.2.6 [**Artificial Intelligence**]:Learning—*Knowledge acquisition*

General Terms

Algorithms, Design, Experimentation, Theory

Keywords

Biomedical Text Mining, Information Extraction, Active Learning, Semi-supervised Learning, Protein-protein interaction

1. INTRODUCTION

Automated protein-protein interaction (PPI) extraction from unstructured text collections is a task of significant interest in the bio-literature mining field. The most commonly addressed problem has been the extraction of binary interactions, where the system identifies which protein pairs in a sentence have a biologically relevant relationship between them. Proposed solutions include both hand-crafted rule-based systems and machine learning approaches [4]. Recently Semi-supervised Learning (SSL) techniques have been applied to PPI tasks [34]. SSL is a Machine Learning (ML) approach that combines supervised and unsupervised learning where typically a small amount of labeled and a large amount of unlabeled data are used for training. SSL has gained significant attention to PPI extraction because of two reasons. First, labeling of a large set of instances is labor-intensive and time-consuming. This task has to be also carried out by qualified experts and thus is expensive. Second, several studies show that using unlabeled data for learning improves the accuracy of classifiers [2, 27].

One major problem of SSL is that it may introduce incorrect labels to the training data, as the labeling is done by machine, and such labeling errors are critical to the classification performance. Active Learning (AL) can complement the SSL by reducing such labeling errors [32]. AL is a technique of selecting a small sample from the unlabeled data such that labeling on the sample maximizes the learning accuracy. The selected sample is manually labeled by experts. In this paper, we explore combining the AL with the SSL to improve the performance of the PPI task. To our best knowledge, this is the first attempt to apply a combination of semi-supervised and active learning for the extraction task of protein-protein interaction.

The contributions of this paper are three fold. First, we proposed a novel PPI extraction technique called PPISpotter by combining Deterministic Annealing-based SSL and an AL technique to extract protein-protein interaction. Second, we extracted a comprehensive set of features from MEDLINE records by Natural Language Processing (NLP) techniques, which further improve the SVM classifiers. In our feature selection technique, syntactic, semantic, and lexical properties of text are incorporated into feature selection that boosts the system performance significantly. Third, we conducted experiments with three different PPI corpuses and showed that PPISpotter is superior to other techniques by precision, recall, and F-measure.

The rest of the paper is organized as follows. Section 2 discusses related work. Section 3 briefly describes our approach and how it can be applied to extract protein—protein interactions from the biomedical literature. Section 4 describes active learning techniques and Section 5 explains semi-supervised learning techniques. Section 6 describes our feature selection method. Experimental results are discussed in Section 7. Section 8 reports and discuss the experimental results. Finally, Section 9 concludes the paper.

2. RELATED WORK

Many approaches have been proposed to extract protein-protein interaction from unstructured text. One approach employs pre-specified patterns and rules for PPI extraction [26]. However, this approach is often inapplicable to complex cases not covered by the pre-defined patterns and rules. Huang et al. [14] proposed a method where patterns are discovered automatically from a set of sentences by dynamic programming.

The second approach utilizes dictionary. Blaschke et al. [3] extracted protein-protein interactions based on co-occurrence of the form "… p1…I1… p2" within a sentence, where p1, p2 are proteins and I1 is an interaction term. Protein names and interaction terms (e.g., activate, bind, inhibit) are provided as a "dictionary." Pustejovsky et al. [23] extracted an "inhibit" relation for the gene entity from MEDLINE. Jenssen et al. [15] extracted gene-gene relations based on co-occurrence of the form "… g1…g2…" within a MEDLINE abstracts, where g1 and g2 are gene names. Gene names were provided as a "dictionary", harvested from HUGO, LocusLink, and other sources. Although their study uses 13,712 named human genes and millions of MEDLINE abstracts, no extensive quantitative results are reported and analyzed. Friedman et al. [12] extracted a pathway relation for various biological entities from a variety of articles.

The third approach is based on machine learning techniques. Bunescu et al. [4] conducted protein/protein interaction identification with several learning methods such as pattern matching rule induction (RAPIER), boosted wrapper induction (BWI), and extraction using longest common subsequences (ELCS). ELCS automatically learns rules for extracting protein interactions using a bottom-up approach. They conducted experiments in two ways; one with manually crafted protein names and the other with the extracted protein names by their name identification method. In both experiments, Zhou et al. [35] proposed two novel semi-supervised learning approaches, one based on classification and the other based on expectation-maximization, to train the HVS model from both annotated and un-annotated corpora. Song et al. [30] utilized syntactical, as well as semantic cues, of input sentences. By combining the text chunking technique and Mixture Hidden Markov Models, They took advantage of sentence structures and patterns embedded in plain English sentences. Temkin and Gilder [31] used a full parser with a lexical analyzer and a context free grammar (CFG) to extract protein-protein interaction from text. Alternatively, Yakushiji et al. [33] propose a system based on head-driven phrase structure grammar (HPSG). In their system protein interaction expressions are presented as predicate argument structure patterns from the HPSG parser. These parsing approaches consider only syntactic properties of the sentences and do not take into account semantic properties. Thus, although they are complicated and require many resources, their performance is not satisfactory. Mitsumori et al. [22] used SVM to extract protein-protein interactions. They use bag-of-words features, specifically the words around the protein names. These systems do not use any syntactic or semantic information. Miyao et al. [21] conducted a comparative evaluation of several state-of-the-art natural language parsers, focusing on the task of extracting protein–protein interaction (PPI) from biomedical papers. They found marginal difference in terms of accuracy but more significant differences in parsing speed. BioPPISVMExtractor is a recent PPI extraction system developed with SVM [34]. It utilizes rich feature sets such as word features, keyword feature, protein names distance feature, and Link Grammar extraction results for protein-protein interaction extraction. They observed that the rich feature sets help improve recall at the cost of a moderate decline in precision.

Cui et al. [10] applied an uncertainty sampling based method of active learning for a lexical feature-based SVM model to tag the most informative unlabeled samples. They reported that the performance of the active learning-based technique on AIMED and CB corpora was significantly improved in terms of reduction of labeling cost.

3. PPISPOTTER ARCHITECTURE

In this section, we describe the overall architecture and procedures of PPISpotter (Figure 1).

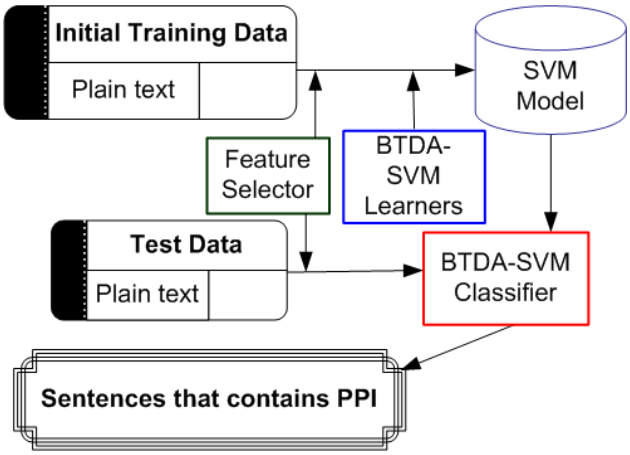


Figure 1: System Architecture of PPISpotter

PPISpotter incorporates AL models into SSL SVMs for extraction of protein-protein interaction. PPISpotter also automatically converts a sentence into 9 feature sets based on the technique described in Section 4.

Below is a set of steps that PPISpotter processes.

Step 1: Preprocess the initial training data. The feature selector applies the feature selection technique proposed in Section 4 to the preprocessed data sets.

Step 2: Train the model. Two classifiers, Break Tie-based SVM (BT-SVM) and Deterministic Annealing-based SVM (DA-SVM) classifiers are combined to train the model (a.k.a. BTDA-SVM). Figure 2 illustrates how to combine these two techniques (Blue dot line is the BT-SVM procedure and red solid line is the DA-SVM procedure). At this stage, the human expert provides feedback to the system for a set of instances in the fuzzy unlabeled data. Note that the BT-SVM classifier is based on the Break Tie active learning approach and DA-SVM classifier is based on the Deterministic Annealing technique.

Step 3: Take the input data and convert it to the same format as the training data. The feature selector performs the same task as in Step 1.

Step 4: Apply the BTDA-SVM learner to identify sentences that contain protein-protein interaction.

Step 5: Store extracted sentences to the database.

Combination of Active Learning with Semi-supervised Learning

One of the goals of this paper is to combine SSL and AL into a unified semi-supervised active learning technique for protein-

protein interaction extraction. We employ a proportion of unlabeled data in the learning tasks in order to resolve the problem of insufficient training data.

Our strategy of combining AL with SSL is inspired by the Tur et al.'s study [32]. We employ the break tie AL technique (BT-SVM) to train a classifier on both labeled and unlabeled data, and return to the user the most relevant results. Then, the learning system trains a classifier based on the Deterministic Annealing SSL technique (DA-SVM) on both the labeled and unlabeled data (S_t, S_k, and S_u), and results in the final model (Figure 2).

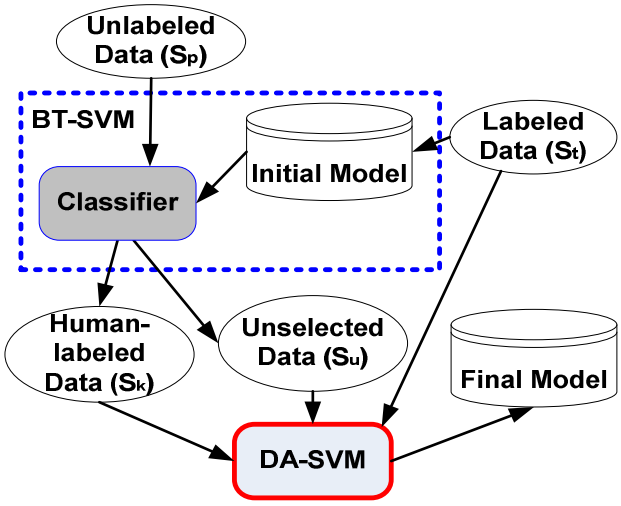


Figure 2: Combination of Active Learning with Semi-supervised Learning

BTDA-SVM is a combination of the active learning algorithm presented in Section 4 and the semi-supervised learning algorithm presented in Section 5. Instead of leaving out the instances classified with high confidence scores, this algorithm exploits them. Figure 3 explains the BTDA-SVM algorithm.

BTDA-SVM Algorithm

1. Given some amount of labeled training data S_t, and a larger amount of unlabeled data in the pool $S_p = \{s_1, ..., s_n\}$

2. $S_u = 0$ *where* S_u *is unselected training data*

3. Train a classifier using the current training data S_t

4. Classify an instance using the current training data S_p using the Active Learning classifier with the probability $P(s_i), i = 1, ..., n$

5. Manually label the set $S_k = \{s_i : P(s_i) < th\}$ where *th* is threshold

6. $S_t = S_t \cup S_k$

7. $S_u = S_u \cup (S_p \setminus S_k)$

8. Train a classifier using the augmented training data $S_t \cup S_u$

9. Get new S_p

Figure 3: BTDA-SVM Algorithm

4. ACTIVE LEARNING

Active learning, known as pool-based active learning, is an interactive learning technique designed to reduce the labor cost of labeling in which the learning algorithm can freely assign the unlabeled data instances to the training set. The basic idea is to select the most informative data instances for labeling by the users in the next learning round. In other words, the strategy of active learning is to select an optimal set of unlabeled data instances that minimizes the expected risk of the next round.

Breaking Tie (BT)

For a given instance, the regular SVMs results in distances among instances whose range is from 0 to 1. The value 0 means that the instance lies on the hyperplane and the value 1 indicates that the instance is a support vector.

To assign a probability value to a class the sigmoid function can be used with the assumption that a probability associated with a classifier indicates to which extent the classification result is trusted. In this case, Luo et al. [19] defines the parametric model in the following form:

$$P(y = 1 \mid f) = \frac{1}{1 + exp(Af + B)'} \quad (3)$$

where *A* and *B* are scalar values, which have to be estimated and *f* is the decision function of the SVMs. This parametric model is used for calculating the probabilities. To use this model, the SVM parameters (complexity parameter *C*, kernel parameter *k*) and the parameter *A* and *B* need to be calculated. Although cross validation can be used for this calculation, it is computationally expensive. An alternative is a pragmatic approximation method that all binary SVMs have the same *A* while eliminating *B* by assigning 0.5 to instances lying on the decision boundary and by trying to compute the SVM parameters and *A* simultaneously [19].

The decision function can be normalized by its margin to include the margin in the calculation of the probabilities.

$$P_{pq}(y = 1 \mid f) = \frac{1}{1 + exp(\frac{Af}{\| \omega \|})'} \quad (4)$$

where we currently look at class *p* and P_{pq} is the probability of class *p* versus class *q*. We assume that P_{pq}, q=*1*,2,... are independent. The final probability for class *p*:

$$P(p) = \prod_{q}^{q \neq p} P_{pq}(y = 1 \mid f) \quad (5)$$

It has been reported that the performance bases on this approximation is fast and accurate [19]. This probability model serves as basis for the Breaking Tie algorithm for semi-supervised learning.

5. SEMI-SUPERVISED SUPPORT VECTOR MACHINES

Support Vector Machines (SVMs) is a supervised machine learning approach designed for solving two-class pattern recognition problems. SVMs adopts maximum margin to find the decision surface that separates the positive and negative labeled training examples of a class [5].

Transductive Support Vector Machines (TSVMs) is an extended version of SVM that uses unlabeled data in addition to labeled data for train classifiers [16]. The goal of TSVMs is to determine which test data instances result in the maximum-margin hyperplane that separates the positive and negative examples for classifiers. Since every test instances need to be included in the SVM's objective function, finding the exact solution to the resulting optimization problem is intractable. To resolve this issue, Joachims [16] proposed an approximation algorithm. One issue of Joachims' approach, however, is that it requires the similar distribution of positive and negative instances between the test data and the training data. This requirement is difficult to meet particularly when the training data is small. The challenge is to find a decision surface that separates the positive and negative instances of the original labeled data and the unlabeled data to unlabeled data to be converted to labeled data with maximum margin. The unlabeled data sets apart the decision boundary from the dense regions, and the optimization problem is NP-hard [36]. Various approximation algorithms are found in [36].

The optimization problem held in TSVMs is a non-linear non-convex optimization [7]. Past several years, researchers have attempted to solve this critical problem. Chapell and Zien [9] proposed a smooth loss function, and a gradient descent method to find the decision boundary in a region of low density. Another technique is a branch-and-bound method [8] that searches for the optimal solution. But, it is applicable to a small number of examples due to involving the heavy computational cost. Despite the success of TSVM, the unlabeled data does not necessarily improve classification accuracy.

As an alternative to TSVMs, we explore an Deterministic Annealing approach to semi-supervised SVMs. The first approach was proposed by Luo and his colleagues [19] that formulated a probabilistic framework for image recognition. The Deterministic Annealing (DA) approach is the second proposed by Sindhwani et al. [28]. In the probabilistic framework, semi-supervised learning can be modeled as a missing data problem, which can be addressed by generative models such as mixture models. In the case of semi-supervised learning, probabilistic approaches provide us with various different ways to query unlabeled instances for labeling. A simple method is to train a model on the given labeled datasets and use this model on the unlabeled data. Each of these unlabeled instances is given probabilities that these instances belong to a given class. We can query the least certain instances or the most certain instances. The detailed description of the Deterministic Annealing semi-supervising learning is provided in the study of Luo and his colleagues [19].

Deterministic Annealing (DA)

Deterministic annealing (DA) is a special case of a homotopy method for combinatorial optimization problems [28]. We adopt the DA technique proposed by Sindhwani et al. [28] to extraction of protein-protein interaction. The detailed description of applying DA for SVMs is provided by Sindhwani et al. [28].

Suppose one is given a following non-convex optimization problem: $y^* = \arg\min_{y\in\{0,1\}^n} F(y)$

DA finds a local minimum of this in the following: First, DA treats the discrete variables as random binary variables over a space of probability distributions *P*. Second, to solve the optimization problem, DA finds a distribution $p \in P$ that minimizes the expected value of *F*. It makes the optimization problem to be continuous. For this reason, an additional convex term is added to the objective function which is the entropy *S* of the distribution denoted in Eq. 1.

$$p^* = \arg\min_{p\in P} E_p(F(y)) - T.S(p) \quad (1)$$

where the parameter *T* controls the trade-off between the expectation and the entropy (called the temperature of the problem) and $y \in \{0,1\}^n$ are the discrete variables for the objective function *F(y)*. For *T = 0* and *P* including all point-mass distributions over $\{0,1\}^n$, the global minimizer p^* in Eq. 1 will place all of its mass on the global minimizer of *F*. However, if $T \gg 0$, the entropy term in Eq. (1) dominates the objective function. With convexity, we can solve a sequence of problems for values of $T_0 > T_1 > \ldots > T_\infty = 0$ where each of them is initialized at the solution obtained by the previous one. This sequence of temperatures is called as the annealing schedule. When *T* is close to zero the influence of the entropy term becomes shrunken. Therefore, the distribution becomes more concentrated on the minimum of $E_p[F]$ which allows us to identify the discrete variables *y* by *p*. Note that there is no guarantee for global optimality because there is not always a path connecting the local minimizers for the chosen sequence of *T* to the global optimum of *F*.

Applying DA to SVMs

Given a binary classification problem, we consider a set of *L* training pairs $L = \{(x_1, y_1), \ldots, (x_L, y_L)\}, x \in \mathbb{R}^n, y \in \{1, -1\}$ and an unlabeled set of *U* test vectors $U = \{x_{L+1}, \ldots, x_{L+U}\}$. SVMs have a decision function $f_\theta(.)$ of the form $f_\theta(x) = w \cdot \Phi(x) + b,$ where $\theta = (w, b)$ are the parameters of the model, and $\Phi(\cdot)$ is the chosen feature map, often implemented implicitly using the kernel trick. Given a training set *L* and a test set *U*, for the TSVM optimization problem, find among the possible binary vectors $\{\gamma = (y_{L+1}, \ldots, y_{L+U})\}$ the one such that an SVM trained on

$L \cup (U \times \gamma)$ yields the largest margin. This combinatorial problem can be approximated as finding an SVM separating the training set under constraints which force the unlabeled examples to be as far as possible from the margin. This can be written as minimizing $\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{L}\xi_i + C^*\sum_{i=L+1}^{L+U}\xi_i$ subject to $y_i f_\theta(x_i) \geq 1-\xi_i, i = 1,...,L$ and $|f_\theta(x_i)| \geq 1-\xi_i, i = L+1,...,L+U$. This minimization problem is equivalent to minimizing

$$w^* = \min_{w,\{y_j\}_{j=1}^{u}} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{L} H_1(y_i f_\theta(x_i)) + C^*\sum_{i=L+1}^{L+U} H_1(|f_\theta(x_i)|) \quad (2)$$

where the function $H1(\cdot) = max(0, 1-\cdot)$ is the classical Hinge Loss function. In other words, TSVM seeks a hyperplane w and a labeling of the unlabeled examples, so that the SVM objective function is minimized. The discussion in Deterministic Annealing motivates a continuous objective function,

$$\tau_T(f, p) = E_p \tau(f, y') - TS(p) \quad (3)$$

that defined by taking the expectation of $\tau(f, y')$ (Eq. 1) with respect to a distribution p on y' and including entropy of p as a homotopy term.

For a fixed T, the solution to the optimization problem above is tracked as the temperature parameter T is lowered to 0. The DA algorithm returns the solution corresponding to the minimum value achieved when some stopping criterion is satisfied. The criterion used in the DA algorithm is the Pair-wise Mutual Information (PMI) between values of p in consecutive iterations. The parameter T is decreased in an outer loop until the total entropy falls below a threshold.

6. FEATURE SELECTION

Rich feature sets improve accuracy of the PPI extraction task [24]. The features used in Yang's paper include word features, keyword features, protein name distance features, and link path features, etc. In this paper, we explore various different features such as syntactic and lexical features as well as semantic features such as negated sentence features, interactor and its POS tag features into the feature sets. The total 9 features were selected for our semi-supervised learning technique (See Table 1).

Feature	Feature Value
Is negated sentence	True
No. of protein occurrences	3
Interactor name	response
Interactor POS	NN
Interactor position	88
No. of words in between proteins	24
No. of left words	-1
No. of right words	12
Link path status	Yes

Table 1: Features extracted from example sentence A.

Negation: We include whether a sentence is negated or not in the feature set. We use NegEx developed by Chapman and colleagues [6] for negation. NegEx is a regular expression-based approach that defines a fairly extensive list of negation phrases that appear before or after a finding of negation. NegEx treats a phrase as a negated one if a negation phrase appears within *n* words of a finding. The output of NegEx is the negation status assigned to each of the UMLS terms identified in the sentence: negated, possible or actual. NegEx uses the following regular expressions triggered by three types of negation phrases:

<pre-UMLS negation phrase> {0-5 tokens} <UMLS term> and <UMLS term> {0-5 tokens} <post-UMLS negation phrase>

There are three types of negation phrases in these expressions: 1) pre-UMLS, 2) post-UMLS and 3) pseudo negation phrases. Pre-UMLS phrases appear before the term they negate, while the post-UMLS phrases appear after the term they negate. Pseudo negation phrases are similar with negation phrases but are not reliable indicators of negation; they are used to limit the negation scope. All UMLS terms inside of the 0-5 tokens window are assigned the negation status depending on the nature of the negation phrase: negated or possible. The example of the negated sentence processed with NegEx is as follows:

[PREN].No[PREN] relevant changes in heart rate , body weight , and plasma levels of [NEGATED]renin[NEGATED] activity and aldosterone concentration were observed ➔ negated

Number of Proteins Named Entities (NE) occurrences: We extracted protein names from each sentence by using a Conditional Random Field (CRF)-based Named Entity Recognition (NER) technique.

To train the CRF NER, we used the training data provided for the BioCreative II Gene Mention Tagging task. The training data consist of 20,000 sentences. Approximately 44,500 GENE and ALTGENE annotations were converted to the MedTag database format [29]. Once we built the train model, we applied the CRF NER to extract proteins or genes from a sentence and counted the number of occurrences of genes in the sentence.

Interactor: Interactor is the term that shows the interaction among proteins in a sentence. The total of 220 interactor terms was identified. We applied a modified UEA stemmer to take care of term variations of interactor [11]. We did not apply an aggressive stemmer like Porter stemmer since we wanted to preserve the POS tag of the interactor.

Interactor POS: As for protein named entities, we applied the CRF-based POS tagging technique to tag tokenized words in a sentence. The CRF-based POS tagger was built on top of the MALLET package [20].

Interactor Position: We included the position of the interactor term in a sentence in the feature set.

Number of Words in between Proteins: We included the number of words in a left most Protein NE and a right most Protein NE in the feature set.

Number of Left Words: We included the number of words in the left side of the first appearance of a Protein NE in the feature set.

Number of Right Words: We included the number of words in the right side of the last appearance of Protein NE in the feature set.

Link Path Status: This feature set is obtained by Link Grammar that was introduced by Lafferty et al. [18]. Link Grammar is used to connect pairs of words in a sentence with various links. Each word is linked with connectors. A link consists of a left-pointing connector connected with a right-pointing connector of the same type on another word. A sentence is validated if all the words are connected. We assume that if a link path between two protein names exists, these two proteins have interaction relation. In our feature selection, if a Link path between two protein names exists, it is set to ''Yes", otherwise, ''No". The Link Grammar parser was used in several papers to extract protein-protein interaction [24, 34].

7. EXPERIMENTS

7.1 Data Sets

One of the issues in protein-protein interaction extraction is that different studies use different data sets and evaluation metrics. It makes it difficult to compare the results reported from the studies.

In this paper, we used three different datasets that have been widely used in protein-protein interaction tasks. These are 1) the AIMED corpus, 2) the BioCreAtIvE2 corpus that is provided as a resource by BioCreAtIvE II (Critical Assessment for Information Extraction in Biology) challenge evaluation, and 3) BioInfer corpus. Table 2 summarizes the characteristics of these three datasets.

Data Set	Total Sentences	Positive Sentences	Negative Sentences
AIMED	4026	951	3075
BioCreative2	4056	2202	1854
BioInfer	1100	573	527

Table 2: Data Sets Used for Experiments

AIMED: Bunescu et al. [4] manually developed the AIMED corpus3 for protein-protein interaction and protein name recognition. They tagged 199 Medline abstracts, obtained from the Database of Interacting Proteins (DIP) and known to contain protein interactions. This corpus is becoming a standard, as it has been used in the recent studies in several studies [4, 22, 33].

BioCreAtIvE2: is a corpus for protein-protein interactions, originated from the BioCreAtIvE task 1A data set for named entity recognition of gene/protein names. We randomly selected 1000 sentences from this set and added additional annotation for interactions between genes/proteins. 173 sentences contain at least one interaction, 589 sentences contain at least one gene/protein. There are 255 interactions, some of which include more than two partners (e.g., one partner occurs with full name and abbreviated) [17].

BioInfer: stands for Bio Information Extraction Resource. It was developed by Pyysalo et al. [25]. The corpus contains 1100 sentences from PubMed abstracts annotated for relationships, named entities, as well as syntactic dependencies.

Since previous studies that used these datasets performed 10-fold cross-validation, we also performed 10-fold cross-validation in these datasets and reported the average results over the runs.

For evaluation methodology, we use precision, recall, F-score, and AUC as our metrics to evaluate the performances of the methods.

7.2 Comparison Techniques

In this section, we briefly describe other techniques incorporated into semi-supervised SVMs and used to evaluate the performance of active semi-supervised learning models adopted in PPISpotter.

Baseline: Random Sampling (RS-SVM)

Random sampling of the unlabeled instances is a naïve approach to semi-supervised learning. We use this approach to compare with the other semi-supervised learning approaches as several studies used this approach to compare it with other semi-supervised learning approaches [13, 19].

Clustering (C-SVM)

One technique is a clustering algorithm applied for the unlabeled data. Fung and Mangasarian [19] used the k-median clustering and showed that the performance was competitive comparing to a supervised learning. The downside of a clustering approach is the correct number of the clusters needs to be pre-defined. We initially tried the two clustering techniques: K-means and Kernel K-means and found that there was only marginal difference in terms of performance. Therefore, we use K-means for the performance comparison.

Supervised SVMs (SVM)

The kernel we used as the baseline supervised SVM model is a linear kernel. One of the advantages of supervised SVMs with a linear kernel is that it can handle high dimensional data effectively. The reason is it compares the "active" features rather than the complete dimensions. This way, we can impose richer feature sets upon each training example to enhance system performance. The richer feature sets showed to be more effective than the simple feature sets [34]. Another advantage of linear kernel SVM is its low training and testing time costs. In addition, using linear kernel SVM only penalty parameter *C* needs to be adjusted in the algorithm, which is usually set as a constant in applications. In our experiments, the SVM-light[1] package was used. The penalty parameter *C* in setting the SVM is an important parameter since it controls the tradeoff between the training error and the margin. The SVM-light package does an excellent job on setting the default value for this parameter. In our experiments the parameter was left as default value since we observed that other manually determined values of this parameter in fact led to worse performance of supervised SVMs when compared with the default one.

8. RESULTS AND DISCUSSION

We evaluate and compare the performance of the active semi-supervised machine learning approach (BTDA-SVM) in several different ways. First, we compare it with three different techniques: random sampling, K-means clustering, and supervised

[1] http://svmlight.joachims.org/

SVMs. In addition, we test the performance of BTDA-SVM with supervised counterparts (SVMs) as well as an active learning technique (BT-SVM) for the task of protein-protein interaction extraction. Second, we exam whether the size of combined training datasets between unlabeled and labeled data have impact on the performance. As discussed in Section 3, we Break Tie and Deterministic Annealing, as a kernel function in BTDA-SVM.

Table 4 shows the results obtained with the AIMED data set. Our approach (BTDA-SVM) performs considerably better than other techniques in terms of precision, recall, and F-measure. BTDA-SVM's performance is superior to the regular SVMs approach by 34.79% in terms of precision. It is 25.55% better than the Random Sampling approach (RS-SVM) in terms of recall. In terms of F-measure, BTDA-SVM is 28.6% better than the regular SVMs. The Break Tie approach (BT-SVM) is the second best in terms of three measures.

We conducted individual t-tests essentially as specific comparisons. Our prediction that BTDA-SVM would be better than the other comparison techniques (BT-SVM, SVM, RS-SVM, and C-SVM) was confirmed $t(11)=3.6966E-11$, $p<0.05$ (one-tailed) at n-1 degrees of freedom (12 runs) while comparing with C-SVM which performed best over the other two comparison techniques. Similarly, the t-test confirmed that the performance difference of BT-SVM is statistically significant from C-SVM $t(11)=0.0169$, $p<0.05$ (one-tailed).

In Table 3, we also show the results obtained previously in the literature by using the same data set. Yakushiji et al. [33] used an HPSG parser to produce predicate argument structures. They utilized these structures to automatically construct protein interaction extraction rules. Mitsumori et al. [22] used SVMs with the unparsed text around the protein names as features to extract protein interaction sentences.

Algorithms	Measures		
	Precision	Recall	F-score
SVM	55.15%	42.47%	48.14%
RS-SVM	56.98%	41.71%	48.92%
C-SVM	64.53%	40.42%	50.67%
BT-SVM	65.23%	42.51%	53.64%
BTDA-SVM	74.34%	50.75%	61.91%
(Yakushiji et al., 2005)	33.70%	33.10%	33.40%
(Mitsumori et al., 2006)	54.20%	42.60%	47.70%

Table 3: Experimental Results – AIMED Data Set

Semi-supervised approaches are usually claimed to be more effective when there is less labeled data than unlabeled data, which is usually the case in real applications. To see the effect of semi-supervised approaches we perform experiments by varying the amount of labeled training sentences in the range [10, 3000]. For each labeled training set size, sentences are selected randomly among all the sentences, and the remaining sentences are used as the unlabeled test set. The results that we report are the averages over 10 such random runs for each labeled training set size. We report the results for the algorithms when edit distance based similarity is used, as it mostly performs better than cosine similarity.

Figure 5 shows the performance differences of five SVM-based learning techniques as the size of training data increases. BTDA-SVM performs considerably better than their supervised counterpart SVM, RS-SVM, C-SVM when we have small number of labeled training data. It is interesting to note that, although SVM is one of the best performing algorithms with more training data, it is the worst performing algorithm with small amount of labeled training sentences. Its performance starts to increase when number of training data is larger than 200. Eventually, its performance gets close to that of the other algorithms. Harmonic function is the best performing algorithm when we have less than 200 labeled training data.

BTDA-SVM consistently outperforms other techniques in this experiment. We observed that most of the techniques made significant improvement when the training data reaches 200 training instances. Compared to other techniques, BTDA-SVM did not make a radical change to the size of training data.

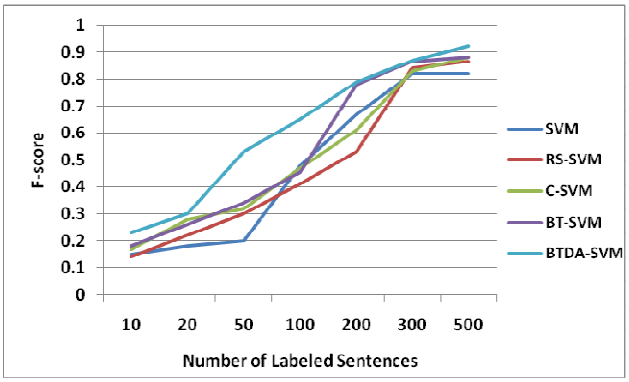


Figure 5: The F-score on the AIMED dataset with varying sizes of training data

Table 3 shows the experimental results with the BioCreative2 PPI data set.

The performance with BTDA-SVM is always better than other techniques by three measures. BTDA-SVM outperforms the regular SVMs (SVM) by 22.34%, 86.13%, and 48.89% respectively in terms of precision, recall, and F-measure. The second best performance is achieved by BT-SVM in terms of three measures.

Algorithms	Measures		
	Precision	Recall	F-score
SVM	70.23%	51.21%	58.33%
RS-SVM	71.7%	56.54%	62.5%
C-SVM	78.23%	88.68%	83.65%

BT-SVM	81.75%	93.5%	85.96%
BTDA-SVM	85.92%	95.32%	86.85%
TSVM-edit [37]	85.62%	84.89%	85.22%

Table 3: Experimental Results – BioCreative2 PPI Data Set

In t-test, we predict that BTDA-SVM would be better than the other three comparison techniques (SVM, RS-SVM, and C-SVM), and the prediction was confirmed t(11)=0.0312, p<0.05 (one-tailed) at n-1 degrees of freedom (12 runs) while comparing with C-SVM. However, our prediction that BT-SVM would be better than C-SVM was not confirmed t(11)=0.092, p<0.05 (one-tailed).

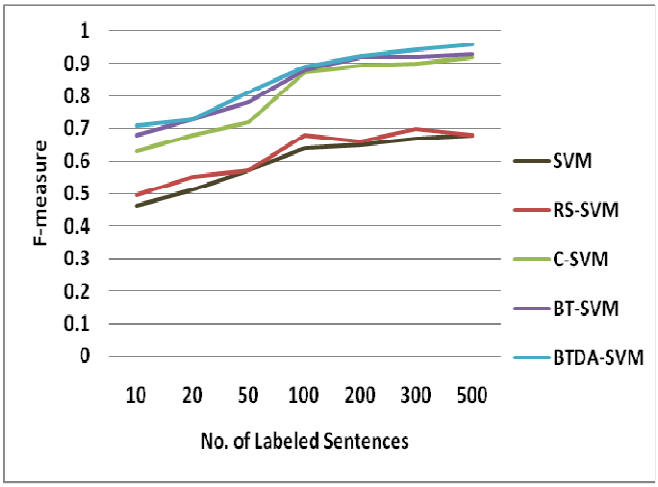


Figure 6: The F-score on the BioCreative II PPI dataset with varying sizes of training data

As shown in Figure 6, performance curves are different from ones with the AIMED data set. The performance of SVM and RS-SVM is consistently inferior to C-SVM, BT-SVM, and BTDA-SVM.

Although BTDA-SVM consistently outperforms other techniques in this experiment, it does not show statistical significance (In t-test, t(6)=0.2124, p<0.05 (one-tailed) at n-1 degrees of freedom). In addition, all techniques did not make a radical change to the size of training data.

We reported the performance of five comparison techniques with the BioInfer data set. Table 4 shows the experimental results in terms of precision, recall, and AUC.

Algorithms	Measures		
	Precision	Recall	AUC
SVM	65.89%	54.6%	0.843
RS-SVM	64.5%	55.2%	0.847
C-SVM	70.24%	60.2%	0.86
BT-SVM	79.29%	63.1%	0.918
BTDA-SVM	82.52%	65.2%	0.93
Graph Kernel [1]	47.7%	59.9%	0.849

Table 4: Comparison Results – BioInfer Data Set

BTDA-SVM's performance is the best over the other four techniques. It is better than the regular SVMs approach by 25.23%, 19.41%, and 10.32% in terms of precision, recall, and AUC respectively. With respect to AUC, the results of the t-test indicates that BTDA-SVM's performance is statistically significantly better than the other three comparison techniques (SVM, RS-SVM, and C-SVM), t(11)=8.3483E-6, p<0.05 (one-tailed) at n-1 degrees of freedom (12 runs) while comparing with C-SVM which performed best over the other two comparison techniques. In the same vein, our prediction that BT-SVM would be better than C-SVM was confirmed t(11)=0.00025, p<0.05 (one-tailed).

9. CONCLUSION

The goal of our study is two-fold: The first is to explore integrating an active learning technique with semi-supervised SVMs to improve the performance of classifiers. The second is to propose rich, comprehensive feature sets for the protein-protein interaction. To this end, we presented an active semi-supervised SVM-based PPI extraction system, PPISpotter, which encompasses the entire procedure of PPI extraction from the biomedical literature: protein name recognition, rich feature selection, and PPI extraction. In PPI extraction stage, besides several common features such as word features and keyword features, some new useful features including protein names distance feature, phrase negation, and link path feature were introduced for the supervised learning problem. We combined an active learning technique, Break Tie (BT-SVM) with the Deterministic Annealing-based semi-supervised learning technique (DA-SVM), which serves the core algorithm for the PPISpotter system (BTDA-SVM). This BTDA-SVM technique, compared with four different techniques including an active learning technique (BT-SVM), was tested on three widely used PPI corpora. The experimental results indicated that our technique, BTDA-SVM, achieves statistically significant improvement over the other three techniques in terms of precision, recall, F-measure, and AUC.

In future work, we plan to further explore the characteristics of active learning approaches to semi-supervised SVMs and refine our approach to achieve a better PPI extraction performance.

10. ACKNOWLEDGMENTS

Partial support for this research was provided by the National Science Foundation under grant DUE- 0937629 and by the New Jersey Institute of Technology. This research was also partially supported by the Brain Korea 21 Project in 2009 and the Korea Research Foundation Grant funded by the Korean Government (KRF-2009-0080667).

11. REFERENCES

[1] Airola, A., Pyysalo, S., Björne, J., Pahikkala, T., Ginter, F., Salakoski, T. A Graph Kernel for Protein-Protein Interaction Extraction. *Proceedings of BioNLP. Columbus, USA 2008*, 1-9.

[2] Bennett, K. P. and Demiriz, A. (1999) Semi-supervised support vector machines. In *Proceedings of the 1998 conference on Advances in neural information processing systems II*, pages 368-374, Cambridge, MA, USA,1999. MIT Press.

[3] Blaschke, C., Andrade, M.A., Ouzounis, C., and Valencia, A. (1999). Automatic Extraction of Biological Information from Scientific Text: Protein-Protein Interactions, *In Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, Heidelberg, Germany, 60-67.

[4] Bunescu, R., Ge, R., Kate, J. R., Marcotte, M. E., Mooney, R. J., Ramani, K. A., and Wong, W. Y. 2005. Comparative experiments on learning information extractors for proteins and their interactions. *Artificial Intelligence in Medicine*, 33(2):139–155, February.

[5] Burges, C. J. C. 1998. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(2):121–167.

[6] Chapman, W.W., Bridewell, W., Hanbury, P., Cooper, G.F., Buchanan, B.G., A Simple Algorithm for Identifying Negated Findings and Diseases in Discharge Summaries. *Journal of Biomedical Informatics*, 2001. 34: p. 301-310.

[7] Chapelle, O., Sindhwani, V., and Keerthi, S. S. Optimization techniques for semi-supervised support vector machines. Journal of Machine Learning Research, 9:203–233, 2008.

[8] Chapelle, O., Sindhwani, V., and Keerthi, S. S. Branch and bound for semi-supervised support vector machines. In B. Sch¨olkopf, J. Platt, and T. Hoffman, editors, Advances in Neural Information Processing Systems 19. MIT Press, Cambridge, MA, 2007.

[9] Chapelle, O. and Zien, A. Semi-supervised classification by low density separation. In Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics, pages 57–64, 2005.

[10] Cui, B., Lin, H., Yang, Z. Uncertainty sampling-based active learning for protein–protein interaction extraction from biomedical literature, Expert Systems with Applications 36 (2009) 10344–10350.

[11] Jenkins, Marie-Claire, Smith, Dan, Conservative stemming for search and indexing, 2005. fizz.cmp.uea.ac.uk/Research/stemmer/stemmer25feb.pdf

[12] Friedman, C., Kra, P., Yu, H.,Krauthammer, M., and Rzhetsky, A. (2001). GENIES: a natural-language processing system for the extraction of molecular pathways from journal articles. *Bioinformatics*, 17 Suppl 1, S74-82.

[13] Fung, F. and Mangasarian, O. Semi-supervised support vector machines for unlabeled data classification. *Optimization Methods and Software*,15: 29-44, 2001.

[14] Huang, M., Zhu, X., Hao, Y., Payan, D.G., Qu, K., Li, M. Discovering patterns to extract protein-protein interactions from full texts, Bioinformatics. 2004 Dec 12;20(18):3604-12. Epub 2004 Jul 29.

[15] Jenssen, T.K., Laegreid, A., Komorowski, J., and Hovig, E. (2001). A literature network of human genes for high-throughput analysis of gene expression. *Nature Genetics*, 28(1):21-8.

[16] Joachims, T. 1999. Transductive inference for text classification using support vector machines. In *Proceedings. of ICML-99*, pages 200–209, Bled, Slovenia, June.

[17] Kim, S., Shin, S.Y., Lee, I.H., Kim, S.J., Sriram, R., and Zhang, B.T. PIE: an online prediction system for protein-protein interactions from text. *Nucleic Acids Res.* 2008 Jul 1;36(Web Server issue):W411-5. Epub 2008 May 28.

[18] Lafferty, J., Sleator, D. and Temperley, D. 1992. Grammatical Trigrams: A Probabilistic Model of Link Grammar. In *Proceedings of the AAAI Conference on Probabilistic Approaches to Natural Language*, October, 1992.

[19] Luo, T., Kramer, K., Goldgof, D. B., Hall, L. O., Samson, S., Remsen, A., and Hopkins, T. Active learning to recognize multiple types of plankton. *Journal of Machine Learning Research*, 6, 589-613, 2005.

[20] McCallum, A. C. MALLET: A Machine Learning for Language Toolkit. http://mallet.cs.umass.edu. 2002.

[21] Miyao, Y., Sagae, K., Sætre, R., Matsuzaki, T., and Tsujii, J. (2009) Evaluating contributions of natural language parsers to protein–protein interaction extraction, *Bioinformatics*, 25: 3, 394-400.

[22] Mitsumori,T., Murata, M., Fukuda, Y., Doi, K., and Doi, H. 2006. Extracting protein-protein interaction information from biomedical text with svm. *IEICE Transactions on Information and Systems*, E89-D(8):2464–2466.

[23] Pustejovsky, J., Castano, J., Zhang, J., Kotecki, M., and Cochran, B. (2002). Robust relational parsing over biomedical literature: extracting inhibit relations. *Pacific Symposium on Biocomputing*, 362-73.

[24] Pyysalo, S., Ginter, F., Pahikkala, T., Koivula, J., Boberg, J., Jrvinen, J., and Salakoski, T. 2004. Analysis of Link Grammar on Biomedical Dependency Corpus Targeted at Protein-Protein Interactions. In *Proceedings of the International Workshop on Natural Language Processing in Biomedicine and its Applications.*

[25] Pyysalo S, Ginter F, Heimonen J, Björne J, Boberg J, Järvinen J, Salakoski T: BioInfer: A corpus for information extraction in the biomedical domain. BMC Bioinformatics 2007, 8:50.

[26] Ono, T., Hishigaki, H., Tanigami, A., and T. Takagi, T. 2001. Automated extraction of information on protein-protein interactions from the biological literature. *Bioinformatics*, 17(2):155–161.

[27] Schohn, G. and David Cohn, D. (2000) Less is more: Active learning with support vector machines. In *ICML '00: Proceedings of the Seventeenth International Conference on Machine Learning*, pages 839-846, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.

[28] Sindhwani, V., Keerthi, S.S. Chapelle, O. Deterministic Annealing for Semi-supervised Kernel Machines, International Conference on Machine Learning (ICML), 2006

[29] Smith, L. et al. 2008. Overview of BioCreative II gene mention recognition. *Genome Biology* 9(Suppl 2):S2.

[30] Song, M., Song, I.-Y., Hu, X., and Allen, R.B. (2005). Integrating Text Chunking with Mixture Hidden Markov Models for Effective Biomedical Information Extraction, *International Workshop on Bioinformatics Research and Applications*, Emory University, Atlanta, USA, May 22-25.

[31] Temkin, J. M. and Gilder, M. R. 2003. Extraction of protein interaction information from unstructured text using a context-free grammar. *Bioinformatics*, 19:2046–2053.

[32] Tur, G., Dilek Hakkani-Tur, D., Schapire, R. E. Combining active and semi-supervised learning for spoken language understanding, Speech Communication 45 (2005) 171–186.

[33] Yakushiji, A., Miyao, Y., Tateisi, Y., and Tsujii, J. 2005. Biomedical information extraction with predicate argument structure patterns. In *Proceedings of the Eleventh Annual Meeting of the Association for Natural Language Processing*, pages 93–96.

[34] Yang, Z., Lin, H., Li, Y. (2009) BioPPISVMExtractor: A protein–protein interaction extractor for biomedical literature using SVM and rich feature sets, *Journal of Biomedical Informatics*, doi:10.1016/j.jbi.2009.08.013

[35] Zhou, D., He, Y., and Kwoh, C. (2007), Semi-supervised learning of the hidden vector state model for extracting protein-protein interactions, Artificial Intelligence in Medicine, 41, 209-222.

[36] Zhu, X. 2005. Semi-supervised learning literature survey. *Technical Report 1530*, Computer Sciences, University of Wisconsin-Madison. http://www.cs.wisc.edu/~jerryzhu/pub/ssl survey.pdf.

[37] Erkan, G., Ozgur, A., and Radev., D. R. Semi-supervised classification for extracting protein interaction sentences using dependency parsing. In Proceedings of the Conference of Empirical Methods in Natural Language Processing (EMNLP '07), Prague, Czech Republic, June 28-30 2007.

Efficient Motif Finding Algorithms for Large-alphabet Inputs

Pavel P. Kuksa
Department of Computer Science
Rutgers University
Piscataway, NJ
pkuksa@cs.rutgers.edu

Vladimir Pavlovic
Department of Computer Science
Rutgers University
Piscataway, NJ
vladimir@cs.rutgers.edu

ABSTRACT

We consider the problem of identifying motifs, recurring or conserved patterns, in the biological sequence data sets. To solve this task, we present a new deterministic algorithm for finding patterns that are embedded as exact or inexact instances in all or most of the input strings. The proposed algorithm (1) improves search efficiency compared to existing algorithms, and (2) scales well with the size of alphabet. Our algorithm is orders of magnitude faster than existing deterministic algorithms for common pattern identification. We evaluate our algorithms on benchmark motif finding problems and real applications in biological sequence analysis and show that they maintain predictive performance with significant running time improvements.

Categories and Subject Descriptors

I.5.1 [**Pattern Recognition**]: Models - Statistical; I.5.2 [**Pattern Recognition**]: Design Methodology - Pattern Analysis; I.5.4 [**Pattern Recognition**]: Applications

General Terms

Algorithms, Design, Measurement, Performance, Experimentation

Keywords

sequence and structural motif finding, pattern discovery, exact algorithms for motif search, protein sequences, motif discovery

1. INTRODUCTION

Finding motifs or repeated patterns in data is of wide scientific interest [1, 2, 3, 4] with many applications in genomic and proteomic analysis. The motif search problem abstracts many important problems in analysis of sequence data, where motifs are, for instance, biologically important patterns. For example, elucidating motifs in DNA sequences is a critical first step in understanding biological processes as basic as the RNA transcription. There, the motifs can be used to identify promoters, the regions in DNA that facilitate the transcription. Finding motifs can be equally crucial for analyzing interactions between viruses and cells or identification of disease-linked patterns.

For the purpose of this study, motifs are (short) patterns that occur in an exact or *approximate* form in all or most of the strings in a data set. Consider a set of input strings $\mathcal{S}$ of size $N = |\mathcal{S}|$ constructed from an alphabet Σ. The solution for the (k, m, Σ, N)-motif finding problem (Figure 1) is the set $\mathcal{M}$ of k-mers (substrings of length k), $\mathcal{M} \subseteq \Sigma^k$, such that each motif $a \in \mathcal{M}, |a| = k$, is at Hamming distance at most m from all (or almost all) strings $s \in \mathcal{S}$.

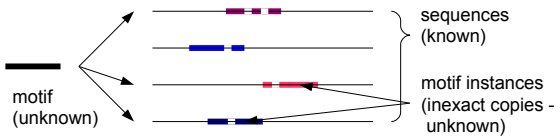


Figure 1: The motif search problem.

In this work, we focus on a deterministic, exhaustive approach to motif search. Exhaustive motif finding approaches are guaranteed to report all instances of motifs in a set of sequences, but are faced by the exponential complexity of such search. As a consequence, the problem quickly becomes intractable for even moderately long motifs and small alphabets. We present a new deterministic algorithm for finding common patterns with the search complexity that scales well with the size of the alphabet. Compared to existing algorithms in this class (e.g. [5, 6]) that have strong dependency on the alphabet size and work with small-alphabet input, our algorithms significantly improve search efficiency in the important case of large-alphabet inputs (e.g. protein alphabet, extended DNA alphabet, etc.) and inputs of large length. As we show in the experiments, using both synthetic and real data, our algorithms are orders-of-magnitude faster than existing state-of-the-art deterministic search algorithms, especially on large-alphabet inputs (e.g., protein sequences). This result extends applicability of the exact motif search algorithms to more complex problems requiring analysis of biological sequence data modeled as strings over large alphabets.

BIOKDD '10 Washington, DC, USA

2. RELATED WORK

The problem of motif discovery has been tackled extensively over the past two decades [7]. Within the class of exhaustive methods, a number of approaches have been proposed, including graph methods (WINNOWER) [2], explicit trie traversal (MITRA) [5], explicit mapping (Voting algorithms) [8], suffix trees [9, 6], sorting and enumeration [10], combinatorial pattern convolution [11], etc. Most of existing exhaustive algorithms use *explicit* exploration of the motif space and require time proportional to the size of the *neighborhood* of a k-mer, i.e. the number of k-mer sequences at Hamming distance of at most m from it. This size, $V(k,m) = \sum_{i=0}^{m} \binom{k}{i} (|\Sigma|-1)^i$, depends on the alphabet size, and can lead to high computational complexity and running times, as shown in Table 1.

Table 1: Exact algorithms for motif search

Algorithm	Time Complexity	Space Complexity
SPELLER [9]	$O(nN^2V(k,m))$	$O(nN^2/w)$
MITRA [5]	$O(knNV(k,m))$	$O(nNk)$
CENSUS [12]	$O(knNV(k,m))$	$O(nNk)$
Voting [8]	$O(nNV(k,m))$	$O(nV(m,k))$
RISOTTO [6]	$O(nN^2V(k,m))$	$O(nN^2)$
PMS [10]	$O(n^2NV(k,m))$	$O(n^2N)$

Explicit mapping (voting) algorithms proposed in [8] use an indicator array V of the maximum size $|\Sigma|^k$ to find motifs through *voting*. Each length-k substring observed in the input has at most one vote for each input sequence and gives this vote to all of its $V(k,m)$ neighbors. The substrings that occur in every input string will receive N votes and will be included in the output motif set $\mathcal{M}$. The algorithm takes $O(k^{m+1}|\Sigma|^m nN)$ time and requires at least $O(k^{m+1}|\Sigma|^m nN)$ space. The large space requirement of the algorithm restricts its usage to small values of k and m, as well as to small alphabet size $|\Sigma|$.

One of the most efficient exact algorithms for motif search, the mismatch tree (MITRA) algorithm [5], uses efficient trie traversal to find a set of motifs in the input strings. Under a trie-based computation framework [5, 13], the list of k-long contiguous substrings (k-mers) extracted from given strings is traversed in a *depth-first* search manner with branches corresponding to all possible symbol substitutions from alphabet Σ. Each leaf node at depth k corresponds to a particular k-mer feature (either exact or inexact instance of the observed exact string features) and will contain a list of matching features from each string. The leaf nodes corresponding to motifs will contain instances from all (or almost all) strings. The complexity of the trie-based traversal algorithm for motif finding is $O(k^{m+1}|\Sigma|^m nN)$. Note that the algorithm essentially explores the neighborhood of all $O(nN)$ k-mers in the input.

Another class of efficient algorithms is based on sorting and enumeration [10]. The PMSP algorithm enumerates all possible neighboring k-mers for the first string s_1 and outputs k-mers that occur in every string with Hamming distance at most m, similar to the Voting algorithm [8]. The PMSprune algorithm [10] employs a more efficient search strategy to traverse the candidate space and is an improvement, in the expected case, over the PMSP. We note that *explicit enumeration* is employed by all above-mentioned algorithms.

While the exact algorithms focus on retrieving all possible motif patterns, an important issue of estimating significance of the found motif patterns can be addressed with existing techniques as used in, for instance, non-exhaustive algorithms based on stochastic optimization (e.g., MEME [14]).

In contrast to existing exact exhaustive algorithms, we approach the problem of motif finding by performing an efficient search over patterns with wildcards. As a consequence, the proposed method's complexity becomes independent of the alphabet size.

3. COMBINATORIAL ALGORITHM FOR MOTIF SEARCH

In this section, we develop an efficient combinatorial algorithm for motif finding with the search complexity independent of the size of the alphabet $|\Sigma|$. The algorithm begins by finding a set of candidate motifs, followed by the construction of the intersections of those candidates' neighborhoods, the sequences that are at most m symbols apart from each candidate pair. In a crucial departure from other approaches, this set is efficiently represented using *stems*, or patterns with wildcards. The number of the stems does not depend on the alphabet size and is a function of the motif length (k), the number of mismatches (m) and the Hamming distance between k-mers. Patterns common to all (or almost all) input strings are then found by pruning the stems that do not satisfy the motif property (i.e., do not occur in all input strings).

The main idea of our approach is to construct a candidate set $\mathcal{C}$ which includes all motifs $\mathcal{M}$ plus some non-motifs, i.e. $\mathcal{M} \subseteq \mathcal{C}$, and then efficiently select true motifs from the candidate set. Given $\mathcal{C}$, the complexity of motif finding is then proportional to its size: the motifs can be extracted from $\mathcal{C}$ by checking each candidate against the motif property, a task we accomplish using $\binom{k}{m}$ rounds of counting sort in Algorithm 2. To generate $\mathcal{C}$, we collect the sets of stems which characterize the common neighbors of the pairs of k-mers (a,b) in the input. We call these sets the *stem sets*, $\mathcal{H}(a,b)$. Finding each $\mathcal{H}(a,b)$ is independent of the alphabet size and is accomplished in Algorithm 3. To further reduce the complexity, we construct the stem sets only for potential motif instances $\mathcal{I}$, those k-mers that are at Hamming distance of at most $2m$ from every input string. We find $\mathcal{I}$ using $\binom{k}{2m}$ rounds of counting sort (Algorithm 2). We outline our motif search algorithm below:

Algorithm 1 Motif search algorithm

1. Use multiple rounds of counting sort to iterate over input strings and construct a set of potential motif instances $\mathcal{I}$, k-mers that are at Hamming distance of at most $2m$ from each string (Algorithm 2).
2. Construct candidate set $\mathcal{C}$ by building stem sets $\mathcal{H}(a,b)$ for k-mer pairs in $\mathcal{I}$ (Algorithm 3)
3. Prune all stems from $\mathcal{C}$ that do not satisfy motif property using $\binom{k}{m}$ rounds of counting sort (Algorithm 2, Section 3.1.1)
4. Output remaining stems as motifs.

This algorithm uses as its main sub-algorithm (in step 2) a procedure that finds the intersection of k-mer neighborhoods for any pair of the k-mers a, b. This intersection finding algorithm is described in Section 3.2. We describe selection and pruning steps (steps 1 and 3) in Section 3.1.

The overall complexity of the algorithm is $O(\binom{k}{2m}nN + \binom{k}{m}HI^2)$, where H is the maximum size of $\mathcal{H}(a,b)$, and I is the size of $\mathcal{I}$, the number of k-mers used to construct the candidate set $\mathcal{C}$. The important fact that makes our algorithm efficient in practice is that typically $I \ll \min(nN, |\Sigma|^k)$ and $H \ll V(k,m)$, particularly for large alphabets. We demonstrate this in our experimental results and provide an expected-size analysis in Section 3.1.

3.1 Selection algorithm

A necessary condition for a group of k-mers to have a shared, common neighbor (motif) is that the Hamming distance between any pair of patterns cannot exceed $2m$. We will use this condition to select k-mers from input that are potential motif instances and place them in set $\mathcal{I}$. A particular k-mer a in the input is a potential motif instance if it is at the minimum Hamming distance at most $2m$ from each of the input strings. All other k-mers that violate the above condition cannot be instances of a motif and can be discarded. To select the valid k-mers, we use multiple rounds of count sort by removing iteratively $2m$ out of k positions and sorting the resulting set of $(k-2m)$-mers. A k-mer is deemed a potential motif instance if it matched at least one k-mer from each of the other strings in at least one of the sorting rounds. The purpose of sorting is to group same k-mers together. Using a simple linear scan over the sorted list of all input k-mers, we can find the set of potential motifs and construct $\mathcal{I}$. This algorithm is outlined in Algorithm 2. As we will see in the experiments (Section 4), the selection

Algorithm 2 Selection algorithm

Input: set of k-mers with associated sequence index, distance parameter d

Output: set of k-mers at distance d from each input string

1. Pick d positions and remove from the k-mers symbols at the corresponding positions to obtain a set of $(k-d)$-mers.
2. Use counting sort to order (lexicographically) the resulting set of $(k-d)$-mers.
3. Scan the sorted list to create the list of all sequences in which k-mers appear.
4. Output the k-mers that appear in every input string.

step significantly reduces the number of k-mer instances considered by the algorithm and improves search efficiency. The number of selected k-mers, i.e. the size of $\mathcal{I}$, is small, especially for large-alphabet inputs. This can be seen from the expected case analysis. For this purpose we assume that sequences are generated from a background process with few motifs implanted in the background-generated sequences. Assuming an iid background model with equiprobable symbols, the expected number of k-mers in the input of N strings of length n that match each of the N strings with up to $2m$ mismatches by chance is

$$E[\mathcal{I}_B] = |\Sigma|^k(1-(1-p_{k,2m})^n)^N =$$

$$= |\Sigma|^k\left(1-\left(1-\sum_{i=0}^{2m}\binom{k}{i}\left(\frac{1}{|\Sigma|}\right)^{k-i}\left(\frac{|\Sigma|-1}{|\Sigma|}\right)^{i}\right)^n\right)^N,$$

where $p_{k,2m}$ is the probability that two randomly selected k-mers are at distance of at most $2m$. For instance, for a set of $N = 20$ protein sequences (sampled from alphabet $|\Sigma| = 20$) of length $n = 600$ the expected number of potential motifs of length $k = 13, m = 4$ by chance is about 8, with $p_{13,8} = 2.9\ 10^{-4}$. Given t implanted motif instances, the average number of k-mers that will be selected from nN input samples, or the expected size of $\mathcal{I}$, is

$$E[\mathcal{I}] = t + nN(1-(1-p_{k,2m})^t) + E[\mathcal{I}_B].$$

Since t and p are typically small, for small pn, $E[\mathcal{I}] \ll nN$, the number of k-mers in the input. In the protein example above the expected size of $\mathcal{I}$ is about $1+3+8 = 12$ for $t = 1$, which is orders of magnitude smaller than $nN = 12000$, signifying the importance of creating $\mathcal{I}$ first. This is empirically demonstrated in Section 4.

3.1.1 Pruning using selection

The sorting approach of Algorithm 2 is also used to select patterns satisfying the motif property from the candidates $\mathcal{C}$ (Step 3 in main Algorithm 1). The pruning step is based on verifying the motif property (i.e. whether given patterns match all input sequences with up to m mismatches) and can be accomplished using $\binom{k}{m}$ rounds of counting sort.

3.2 Motif generation

In what follows, we describe an efficient algorithm that finds the set of *stems* that represent the set of k-mers shared by a pair of k-mers a and b. This process is used to create set $\mathcal{C}$ from potential instances $\mathcal{I}$, which is subsequently pruned to yield the true motif instances.

The number of k-mers in the common neighborhood of any two particular k-mers a and b assumes a fixed set values depending on the Hamming distance $d(a,b)$ between k-mers [15], for given values of $|\Sigma|$, k, and m. We want to represent the shared k-mers in this intersection using a set of *stems*, patterns with wildcards. However, the number of stems will not depend on the alphabet size $|\Sigma|$.

To find all stems shared by k-mers a and b, consider two sets of positions: *mismatch region* in which a and b disagree and *match region* in which a and b agree. We consider two cases depending on the number of mismatch positions (i.e. Hamming distance between a, b). In the first case, the distance $d(a,b)$ is at most m, the maximum number of mismatches allowed. In the second case, the distance $d(a,b)$ exceeds m. When $d(a,b) \leq m$, wildcard characters can appear both inside and outside of the mismatch region. When $d(a,b) > m$, wildcard characters can appear only inside the mismatch regions. Consider for example, the case of $d(a,b) = 0$ and $m = 1$. In this case, the set of stems is the set of patterns with 1 wildcard at each of the possible k positions (with the remaining positions as in a) plus one stem with 0 wildcards. When $m = 2$, and $d(a,b) = 1$, the set of stems will include patterns with 0 or 1 wildcard in $k-d$ positions and 0 or 1 wildcards in the remaining $d = 1$ positions. For example, for the pair (tgt, tgc) the corresponding patterns with wildcards are $tg?$, $t??$, $?g?$, $t?c$, and $?gc$, where ? denotes a wildcard.

We outline our algorithm for finding set of stems for the k-mer neighborhood intersection in Algorithm 3. The number of stems generated by the algorithm is

$$0 \leq d \leq m : \sum_{i=0}^{d}\sum_{j_1=0}^{d-i}\sum_{j_2=0}^{\min(m-d+i,m-i-j_1)}\binom{d}{i}\binom{d-i}{j_1}\binom{k-d}{j_2}$$

$$2m \geq d > m : \sum_{i=d-m}^{m} \sum_{j=0}^{m-i} \binom{d}{i}\binom{d-i}{j}.$$

The number of stems describing all the explicit k-mers shared between a, b does not depend on the alphabet size. The complexity of the stemming algorithm is proportional to the number of stems generated. The maximum number of stems H is $O(\sum_{i=0}^{2m}\binom{k}{i})$ for typical values of $m < k/2$. We use Algorithm 3 for every pair of k-mers in $\mathcal{I}$ (step 2) to construct $\mathcal{C}$ as outlined in the main algorithm.

Algorithm 3 Stem generation (independent of the alphabet size $|\Sigma|$)

Input: pair of k-mers a, b
Output: set of stems (patterns with wildcards) shared by a and b
  if if $d(a,b) \leq m$ **then**
    Set stem $= a$
    Set $i = 0 \ldots d$ positions in the mismatch region of the stem as in b
    Place $j_1 = 0 \ldots d - i$ wildcards inside the mismatch region
    Place $j_2 = 0 \ldots m - \max(d-i, j_1 + i)$ wildcards outside the mismatch region
  end if
  if $d(a,b) > m$ **then**
    Set stem $= a$
    Fix $i = d - m \ldots m$ positions in the mismatch region of the current stem as in b
    Place $j = 0 \ldots m - i$ wild-cards in the remaining $d - i$ positions in the mismatch region
  end if
  Output resulting stems (patterns with wildcards)

3.3 Algorithm analysis

The complexity of the selection step 1 for constructing $\mathcal{I}$ is $O(\binom{k}{2m})nN$ and does not depend on the alphabet size $|\Sigma|$. Steps 2 and and 3 have the complexity $O(\binom{k}{m}HI^2)$ and again do not depend on $|\Sigma|$. As a consequence, the three-step procedure gives us an efficient, alphabet-independent motif search algorithm that outputs all motifs embedded in the input $\mathcal{S}$. Our experiments will next demonstrate that this allows efficient exploitation of sparsity of typical solutions—we explore only a small portion of the motif space by focusing (using Algorithm 2) only on the support samples that are potential instances of the motifs. This results in significant reductions in running times, especially for large-alphabet inputs, i.e. the cases difficult for the current exact motif finding algorithms.

3.4 Extensions

Our proposed framework can be used to reduce search complexity for other exact search-based motif finding algorithms. Existing exhaustive algorithms typically (e.g. [5, 8, 10]) use the entire input (i.e. all the k-mers in the input) and find motif by essentially exploring neighborhoods of every k-mer in the input. Their search complexity can be improved by using a *reduced* set of k-mers instead of all input samples. This reduced set of k-mers can be obtained using our linear time selection algorithm (Algorithm 2, Section 3.1). Using reduced set of k-mers, the actual search complexity after the selection step becomes sublinear in the input size (since the number of selected k-mers $I = |\mathcal{I}|$ is much smaller than input length $O(nN)$). For instance, the search complexity of the trie-based algorithms (e.g., [5]) can be reduced to $O(\binom{k}{m}knN + IV(k,m))$ instead of $O(knNV(k,m))$, where $V(k,m)$ is $O(k^m|\Sigma|^m)$.

4. EXPERIMENTAL EVALUATION

We evaluate our algorithms on a synthetic benchmark motif finding task and real data. We first test our algorithms on the planted motif problem commonly used as a benchmark for evaluation of the motif finding algorithms [5, 10, 2]. We then illustrate our method on several DNA and protein sequence data sets.

4.1 Planted motif problem

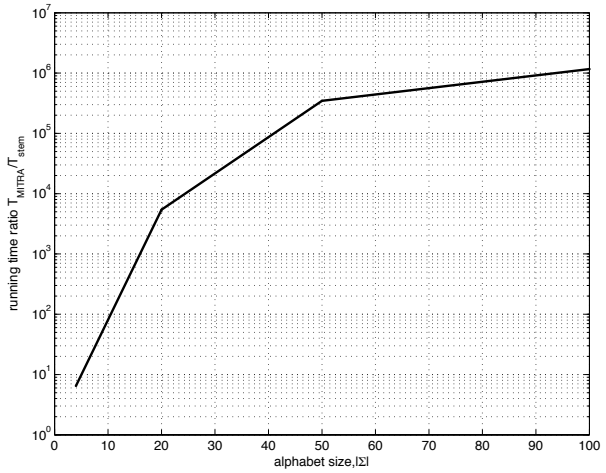
A planted motif problem [2] is the task of finding motifs and their instances in a set of sequences with variants of the consensus string (motif) implanted with up to m mismatches in every string. This task represents a well-defined subtle motif discovery problem. Instances of this problem with large number of mutations m are known to be challenging for most of the motif finding algorithms.

We follow the standard setting used in previous studies [2, 5, 10] and generate $N = 20$ random strings of length $n = 600$ using iid, uniformly distributed symbols from an alphabet of size $|\Sigma|$. We then embed a copy (with up to m substitutions at random positions) of a motif at a random location in every string. The task is then to identify motifs hidden in the input.

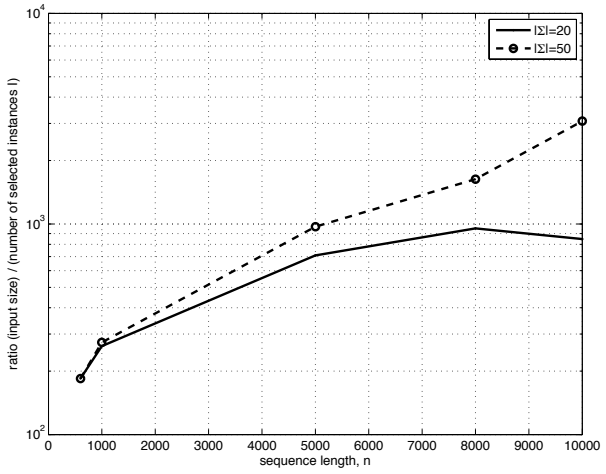
In Table 2, we compare running time of our algorithms with state-of-the-art motif finding algorithms on several challenging instances of the planted motif problem. We give running time comparison for large-alphabet ($|\Sigma| = 20-100$) instances in Table 3. As we can see from the results in Table 2 and Table 3, our algorithms show significant reduction in running times compared to state-of-the-art methods, especially for large-$|\Sigma|$ inputs (Table 3). For large alphabets and large k,m trie traversal takes substantial amount of time and results in these cases are not reported. We note that all of the compared algorithms (including the mismatch trie algorithm, MITRA) use the same setting and search for contiguous motifs only, while composite motifs such as dyads could be also recovered as, for instance, in the extension of MITRA algorithm [5]. In Figure 2(a), we show the running time ratio (logarithmic scale) between the mismatch trie traversal (MITRA) algorithm and our algorithm as a function of the alphabet set size. The running time is measured on (13,4) instances of the planted motif problem. For relatively small alphabet of size 20 our algorithm is about 10^4 times faster than the mismatch trie. The difference in running time increases with the size of the alphabet. Large alphabets can, for instance, arise when encoding the 3D protein structure or additional physical or chemical properties (cf. [16, 17]), a necessity in cases when sequences share little similarity at primary level.

Figure 2(b) shows efficiency of the selection (step 1 in the algorithm) as a ratio between the input size and the number of the selected samples (k-mers) $|\mathcal{I}|$. We observe that across different input sizes selection reduces the number of samples by a factor of about 10^3. The observed number of selected samples $I = |\mathcal{I}|$ agrees with the theoretical estimates (Section 3.1) (e.g., for $|\Sigma|$=50, n=5000, N=20, we expect about

52 k-mers to be selected, and the observed size of $\mathcal{I}$ is 103 k-mers). For small $np_{k,2m}$ the planted motif terms dominate the expected size of $\mathcal{I}$. For large $np_{k,2m}$ (large n and small $|\Sigma|$) the number of matches by chance increases and can even result in the decrease exhibited in the $|\Sigma| = 20$ case for $n > 8000$ when $E[\mathcal{I}_B]$ increases faster than nN.



(a)



(b)

Figure 2: (a) Running time ratio (T_{MITRA}/T_{stem}) as a function of the alphabet size (planted motif problem, $k = 13$, $m = 4$). (b) Ratio between input size (nN) and the number of selected samples ($I = |\mathcal{I}|$) as a function of the input length and alphabet size (planted motif problem, $k = 13$, $m = 4$). Note logarithmic scale.

4.2 Identifying TF binding sites

We use several data sets with experimentally confirmed TF binding sites: CRP, FNR, and LexA. The CRP data set contains 18 DNA sequences of length 105 with one or two CRP-binding sites [18, 2]. The FNR and LexA data sets are obtained from RegulonDB [19] database and contain 30 and 91 sequences known to have sites of length 14 and 20 bases. The task is to identify the sequence motif corresponding to the binding sites and the positions of sites within sequences.

For CRP, we use relatively long k-mers of length $k = 18$, with a large number of allowed mismatches ($m = 7$) from a given set of 18 DNA sequences ($|\Sigma| = 4$). For FNR and LexA data sets, we set motif length to $k = 14$ and $k = 16$ bases, with the maximum number of mismatches set to $m = 4$ and $m = 6$, respectively.

Figure 3(a) illustrates motifs found by the algorithm on the CRP data set. In the figure, colors indicate the importance of positions as measured by the number of hits between the found motif patterns and the sequences, with blue horizontal lines denoting true (confirmed) locations of the binding sites. The set of discovered locations agrees with the set of experimentally confirmed primary positions. The discovered motif patterns correspond to instances of the reference consensus motif `TGTGAnnnnnnTCACA` [20, 18]. Because of large k and m we observe running time improvements similar to the benchmark planted motif problems: our algorithm takes about 6 minutes, while the mismatch trie traversal requires about 12 times as long (4489 seconds). Allowing a large number of mismatches ($m = 7$) in this case is critical for the motif prediction performance, because fewer mismatches do not lead to successful identification of the binding sites.

For FNR and LexA motifs, our algorithm correctly finds consensus patterns `TTGATnnnnATCAA` and `CTGTnnnnnnnnnCAG`, in line with the validated transcription factor binding sites, with the performance coefficients [2] of 83.69 and 90.38.

4.3 Protein motif finding

We also apply our algorithm to finding subtle sequence motifs on several protein sequence datasets, a challenging task due to the increased alphabet size ($|\Sigma| = 20$) coupled with large k and m.

Lipocalin motifs. We first consider motifs in *lipocalins* which are topologically similar but have very diverse primary sequences. Using k-mer of length $k = 15$ with $m = 7$ mismatches, we identify motifs containing 15 residues with the instance majority `FD[IKLW]S[AKNR]FAGTWYE[ILMV]AK` (Figure 3(b)), which agrees with the known reference motif [21]. Our algorithm takes about 5 minutes to complete this task, while the mismatch trie algorithm takes more than a day. As in the case of the DNA, a large number of mismatches is critical for finding motifs, while smaller values of k, m do not result in motif identification.

Zinc metallopeptidase motif. In this experiment, 10 relatively long (average length is 800) human zinc metallopeptidase sequences used to test motif finding. Identification of subtle motifs in this case is made even more challenging by the length of the sequences. We use 11 residues long k-mer with $m = 5$ mismatches and find sequence motifs with the instance majority `VAAHELGHS[GL]G` in 9 out of 10 sequences that correspond to previously confirmed locations. We note the large number of mismatches ($m = 5$) was critical to motif identification.

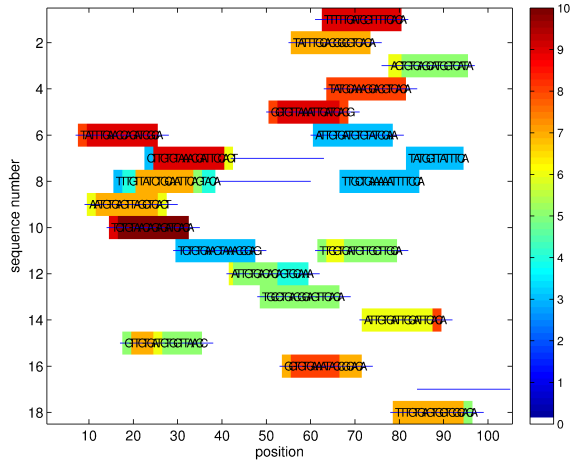
Super-secondary structure sequence motifs. We consider now two data sets of protein sequences with interesting 3D sandwich structure studied previously by biologists, for which existence of corresponding sequence motifs has been postulated [22]. Using Cadherin and Immunoglobin superfamilies as an example, our algorithm finds sequence patterns that correspond to the supersecondary structure (SSS) motifs [22, 23], i.e. arrangements of the secondary structure units (loops, strands). In particular, in Cadherin

Table 2: Running time comparison on the challenging instances of the planted motif problem (DNA, $|\Sigma| = 4$, $N = 20$ sequences of length $n = 600$). Problem instances are denoted by $(k, m, |\Sigma|)$, where k is the length of the motif implanted with m mismatches.

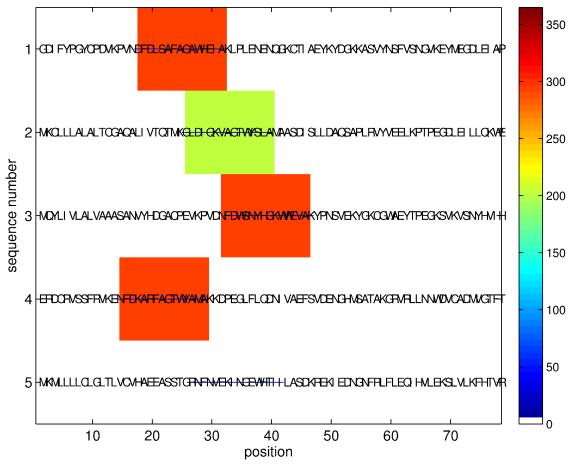
	Motif problem instances $(k, m, \|\Sigma\|)$					
Algorithm	(9,2,4)	(11,3,4)	(13,4,4)	(15,5,4)	(17,6,4)	(19,7,4)
Stemming	0.95	**8.8**	**31**	**187**	**1462**	**8397**
MITRA [5]	**0.89**	17.9	203	1835	4012	n/a
PMSPrune [10]	0.99	10.4	103	858	7743	81010
RISOTTO [6]	1.64	24.6	291	2974	29792	n/a

Table 3: Running time, in seconds, on large-$|\Sigma|$ inputs. (k, m) instances denote implanted motifs of length k with up to m substitutions.

$\|\Sigma\|$	(9,2)		(11,3)		(13,4)		(15,5)	
	MITRA	Stemming	MITRA	Stemming	MITRA	Stemming	MITRA	Stemming
20	8.39	**0.637**	1032.17	**1.07**	28905	**5.247**	n/a	**12.31**
50	89.82	**0.633**	12295.73	**0.963**	685015	**2.244**	n/a	**11.92**
100	265.94	**0.645**	n/a	**0.967**	>1 month	**2.227**	n/a	**11.86**



(a) CPR binding sites



(b) Lipocalin motifs

Figure 3: (a) Recognition of CRP binding sites ($k = 18, m = 7, |\Sigma| = 4$). (b) Lipocalin motifs ($k = 15, m = 7, |\Sigma| = 20$).

superfamily we find long motifs of length 20 (using $m = 4$ mismatches) corresponding to the secondary structure units *strand 1 - loop - strand 2* (`VIPPISCPENE[KR]GPFPKNLV`) and *strand 3 - loop - strand 4* (`YSITGQGAD[KNQT]PPVGVFII`) (3D SSS motif [23]). Our algorithm finds 36 potential motif instances (out of 330 samples) after the selection (step 1) and takes about 47 seconds (compared to about 600 seconds using the trie traversal). In Immunoglobin superfamily (C1 set domains), we find a sequence motif of length 19 `SSVTLGCLVKGYFPEPVTV` which corresponds to *strand 2-loop-strand 3* secondary structure units (2E SSS motif).

5. CONCLUSIONS

We presented a new deterministic and exhaustive algorithm for finding motifs, the common patterns in sequences. Our algorithm reduces computational complexity of the current motif finding algorithms and demonstrate strong running time improvements over existing exact algorithms, especially in large-alphabet sequences (e.g., proteins), as we showed on several motif discovery problems in both DNA and protein sequences. The proposed algorithms could be applied to other cases and challenging problems in sequence analysis and mining.

6. ACKNOWLEDGMENT

This research was partially supported by DIMACS, Center for Discrete Mathematics and Theoretical Computer Science, Rutgers University.

7. REFERENCES

[1] Eric P. Xing, Michael I. Jordan, Richard M. Karp, and Stuart Russell. A hierarchical Bayesian Markovian model for motifs in biopolymer sequences. In *In Proc. of Advances in Neural Information Processing Systems*, pages 200–3. MIT Press, 2003.

[2] Pavel A. Pevzner and Sing-Hoi Sze. Combinatorial approaches to finding subtle signals in dna sequences. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, pages 269–278. AAAI Press, 2000.

[3] Jean-Marc Fellous, Paul H. E. Tiesinga, Peter J. Thomas, and Terrence J. Sejnowski. Discovering Spike

Patterns in Neuronal Responses. *J. Neurosci.*, 24(12):2989–3001, 2004.

[4] Nebojsa Jojic, Vladimir Jojic, Brendan Frey, Christopher Meek, and David Heckerman. Using "epitomes" to model genetic diversity: Rational design of HIV vaccine cocktails. In Y. Weiss, B. Schölkopf, and J. Platt, editors, *Advances in Neural Information Processing Systems 18*, pages 587–594. MIT Press, Cambridge, MA, 2006.

[5] Eleazar Eskin and Pavel A. Pevzner. Finding composite regulatory patterns in DNA sequences. *Bioinformatics*, 18(suppl1):S354–363, 2002. http://www.ccls.columbia.edu/compbio/mitra/.

[6] Nadia Pisanti, Alexandra M. Carvalho, Laurent Marsan, and Marie-France Sagot. RISOTTO: Fast extraction of motifs with mismatches. In *LATIN*, pages 757–768, 2006.

[7] M Tompa, N Li, T Bailey, G Church, and B De Moor. Assessing computational tools for the discovery of transcription factor binding sites. *Nature Biotechnology*, Jan 2005.

[8] Francis Y. L. Chin and Henry C. M. Leung. Voting algorithms for discovering long motifs. In *APBC*, pages 261–271, 2005.

[9] Marie-France Sagot. Spelling approximate repeated or common motifs using a suffix tree. In *LATIN '98: Proceedings of the Third Latin American Symposium on Theoretical Informatics*, pages 374–390, London, UK, 1998. Springer-Verlag.

[10] Jaime Davila, Sudha Balla, and Sanguthevar Rajasekaran. Fast and practical algorithms for planted (l, d) motif search. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 4(4):544–552, 2007.

[11] I Rigoutsos and A Floratos. Combinatorial pattern discovery in biological sequences: The TEIRESIAS algorithm [published erratum appears in Bioinformatics 1998;14(2):229]. *Bioinformatics*, 14(1):55–67, 1998.

[12] Patricia A. Evans and Andrew D. Smith. Toward optimal motif enumeration. In *WADS*, pages 47–58, 2003.

[13] Christina Leslie and Rui Kuang. Fast string kernels using inexact matching for protein sequences. *J. Mach. Learn. Res.*, 5:1435–1455, 2004.

[14] Timothy L. Bailey and Charles Elkan. Unsupervised learning of multiple motifs in biopolymers using expectation maximization. *Mach. Learn.*, 21(1-2):51–80, 1995.

[15] Pavel Kuksa, Pai-Hsi Huang, and Vladimir Pavlovic. Scalable algorithms for string kernels with inexact matching. In *NIPS*, pages 881–888, 2008.

[16] S. Rackovsky. Sequence physical properties encode the global organization of protein structure space. *Proceedings of the National Academy of Sciences*, 106(34):14345–14348, 2009.

[17] Qi wen Dong, Xiao long Wang, and Lei Lin. Methods for optimizing the structure alphabet sequences of proteins. *Computers in Biology and Medicine*, 37(11):1610 – 1616, 2007.

[18] G D Stormo and G W Hartzell. Identifying protein-binding sites from unaligned DNA fragments. *Proceedings of the National Academy of Sciences of the United States of America*, 86(4):1183–1187, 1989.

[19] RegulonDB. `http://regulondb.ccg.unam.mx/`.

[20] CE Lawrence and AA Reilly. An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. *Proteins*, 7(1):41–51, 1990.

[21] CE Lawrence, SF Altschul, MS Boguski, JS Liu, AF Neuwald, and JC Wootton. Detecting subtle sequence signals: a Gibbs sampling strategy for multiple alignment. *Science*, 262(5131):208–214, 1993.

[22] A. E. Kister, A. S. Fokas, T. S. Papatheodorou, and I. M. Gelfand. Strict rules determine arrangements of strands in sandwich proteins. *Proceedings of the National Academy of Sciences of the United States of America*, 103(11):4107–4110, 2006.

[23] Super-Secondary Structure Database. `http://binfs.umdnj.edu/sssdb/`.

Planning combinatorial disulfide cross-links for protein fold determination

Fei Xiong
Dept. of Computer Science
Dartmouth College
6211 Sudikoff Laboratory
Hanover, NH 03755, USA
fei.xiong@dartmouth.edu

Alan M. Friedman
Dept. of Biological Sciences,
Markey Center for Structural
Biology, Purdue Cancer
Center, and Bindley
Bioscience Center
Purdue University
Lilly Hall
West Lafayette, IN 47907
afried@purdue.edu

Chris Bailey-Kellogg*
Dept. of Computer Science
Dartmouth College
6211 Sudikoff Laboratory
Hanover, NH 03755, USA
cbk@cs.dartmouth.edu

ABSTRACT

This paper presents an integrated computational-experimental method to determine the fold of a target protein by probing it with a set of planned disulfide cross-links. We start with predicted structural models obtained by standard fold recognition techniques. In a first stage, we characterize the fold-level differences between the models in terms of topological (contact) patterns of secondary structure elements (SSEs), and select a small set of SSE pairs that differentiate the folds. In a second stage, we determine a set of residue-level cross-links to probe the selected SSE pairs. Each stage employs an information-theoretic planning algorithm to maximize information gain while minimizing experimental complexity, along with a Bayes error plan assessment framework to characterize the probability of making a correct decision once data for the plan are collected. By focusing on overall topological differences and planning cross-linking experiments to probe them, our *fold determination* approach is robust to noise and uncertainty in the models (e.g., threading misalignment) and in the actual structure (e.g., flexibility). We demonstrate the effectiveness of our approach in case studies for a number of CASP targets, showing that the optimized plans have low risk of error while testing only a small portion of the quadratic number of possible cross-link candidates. Simulation studies with these plans further show that they do a very good job of selecting the correct model, according to cross-links simulated from the actual crystal structures. Fold determination can overcome scoring limitations in purely computational fold recognition methods, while requiring less experimental effort than traditional protein structure determination approaches.

*Corresponding author.

BioKDD '10 July 25, 2010, Washington DC, USA

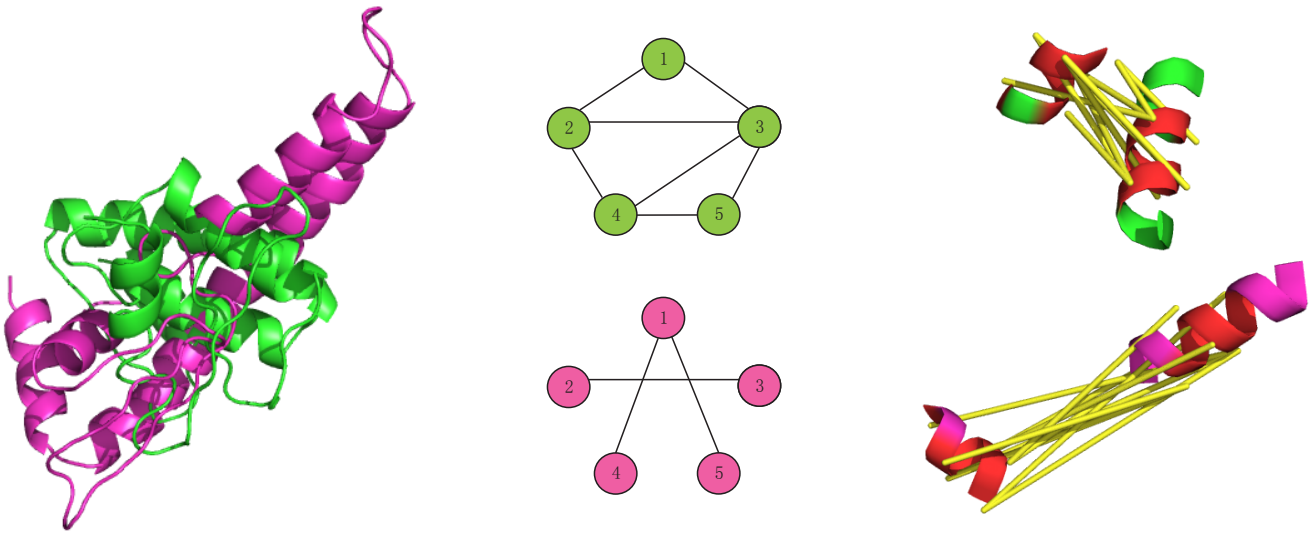
Keywords

Protein structure, structural genomics, disulfide cross-linking, Bayes error, experiment planning algorithm, feature subset selection

1. INTRODUCTION

Despite significant efforts in structural genomics, the vast majority (> 90% [4]) of available protein sequences do not have experimentally determined three-dimensional structures, due to experimental expense and limitations (e.g., lack of crystallizability). At the same time, since structure is more conserved than sequence, there may be only a small number (a thousand or two [16, 5]) of distinct natural "folds" (overall structural organizations), and many of them can already be found in the protein databank (PDB). Fold recognition techniques [20, 4] take advantage of this, and have become increasingly effective at identifying the fold of a given target sequence. However, the series of Critical Assessment of Structure Prediction (CASP) [15] contests demonstrates that, in the absence of sufficient sequence identity, it remains difficult for fold recognition methods to always select the correct model. While a native-like model is often among a pool of highly ranked models, it is not necessarily the highest-ranked one, and the model rankings depend sensitively on the scoring function used [24, 15]. Fig. 1(a) illustrates two possible alternative models for one target from a recent CASP competition.

Seeking to close the gap between computational structure prediction and experimental structural determination, we [22, 21] and others [6, 23, 3] have developed methods (which we call *structure elucidation*) to select structural models based on relatively rapid biochemical/biophysical experiments. One type of experiment particularly suitable for this purpose is *cross-linking*, which essentially provides distance restraints between specific pairs of residues, based on the formation (or not) of chemical cross-links. While residue-specific (e.g., lysine-specific) cross-linking has been effectively used for this task [8, 23, 12], we previously showed that planned *disulfide* cross-linking has a number of advantages, in terms of the ease and reliability of experiment and the quality of the resulting information content [22]. In disulfide cross-linking (or "trapping") [2, 9, 13], a pair of cysteine sub-



(a) (b) (c)

Figure 1: Protein fold determination by disulfide cross-linking. The example shows two models, but the method readily handles tens or even hundreds of models. (a) Two models, TS125_3 (green) and TS194_2 (magenta), for CASP target T0351, are of reasonable quality but have rather different topologies. (b) The three-dimensional structures are compiled into graphs on the secondary structure elements (SSEs), representing the topology in terms of contacting SSE pairs. A topological fingerprint is selected based on differences in SSE contacts (e.g., 1-2, 2-4, 3-5, etc.) that together distinguish the models. (c) For each SSE pair in the topological fingerprint, a set of residue pairs is selected for disulfide cross-linking, in order to robustly determine whether or not the SSE pair is actually in contact. The figure shows the selected cross-links (yellow) to test for SSE pair (1, 2). Residues selected for cross-linking are colored red.

stitutions is made and the formation of a disulfide bond after oxidation is evaluated, e.g., by alteration in electrophoretic mobility [2, 13, 22]. An important point for our purposes here is that disulfide cross-links are *plannable*—we control exactly which pair of residues is probed in a particular experiment.

While earlier methods have focused on probing geometry and selecting a model, we target here a more defined characterization of protein structure, ascertaining the overall protein fold. We call this approach *fold determination*, named in contrast to purely computational *fold recognition* and our less defined structure elucidation approach. We first characterize the topological / fold-level differences in a set of models in terms of contact patterns of secondary structure elements (SSEs); see Fig. 1(b). The topological representation allows for a robust experimental characterization of the structure, less sensitive to noise and uncertainty in both the models (e.g., threading misalignment) and the actual structure (e.g., flexibility). As a representation with fewer degrees of freedom than the complete threading models, the topological representation also enables us to explicitly consider all possibilities and handle the case when none of the models is correct. Once we have identified a subset of SSE pairs that are most informative for fold determination, we plan disulfide cross-links to evaluate these SSE pairs; see Fig. 1(c). By specifically planning for each such SSE pair, we can account for the dependence among the cross-links and select a set that will be robust to, and even help characterize, model misalignment and protein flexibility.

The method presented here strikes a balance between very limited cross-linking (e.g., six disulfide pairs in our earlier work [22]) and testing all residue pairs. We assume that robotic genetic manipulation methods (e.g., based on SPLISO [18] and RoboMix [1]) can construct a combinatorial set of dicysteine mutants, but that we still should test a much smaller set than all residue pairs. (Our plans require tens to around a hundred cross-links, depending on error requirements.) Thus we must optimize a plan so as to maximize information gain while minimizing experimental complexity. This is analogous to feature subset selection, where the goal is to choose a subset of features from a dataset such that the reduced set still keeps the most "distinguishing" characteristics of the original [14, 7]. At the topological level (Fig. 1(b)) the features are SSE pairs, and the objective is to select those that will correctly classify the real structure to a model. At the cross-link level (Fig. 1(c)) the features are potential disulfide pairs and the objective is to select those that will correctly classify contact/not for the SSE pair. For each level, we optimize a plan by employing an information-theoretic planning algorithm derived from the minimum redundancy maximum relevance approach [17]. We then evaluate a plan with a Bayes error framework that characterizes the probability of making a correct decision from the experimental data.

2. METHODS

We are given a set M of models. They may be redundant (i.e., some may have the same fold), and they may be incomplete (i.e., a representative of the correct fold may not be included). Our goal is to plan a set of disulfide cross-linking experiments (i.e., identify residue pairs to be individually tested) in order to select among them. As discussed in the

introduction, we do this in two stages (Fig. 1(b) and (c)), first selecting a "topological fingerprint" of SSE pairs to distinguish the folds, and then selecting cross-links to assess these SSE pairs.

2.1 Topological fingerprint selection

In order to compare SSE topologies, we need a common set of SSEs across the models. Since secondary structure prediction techniques are fairly stable [11, 10], it is generally the case that models have more-or-less the same set of SSEs, covering more-or-less the same residues ($> 50\%$ overlapping as observed in our test data). Our approach starts with a set S of SSEs that are common to at least a specified fraction (default 50%) of the given models. For example, both models in Fig. 1 have 5 α-helices, as do 63 other models for the same target. The later cross-link planning stage will account for the fact that the common SSEs may in fact extend over slightly different residues in the different models.

Given the SSE identities, we form for each model $m_i \in M$ an *SSE contact graph* $G_{SSE,i} = (S, C_i)$ in which the nodes S are the SSEs (common to the specified fraction of models, as described in the preceding paragraph) and the edges $C_i \subset S \times S$ are between contacting SSEs (specific to each model). We determine SSE contacts from residue contacts, deeming an SSE pair to be in contact if a sufficient set of residues are. Our current implementation requires at least 5 contacts (at < 9 Å C^β-C^β distance), and at least 20% of each SSE's residues to have a contact partner in the other SSE.

Our goal then is to find a minimum subset $F \subset S \times S$ of SSE pairs providing the maximum information content to differentiate the models. As discussed in the introduction, this is much like feature subset selection; in particular, the *max-dependency* feature selection problem seeks to find a set of features with the largest dependency (in term of mutual information) on the target class (here, the predicted structural model) [17]. While max-dependency leads to the minimum classification error, there is unfortunately a combinatorial explosion in the number of possible feature subsets that must be considered. To deal with the combinatorial explosion, we develop here an approach based on the minimum Redundancy Maximum Relevance (mRMR) method [17].

Probabilistic model.

First we develop a probabilistic model in order to evaluate the information content in a possible experiment plan. Let us treat each edge as being a binary random variable c representing whether or not the SSE pair is in contact, with $\Pr(c)$ the probability of being in contact ($c = 1$) or not ($c = 0$). We estimate $\Pr(c)$ by counting occurrence frequencies over the contact edge sets C_i for the models:

$$\Pr(c = x) = \frac{\sum_y q(c, x, y) \cdot |\{C_i : y = \mathbf{1}_{C_i}(c)\}|}{\sum_z \sum_y q(c, z, y) \cdot |\{C_i : y = \mathbf{1}_{C_i}(c)\}|}, \quad (1)$$

where the summed variables range over $\{0, 1\}$ and the indicator function $\mathbf{1}$ tests for membership of c in set C_i, and thus the set includes those SSE contact graphs for which the contact state of c agrees with y. To allow for noise, when evaluating $x = 1$ we include a contribution from $y = 0$ (false negative) along with that for $y = 1$ (true positive), and similarly when evaluating $x = 0$ we consider both $y = 1$ (false positive) and $y = 0$ (true negative). The q function weights the contributions for the agreeing and disagreeing case. We currently employ a uniform weighting independent of edge, since we observed in cross-link planning (below) that the expected error rate in evaluating any SSE contact was well below 10% when using a reasonable number of cross-links.

$$q(c, x, y) = \begin{cases} 0.9 & x = y; \\ 0.1 & x \neq y. \end{cases} \quad (2)$$

The approach readily extends to be less conservative and to allow different weights for different SSE pairs, e.g., according to cross-link planning (discussed in the next section).

We can likewise compute a joint probability $\Pr(c, c')$ from co-occurrence frequencies:

$$\Pr(c = x, c' = x') = \frac{\sum_{y,y'} q(c, x, y) \cdot q(c', x', y') \cdot |\{C_i : y = \mathbf{1}_{C_i}(c), y' = \mathbf{1}_{C_i}(c')\}|}{\sum_{z,z'} \sum_{y,y'} q(c, z, y) \cdot q(c', z', y') \cdot |\{C_i : y = \mathbf{1}_{C_i}(c), y' = \mathbf{1}_{C_i}(c')\}|} \quad (3)$$

where again the sums are over $\{0, 1\}$ and the indicator function is as described above.

Then we can evaluate the *relevance* of each SSE contact edge c in terms of its entropy $H(c)$; a high-entropy edge will help differentiate models while a low-entropy one won't. We can also evaluate the *redundancy* of a pair (c, c') of edges in terms of their mutual information $I(c, c')$; a high mutual-information pair contains redundant information.

$$H(c) = -\sum_x \Pr(c = x) \log \Pr(c = x) \quad (4)$$

$$I(c, c') = \sum_x \sum_{x'} \Pr(c = x, c' = x') \log \frac{\Pr(c = x, c' = x')}{\Pr(c = x) \Pr(c' = x')} \quad (5)$$

Experiment planning.

The mRMR approach seeks to minimize the total mutual information (redundancy) and maximize the total entropy (relevance). In this paper, we define the objective function as the difference of the two terms.

$$s(F) = \frac{1}{|F|} \sum_{c \in F} H(c) - \frac{1}{|F|^2} \sum_{c,c' \in F} I(c, c') \quad (6)$$

To optimize this objective function, we employ a first-order incremental search [17], which builds up a set F starting from the empty set and at each step adding to the current F the edge c_* that maximizes

$$c_* = \arg \max_{c \in (S \times S) \setminus F} \Big(H(c) - \frac{1}{|F|} \sum_{c' \in F} I(c, c') \Big) \quad (7)$$

The search algorithm stops when the score for c_* drops below a threshold (we use 0.01 for the results shown below).

The original mRMR formulation with first-order incremental search was proved to be equivalent to max-dependency (i.e., to provide the most information about the target classification) [17]. The proof carries over to our version upon substituting our formulations of redundancy and relevance (discrete, with choices of SSE pairs providing information about models) in place of the original ones (continuous,

with gene profiles representing different types of cancer or lymphoma). Essentially, it can be proved that the optimal max-dependency value is achieved when each feature variable is maximally dependent on the class of samples, while the pairwise dependency of the variables is minimized. Furthermore, this objective can be obtained by pursuing the mRMR criterion in the "first-order" incremental search (i.e., greedy) where one feature is selected at a time. Therefore we don't need to explicitly compute the complicated multivariate joint probability, but can instead compute just the pair-wise joint probabilities. We thus have an efficient algorithm for finding an optimal set of SSE pairs to differentiate models.

Data interpretation.

In the next section, we will describe the planning of disulfide cross-linking experiments to evaluate a given fingerprint. For now, let us assume that the form of experimental data X regarding a fingerprint F is a binary vector indicating for each edge whether or not the SSE pair was found to be in contact. Let us denote by $\mathcal{X} = \{0,1\}^{|F|}$ the set of possible binary vector values for X. Then the likelihood takes the joint probability over the edges, testing agreement between the observed contact state and that expected under the model:

$$\Pr(X \mid m) = \prod_{i=1}^{|F|} \Pr(F_i = X_i \mid m) \tag{8}$$

where we use the subscript to get the i^{th} element of the set. The naive conditional independence assumption here is reasonable, since the elements of F_i (SSE contact states) depend directly on the model, and are thus conditionally independent given the model. We then select the model with the highest likelihood. (If we have informative priors, evaluating model quality, we could instead select based on posterior probabilities.)

Plan evaluation.

In the experiment planning phase, we don't yet have the experimental data. However, we can evaluate the potential for making a wrong decision using a given plan by computing the *Bayes error*, ϵ. If we knew which model m were correct and which dataset X we would get, we could evaluate whether or not we would make the wrong decision, choosing a wrong model m' due to its having a higher likelihood for X than the correct model m. The Bayes error considers separately each case where one particular model is correct and one particular dataset results, and sums over all the possibilities. It weights each possibility by its probability—is the model likely to be correct, and if it is, are we likely to get that dataset. Thus:

$$\epsilon = \sum_{m \in M} \Pr(m) \cdot \sum_{X \in \mathcal{X}} \Pr(X \mid m) \cdot \mathbf{1}(\Pr(X \mid m) < \max_{m' \neq m} \Pr(X \mid m')) \tag{9}$$

where $\Pr(m)$ is the prior probability of a model, which we currently take as uniform, but could instead be based on fold recognition scores. Here and in the following formulas we use an indicator function $\mathbf{1}$ that gives 1 if the predicate is true and 0 if it is false. So we assume each different model is correct (at its prior probability), and assess whether or not it would be beaten for each different data set (at probability conditioned on the assumed correct model). This framework thereby gives a probabilistic evaluation of how likely it is that we will make an error, in place of the usual empirical cross-validation that is performed to assess a feature subset selected for classification.

In the case of fold determination, there may not be a single best model—a number of models may in fact have the same fold, and thus be equally consistent with the experimental data. Thus in the data interpretation phase we would not want to declare a single winner, but instead would return a set of the tied-for-optimal models. In the experiment planning phase, we develop a complementary metric to the Bayes error, which we call the *expected tie ratio*, τ:

$$\tau = \sum_{m \in M} \Pr(m) \cdot \sum_{X \in \mathcal{X}} \Pr(X \mid m) \cdot \frac{1}{|M|} \cdot \sum_{m' \neq m} \mathbf{1}(\Pr(X \mid m) = \Pr(X \mid m')) \tag{10}$$

The formula mirrors that for ϵ, but instead of counting the number of incorrect decisions, it counts the fraction of ties. Evaluating τ as we build up a topological fingerprint allows us to track the incremental power to differentiate folds, up to the point where we find that a set of models has the same fold and τ has flat-lined. The metric can readily be extended to account for sets of models whose likelihood is within some threshold of the best.

Finally, the topological fingerprint approach allows us to handle the "none-of-the-above" scenario, when we decide that no model is sufficiently good; i.e., the correct fold isn't represented by a predicted model. While in other contexts that would be done by comparing the likelihood to some threshold (is the selected model "good enough"?), here we can actually explicitly consider the chance of not considering the correct fold. Note that since a fingerprint typically has a small number of SSE pairs, we can enumerate the space $\mathcal{F} = \{0,1\}^{|F|}$ of its possible values (indicating whether or not each SSE pair in the fingerprint is in contact). Some of those values, $\mathcal{F}_M$, correspond to models in M, while the rest, $\mathcal{F} - \mathcal{F}_M$, are "uncovered". We want to decide if an uncovered fold $f' \in \mathcal{F} - \mathcal{F}_M$ is better than the fold f for the selected model. Moving from models to folds, we can evaluate $\Pr(X \mid f)$ by a formula like Eq. 8, simply testing whether each X_i has the value specified in f. Then we can decide that it is "none of the above" (models) if $\exists f' \in \mathcal{F} - \mathcal{F}_M$ such that $\Pr(X \mid f') \geq \max_{f \in \mathcal{F}_M} \Pr(X \mid f)$.

Moving from data interpretation to experiment planning, we can again evaluate a plan for the probability of deciding none of the above. If we think of Bayes error as the false positive rate, then we want something more like a false negative rate. We call this metric ν, the *expected none-of-the-above ratio*.

$$\nu = \sum_{f' \in \mathcal{F} \setminus \mathcal{F}_M} \Pr(f') \cdot \sum_{X \in \mathcal{X}} \frac{1}{2^{|F|}} \cdot \mathbf{1}(\Pr(X \mid f') > \max_{f \in \mathcal{F}_M} \Pr(X \mid f)) \tag{11}$$

Thus ν is the fraction of experimental datasets for which an uncovered fold will be better than the best covered fold. We

currently do not include a prior on X, in order to provide a direct assessment of how many experiments could lead to a none-of-the-above decision. However, we could obtain a weighted value by estimating $\Pr(X)$, e.g., from the priors on the individual SSE pairs (from Eq. 1). For the same reason, we treat $\Pr(f')$ as uniform over the uncovered folds f', rather than evaluating it by priors on SSE pairs.

Note that the formula does not include SSE pairs in $(S \times S) \setminus F$; i.e., pairs not in the fingerprint. This is as if they contribute equally to covered and uncovered folds, and thus do not affect the outcome. In the absence of other information or assumptions about the uncovered folds, this is a reasonable (and conservative) assumption, and yields an interpretable metric.

2.2 Cross-link selection

Once a topological fingerprint F has been identified, the next task is to optimize a disulfide cross-linking plan to experimentally evaluate the SSE pairs in the fingerprint. We separately plan for each SSE pair (their conditional independence was discussed in the previous section), optimizing a set of disulfide cross-link experiments (a single cross-link per experiment), such that, taken together, these cross-links will reveal whether or not the SSE pair is in contact. The overall plan is then the union of these SSE-pair plans. Thus we focus here on planning for a single SSE pair. We must account for noise and uncertainty in both the model and the actual protein, as well as for dependency among cross-links. This paper represents the first to address these issues.

Different models may place an SSE at somewhat different residues, so when planning cross-links to probe that SSE's contacts, it is advantageous to focus on residues common to many models (and thus able to provide information about cross-linkability in those models). We define for each SSE a set of common residues that may be used in a disulfide plan. Our current implementation includes all residues that appear in at least half of the models that have that SSE. In the following, let R denote the common residues for a target SSE pair.

For each model m_i we construct a *residue cross-link graph* $G_{xlink,i} = (R, D_i)$, in which the nodes are common residues R and there are edges $D_i \subset R \times R$ between possible disulfide pairs (specific to each model). We compute the *cross-linking distance* for a residue pair as the C^β–C^β distance, and take as edges those with distance at most 19 Å, based on an analysis of rates of disulfide formation [2, 22]. Our method could be generalized to include a more detailed geometric evaluation of the likelihood of cross-linking.

Probabilistic model.

We must define a probabilistic model in order to evaluate the information content provided by a set of cross-links. We treat possible cross-link (pair of residues) as a binary random variable indicating whether or not there is a cross-link. We start with the model of our earlier work, in which the prior probability of a cross-link wrt a model is 0.95 for distances $\leq$ 9Å, 0.5 for distances between 9 and 19 Å, and 0.05 for those $>$ 19 Å [22]. However, we also account for two important types of noise in this context: threading misalignment and structural flexibility (Fig. 2).

We place a distribution $\Pr(\delta)$ over possible offsets by which an SSE could be misaligned in a model. That is, residue number r in the model is really residue $r + \delta$ in the protein,

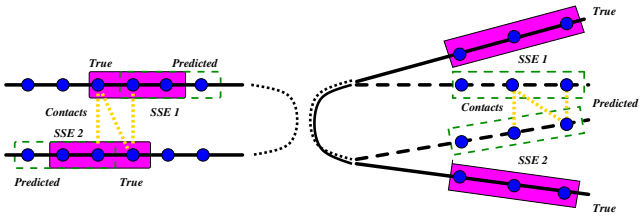


Figure 2: Noise factors in cross-link planning: misalignment (left) and flexibility (right). Blue dots represent residues and yellow lines their contacts. Regions in dashed lines are the modeled SSE and those in solid lines those measured by cross-linking experiments.

and thus a cross-link involving residue $r + \delta$ is really testing proximity to residue r. We use a distribution with 0.5 probability at 0 offset, decaying exponentially on both sides up to a maximum offset. Analysis of a model or the secondary structure prediction could provide a more problem-specific distribution. We currently consider each SSE separately; a future extension could model correlated misalignments resulting from threading.

We sample a set of alternative backbones for a model, and place a distribution $\Pr(b)$ over the identities of these alternatives. While there are many ways to sample alternative structures, we currently use Elastic Normal Modes (ENMs) as implemented by *elNémo* [19], sampling along the lowest non-trivial normal mode. We set $\Pr(b)$ according to the amplitude of the perturbation, using a Hookean potential function derived from ENMs. Future extensions could model different aspects of flexibility, such as local unfolding events during which a cross-link may be captured.

These two factors result in dependence among possible cross-links: if an SSE is misaligned or has moved relative to the original model, all its cross-links will be affected. However, the cross-links are conditionally independent given the particular value of misalignment or backbone choice. Thus we have for any two cross-links ℓ, ℓ':

$$\Pr(\ell, \ell') = \sum_m \Pr(m) \cdot \sum_\delta \Pr(\ell \mid m, \delta) \cdot \Pr(\ell' \mid m, \delta) \cdot \Pr(\delta) \tag{12}$$

and similarly for backbone flexibility. Furthermore, misalignment and flexibility are independent.

Experiment planning.

Our goal is to select a "good" set of residue pairs $L \subset R \times R$ to experimentally cross-link, in order to assess whether or not the SSE pair is in contact. This is another feature subset selection problem, and we again employ an mRMR-type incremental algorithm. Here a possible cross-link ℓ's relevance is evaluated in terms of the information it provides about whether or not the SSE pair is in contact: $I(\ell, c)$, where c is the binary random variable for contact of a target SSE pair. Redundancy is again evaluated in terms of mutual information. Thus the objective is:

$$s(L) = \frac{1}{|L|} \sum_{\ell \in L} I(\ell, c) - \frac{1}{|L|^2} \sum_{\ell, \ell' \in L} I(\ell, \ell') \tag{13}$$

and we incrementally select cross-links to maximize the difference in relevance regarding contact and average redun-

dancy with already-selected cross-links.

Data interpretation.

Once we have experimentally assessed cross-link formation for each selected residue pair, we can evaluate the probability of the SSE pair being in contact. Let Y be the set of cross-linking data, indicating for each residue pair in L whether or not a disulfide was detected. To decide whether or not c is in contact, we will compare $\Pr(Y \mid c = 1)$ and $\Pr(Y \mid c = 0)$, and take the one with higher likelihood. Intuitively, the more cross-links that are detected, the more confident we are that the SSE pair is in contact. Thus we currently employ a sigmoidal function to evaluate the likelihood:

$$\Pr(Y \mid c = x) = \frac{1}{1 + e^{(-1)^x \cdot (k - k_0)}} \ . \tag{14}$$

Here k is the number of detected cross-links in Y, and k_0 is the minimum number of positive cross-links for us to start believing c is in contact. For example, for $c = 1$, given a default number of 10 experiments, we set $k_0 = 3$ and the likelihoods of $c = 1$ for $k = 0$, 3, 6 are then approximately 0.05, 0.5, and 0.95, respectively. The metric could be extended to reward the broader distribution of cross-links throughout each SSE. However, in our current framework, we find that having a sufficient number of cross-links without regard to location tends to achieve that goal.

Plan evaluation.

Finally, in order to assess an experiment plan's robustness, we develop a Bayes error criterion to evaluate the probability of making a wrong decision regarding SSE contact.

$$\epsilon = \sum_{x \in \{0,1\}} \Pr(c = x) \cdot \sum_{Y \in \mathcal{Y}} \Pr(Y \mid c = x) \cdot \mathbf{1}(\Pr(Y \mid c = x) < \Pr(Y \mid c \neq x)) \tag{15}$$

As in the previous section, we sum over the possible outcomes (here, in contact or not) and the possible experimental results ($\mathcal{Y} = \{0,1\}^{|L|}$, all binary choices for cross-links in plan L), weighted by their probabilities, and see which yield the wrong decision. In the absence of an informative prior for c (and one that we want to use in interpreting the data), we simply use $\Pr(c = 1) = \Pr(c = 0) = 0.5$.

Note that, if desired, we could use the cross-linking Bayes error as a replacement for q (as $1 - \epsilon$) in evaluating $\Pr(c = x)$. These values could be precomputed for all candidate SSE pairs, or a fingerprint could be reevaluated and perhaps modified upon evaluating its possible cross-link plan.

3. RESULTS

We demonstrate the effectiveness of our approach with a representative set of 9 different CASP targets (Tab. 1), including proteins that are all-α, some that all-β, and some that are mixed α and β. For each target, a number of high-quality models have been produced by different groups; we evaluate those of common SSE content, as described in the methods. The models vary in similarity to the crystal structure (the PDB ID indicated), which is unknown at the time of modeling and furthermore not used for experiment planning, as well as to each other (the average root mean squared deviation in atomic coordinates, RMSD, between pairs of models is indicated). Our goal is to select for each target an experiment plan to robustly determine the model(s) of the same fold as the crystal structure.

CASP ID	PDB ID	2^o	AAs	Models	Av. RMSD
T0283_D1	2hh6	5α	97	162	17.26
T0289_D2	2gu2	5β	74	34	13.45
T0299_D1	2hiy	3α, 3β	91	30	15.23
T0304_D1	2h28	2α, 5β	101	26	15.76
T0306	2hd3	7β	95	45	14.22
T0312_D1	2h6l	2α, 5β	132	55	16.13
T0351	2hq7	5α	117	65	15.42
T0382_D1	2i9c	6α	119	196	12.79
T0383	2hnq	2α, 4β	127	59	11.61

Table 1: Test data sets (from CASP7)

Topological fingerprint selection.

Fig. 3 shows the trends of Bayes error (ϵ), expected tie ratio (τ), and expected none-of-the-above ratio (ν) as more SSE pairs are included in the topological fingerprint. It may seem counterintuitive that ϵ initially increases with the addition of SSE pairs. However, this is because we define the Bayes error of a tie as zero (Eq. 9), and separate out the tie ratio. With few SSE pairs in the fingerprint, τ is generally high—few decisions will be made, as many models look equally good, and the Bayes error is small. Then as SSE pairs are added, τ drops sharply—the fold is more specifically determined, decisions will be made, and the potential for error (as reflected in the Bayes error) increases. Once a sufficient number of SSE pairs has been selected, the specifically-determined fold is distinct, and the decisions are likely to be right, and ϵ will decrease. Thus it is both appropriate and helpful to consider ϵ and τ together, as they provide complementary information in the progress toward obtaining a unique and correct fold.

On the other hand, we observe that the ν value is usually 0 in the first few steps, because at that point there are not distinct folds separated, and it is easy for the SSE graphs from the predicted models to "cover" all the possible folds. ν becomes non-zero when there are uncovered folds. Its value first decreases because the number of covered folds and the number of uncovered folds are both increasing as more SSE pairs are included, and ν only gets contributions from an uncovered fold with *greater* (not equal) likelihood as the best covered fold. At some point the number of covered folds stops increasing (due to the limited set of predicted fold types), while the number of uncovered folds is still growing. Then the additional fold possibilities in the uncovered space result in a higher risk of "none-of-the-above", and thus the ν value starts increasing again. This trend is particularly obvious for targets T0289_D2 and T0304_D1; in fact, we return to T0304_D1 below as a real example of "none-of-the-above".

The fingerprint evaluation incorporates a parameter in the q function (Eq. 2), essentially indicating the confidence we expect to have in the experimental evaluation of an SSE pair. We performed a sensitivity analysis for three values of q, from 0.7 (fairly ambiguous) to 0.9 (fairly confident). Fig. 4 shows that for one target the trends are very similar for all three values; our algorithm is insensitive to the choice. Other targets display similar insensitivity (not shown).

Figure 3: Bayes error (ϵ), expected tie ratio (τ), and expected none-of-the-above ratio (ν) with addition of SSE pairs to fingerprints for targets. x-axis: SSE pairs. y-axis (left): τ, (%). y-axis (right): ϵ, ν.

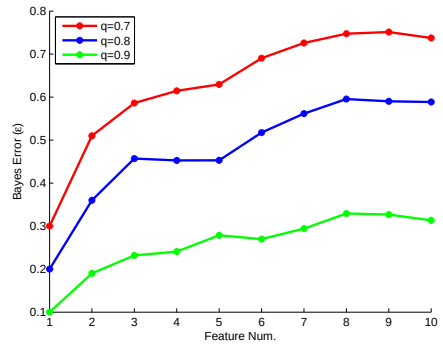
End-to-end simulation study.

Once we have selected a topological fingerprint, we next design a disulfide cross-linking plan to determine the contact state of the selected SSE pairs. To validate the overall process (fingerprint + disulfides), we perform a simulation study. Given a selected set of residue pairs for cross-linking, we use the crystal structure (PDB entry in Tab. 1) to determine whether or not they should form disulfides (C^{β}-C^{β} distance < 9 Å), and treat those evaluations as the data. We also use the set of all SSE pairs to directly compare the fold of each model with that of the crystal structure, and thereby label each model as being the "correct" fold or not depending on whether or not they have the same SSE contacts for the same SSE pairs. We then evaluate whether or not the simulated data for the selected cross-linking plans result in the same conclusions as the direct comparisons of folds.

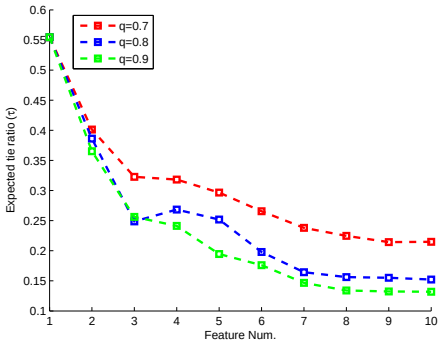
To compare the decision based on simulated cross-linking data with that based on fold analysis, we performed a Receiver Operator Characteristic (ROC) analysis. The area under the ROC curve (AUC) measures the probability that our experiment plan will rank a randomly chosen positive instance higher than a randomly chosen negative one. The larger the AUC, the better classification power our algorithm has to detect the right fold. Fig. 5 illustrates the simulation results on eight example protein targets (ROC analysis for T0304_D1 is not applicable and we will discuss it below). ROC curves are shown for different thresholds for the percentage r of residues that must be in contact to declare that the SSE pair is in contact in the structure or model. A high r value results in very few SSE pairs deemed to be in contact (we found that to happen with $r = 0.3$), while a low one yields some fairly weak contacts. As the figure shows, a moderate r value of around 0.2 generally results in quite good fold determination results.

Robustness.

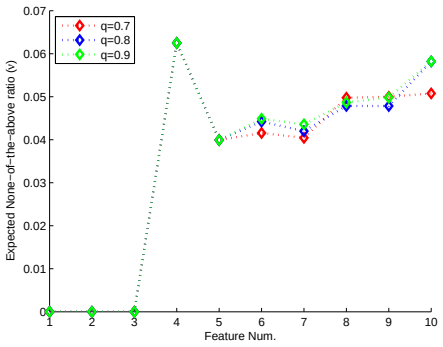
One of the merits of the fold determination approach is that it is robust to errors in models, and can even account for the case when none of the models is correct. The selected targets provide examples requiring such robustness; we summarize here just a couple. *Misalignment.* In Eq. 12



ϵ



τ



ν

Figure 4: Sensitivity analysis for three q function values (0.7, 0.8, and 0.9) for target T0383.

we account for being off by up to δ residues in the SSE locations. In the case of T0312_D1, there are 23 models of the correct fold, but with $\delta = 0$, only 7 of them agree with the crystal structure regarding all the cross-links in the experimental plan, while with $\delta = 1$ there are 14 that agree, and with $\delta = 2$ there are 16. The remaining unmatched models are looser in structure, and the match is sensitive to the threshold we use to measure SSE contacts. *None-of-the-above.* For target T0304_D1, none of the models has the same SSE contact graph as the crystal structure. The GDT [24] scores of predicted models are in the low 30s, which indicates relatively poor agreement with the crystal structures. As shown in Fig. 3, the ν value is relatively high, indicating a potential risk of missing the right fold. Indeed once we evaluate the models under the simulated data, we find that the likelihoods are low ($< 2 \times 10^{-3}$), compared to that (≈ 0.66) of the uncovered but correct fold, which is found by enumeration.

4. CONCLUSION

This paper presents a computational-experimental mechanism to rapidly determine the overall organization of secondary structure elements of a target protein by probing it with a planned set of disulfide cross-links. By casting the experiment planning process as two stages of feature selection—SSE pairs characterizing overall fold and residue pairs characterizing SSE pair contact states—we are able to develop efficient information-theoretic planning algorithms and rigorous Bayes error plan assessment frameworks. Focusing on fold-level analysis results in a novel approach to elucidating three-dimensional protein structure, robust to common forms of noise and uncertainty. At the same time, the approach remains experimentally viable by finding a greatly reduced set of residue pairs (tens to around a hundred, out of hundreds to thousands) that provide sufficient information to determine fold.

5. ACKNOWLEDGEMENTS

This work was inspired by conversations with and related work done by Michal Gajda and Janusz Bujnicki, International Institute of Molecular and Cell Biology, Poland. It was supported in part by US NSF grant CCF-0915388 to CBK.

6. REFERENCES

[1] L. V. Avramova, J. Desai, S. Weaver, A. M. Friedman, and C. Bailey-Kellogg. Robotic hierarchical mixing for the production of combinatorial libraries of proteins and small molecules. *J. Comb. Chem.*, 10:63–68, 2008.

[2] C. Careaga and J. Falke. Thermal motions of surface alpha-helices in the D-galactose chemosensory receptor. Detection by disulfide trapping. *J. Mol. Biol.*, 226:1219–1235, 1992.

[3] T. Chen, J. Jaffe, and G. Church. Algorithms for identifying protein cross-links via tandem mass spectrometry. *J. Comp. Biol.*, 8:571–583, 2001.

[4] A. Godzik. Fold recognition methods. *Methods Biochem. Anal.*, 44:525–546, 2003.

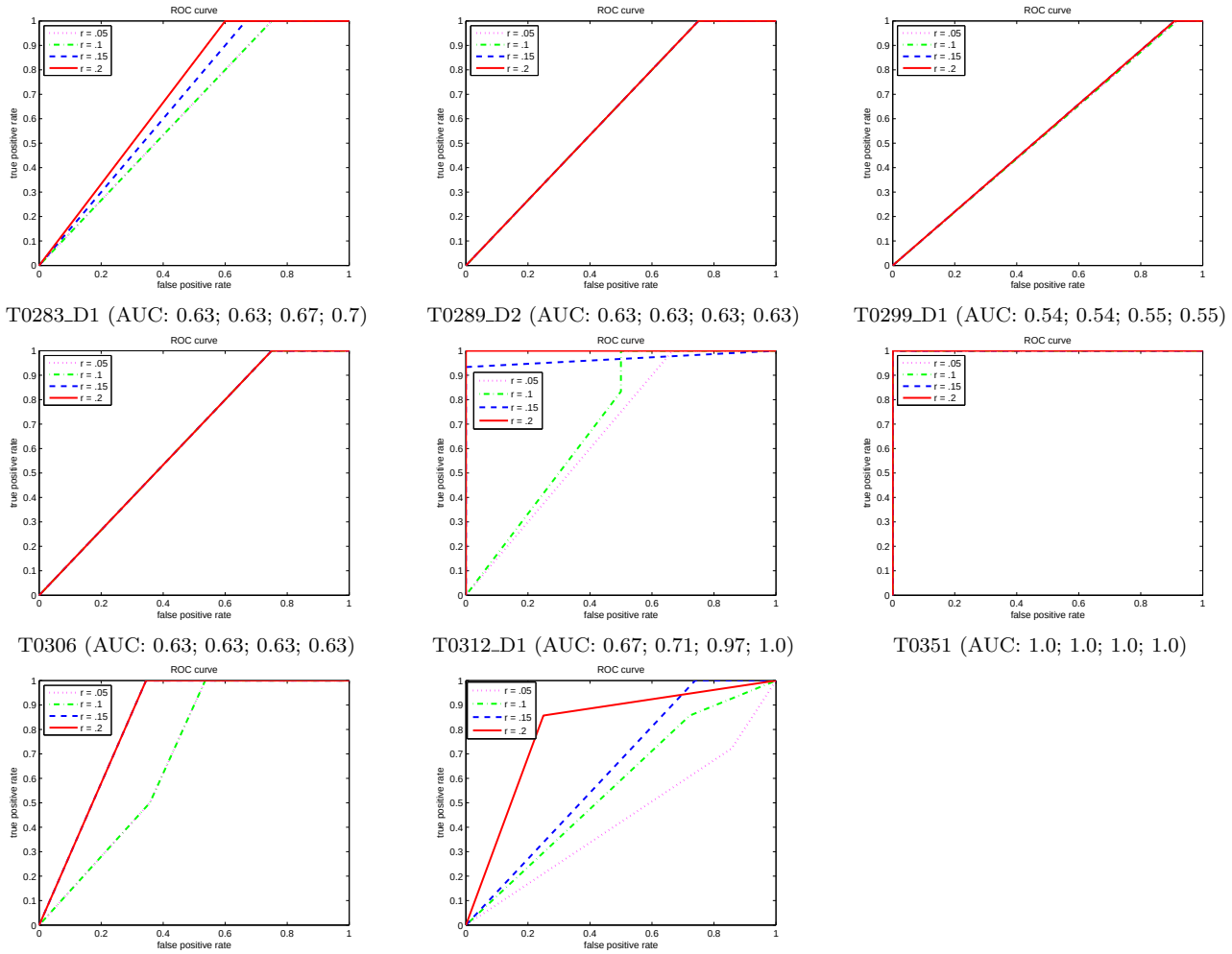
[5] S. Govindarajan, R. Recabarren, and R. A. Goldstein. Estimating the total number of protein folds. *Proteins*, 35:408–414, 1999.

[6] V. Grantcharova, D. Riddle, and D. Baker. Long-range order in the src SH3 folding transition state. *PNAS*, 97:7084–7089, 2000.

[7] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157–1182, 2003.

[8] M. Haniu, L. O. Narhi, T. Arakawa, S. Elliott, and M. F. Rohde. Recombinant human erythropoietin (rHuEPO): cross-linking with disuccinimidyl esters and identification of the interfacing domains in EPO. *Protein Sci.*, 9:1441–1451, 1993.

[9] R. Hughes, P. Rice, T. Steitz, and N. Grindley. Protein-protein interactions directing resolvase



T0283_D1 (AUC: 0.63; 0.63; 0.67; 0.7)

T0289_D2 (AUC: 0.63; 0.63; 0.63; 0.63)

T0299_D1 (AUC: 0.54; 0.54; 0.55; 0.55)

T0306 (AUC: 0.63; 0.63; 0.63; 0.63)

T0312_D1 (AUC: 0.67; 0.71; 0.97; 1.0)

T0351 (AUC: 1.0; 1.0; 1.0; 1.0)

T0382_D1 (AUC: 0.69; 0.69; 0.83; 0.83)

T0383 (AUC: 0.43; 0.57; 0.63; 0.80)

Figure 5: ROC curves for eight simulation studies, at different SSE contact fraction thresholds r. T0304_D1 doesn't have a predicted model that matches the crystal structure and thus is analyzed separately (see the discussion for *Robustness*). x-axis: False Positive Rate. y-axis: True Positive Rate. AUC: Area under the ROC curve, for r of 0.05, 0.1, 0.15, and 0.2 respectively.

site-specific recombination: A structure-function analysis. *EMBO J.*, 12:1447–1458, 1993.

[10] D. Jones. Protein secondary structure prediction based on position-specific scoring matrices. *J. Mol. Biol.*, 292(2):195–202, 1999.

[11] W. Kabsch and C. Sander. Dictionary of protein secondary structure: pattern recognition of hydrogen-bonded and geometrical features. *Biopolymers*, 22:2577–2637, 1983.

[12] G. H. Kruppa, J. Schoeniger, and M. M. Young. A top down approach to protein structural studies using chemical cross-linking and Fourier transform mass spectrometry. *Rapid Commun. Mass Spectrom.*, 17:155–62, 2003.

[13] I. Kwaw, J. Sun, and H. Kaback. Thiol cross-linking of cytoplasmic loops in lactose permease of *Escherichia coli*. *Biochemistry*, 39:3134–3140, 2000.

[14] H. Liu and H. Motoda. *Feature Selection for Knowledge Discovery and Data Mining*. Springer, 1998.

[15] J. Moult. A decade of CASP: progress, bottlenecks and prognosis in protein structure prediction. *Curr. Opin. Struct. Biol.*, 15:285–289, 2005.

[16] C. Orengo, A. Michie, S. Jones, D. Jones, M. Swindells, and J. Thornton. CATH—a hierarchic classification of protein domain structures. *Structure*, 5:1093–1108, 1997.

[17] H. Peng, F. Long, and C. Ding. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. Patt. Anal. Machine Intell.*, 27:1226–1238, 2005.

[18] L. Saftalov, P. A. Smith, A. M. Friedman, and C. Bailey-Kellogg. Site-directed combinatorial construction of chimaeric genes: General method for optimizing assembly of gene fragments. *Proteins*, 64:629–642, 2006.

[19] K. Suhre and Y. H. Sanejouand. ElNemo: a normal

mode web server for protein movement analysis and the generation of templates for molecular replacement. *Nucleic Acids Res.*, 32:610–614, 2004.

[20] J. Xu, M. Li, D. Kim, and Y. Xu. RAPTOR: Optimal protein threading by linear programming. *J. Bioinform. Comput. Biol.*, 1:95–117, 2003.

[21] X. Ye, A. M. Friedman, and C. Bailey-Kellogg. Optimizing Bayes error for protein structure model selection by stability mutagenesis. In *Proc. CSB*, pages 99–108, 2008.

[22] X. Ye, P. K. O'Neil, A. N. Foster, M. J. Gajda, J. Kosinski, M. A. Kurowski, J. M. Bujnicki, A. M. Friedman, and C. Bailey-Kellogg. Probabilistic cross-link analysis and experiment planning for high-throughput elucidation of protein structure. *Protein Sci.*, 13:3298–3313, 2004.

[23] M. M. Young, N. Tang, J. C. Hempel, C. M. Oshiro, E. W. Taylor, I. D. Kuntz, B. W. Gibson, and G. Dollinger. High throughput protein fold identification by using experimental constraints derived from intramolecular cross-links and mass spectrometry. *PNAS*, 97:5802–5806, 2000.

[24] A. Zemla, C. Venclovas, J. Moult, and K. Fidelis. Processing and analysis of CASP3 protein structure predictions. *Proteins*, Suppl. 3:22–29, 1999.

A new approach for detecting bivariate interactions in high dimensional data using quadratic discriminant analysis

Jorge M. Arevalillo
Department of Statistics and Operational Research. UNED
Paseo Senda del Rey 9
28040. Madrid, Spain
jmartin@ccia.uned.es

Hilario Navarro
Department of Statistics and Operational Research. UNED
Paseo Senda del Rey 9
28040. Madrid, Spain
hnavarro@ccia.uned.es

ABSTRACT

One of the drawbacks we face up when analyzing genomic and proteomic data is the degradation of the performance of the design classifier as the number of inputs increases. This phenomenon is usually known as the peaking phenomenon; it appears in small sample-high dimensional data ($n \ll p$) which are very common in the bioinformatics domain. Highly predictive bivariate interactions whose marginals have a weak discriminatory power are also affected by this phenomenon; so they are usually considered as noisy inputs and the bivariate pattern gets lost. This paper studies the peaking phenomenon for a benchmark of classification rules in regards to this type of weak marginal / strong bivarite interactions. We conclude that quadratic discriminant analysis (QDA) is comparable or outperforms the remaining design classifiers when computational cost and predictive accuracy criteria are considered. The paper proposes an exhaustive search strategy that divides the input space in a blockwise manner and explores it by fitting QDA classifiers; the size of the blocks is determined by the resistance of QDA to peaking. The search leads to a few small chunks of features with a high predictive accuracy; these are likely to contain the hidden interaction patterns we are looking for; now a closer look at this smaller subset of inputs by means of an exhaustive search will lead to their detection. The efficiency of the algorithm will be studied in synthetic scenarios. It will also be applied to a microarray data experiment in order to illustrate its usefulness in a real case.

General Terms

Algorithms, Experimentation

Keywords

Bivariate interaction, quadratic discriminant analysis, high dimensional data

1. INTRODUCTION

The development of high-throughput technologies, such as gene or protein microarrays, has provided the scenario of the state of cells by monitoring the expression levels of hundreds or thousands of biological inputs (p) for a few number (n) of experimental units measured under different clinical conditions. A challenging problem within this domain is the identification of inputs or interactions of them highly correlated to the outcome. The low sample-high dimensional ($n \ll p$) structure of the data we handle makes the challenge a difficult task, specifically when we are concerned with the detection of bivariate biomarker interactions. Some papers that tackle this problem by using scores of pairwise feature association are [19, 9], which introduced the TSP score, and [7] that explores the data for the search of gap/substitution and on/off association patterns with scores based on the changes of intra-class correlation. These approaches assume a specified shape for the interaction.

In this paper we address the problem by evaluating the performance of a classification rule trained on the data at hand; hopefully, this will enrich the typology of interactions that might appear. One of the main drawbacks for facing up this problem is the well known peaking phenomenon. It consists of the deterioration of the performance of the design classifier when the number of inputs increases and many noisy variables are involved in fitting the classifier, so the signal gets masked among them and the classification rule confuses it with the noise. There is a great deal of literature discussing this phenomenon; some recent papers are [10, 11], which study the problem within a general framework, and [15] which tackles it in the context of feature selection.

The peaking phenomenon is even more acute for weak marginal / strong bivariate signals as pointed out in [3], that is, for highly predictive interactions whose marginal distributions are uninformative for classifying the output. This paper studies the peaking phenomenon for this type of interaction patterns. We propose a search algorithm for detecting them in high dimensional settings.

The paper is organized as follows: section 2 studies the peaking phenomenon in presence of weak marginal / strong bivariate interactions for a benchmark of classification rules. We conclude that quadratic discriminant analysis (QDA) is a classifier with an acceptable resistance to peaking; in addition, the training process of QDA classification rule has low computational cost in comparison with other more sophisticated rules as the ensembles Random Forests (RF) and Adaboost. In section 3 we use QDA for the design of an

algorithm that utilizes an exhaustive search strategy in the input space in order to detect weak marginal / strong bivariate signals; the internal workings of the algorithm are controlled by the resistance of the QDA classifier to peaking. The efficiency of the algorithm is illustrated both for synthetic data and for a real microarray experiment. We summarize our findings and establish some conclusions in section 4.

2. STUDY OF PERFORMANCE AND PEAKING FOR A BENCHMARK OF CLASSIFIERS

In this section we give a detailed description of how weak marginal / strong bivariate interactions are lost by different classification rules as noise features are added to the input space. For the sake of simplicity we confine ourselves to the binary classification problem, where n_0 and n_1 are the sample sizes for each class.

2.1 Weak marginal / strong bivariate interactions

Four representative examples of weak marginal / strong bivariate interactions are given by the following scenarios.

Scenario 1

The observations are drawn from bivariate normal distributions $Z = (Z_1, Z_1)$ in accordance with the following scheme: $Z|Y = 0$ —black labels— is a bivariate normal variable with means $\mu_0 = (0, 0)$; meanwhile, $Z|Y = 1$ —red labels— is a bivariate normal distribution with means $\mu_1 = (-1, 1)$. Both distributions have the same covariance matrix

$$\Sigma = \begin{pmatrix} 1 & 0.9 \\ 0.9 & 1 \end{pmatrix}.$$

Scenario 2 (XOR)

The observations for the XOR pair $X = (X_1, X_2)$ are drawn from uniform distributions in accordance with the following scheme: $X|Y = 0$ —black labels— has a uniform distribution over $\mathcal{R}_0 = [0, 1] \times [0, 1] \cup [-1, 0] \times [-1, 0]$. On the other hand, $X|Y = 1$ —red labels— has a uniform distribution over $\mathcal{R}_1 = [-1, 0] \times [0, 1] \cup [0, 1] \times [-1, 0]$.

Scenario 3 (circular pattern)

Cases are simultated from a bivariate normal distribution $R = (R_1, R_2)$ with vector means $\mu = (0, 0)$ and covariance matrix the indentity I. The labels are assigned in accordance to the following rules: if $R_1^2 + R_2^2 > 1$ then $Y = 0$ —black labels— and if $R_1^2 + R_2^2 \leq 1$ then $Y = 1$ —red labels.

Scenario 4 (V-shaped pattern)

Observations in this situation are drawn from uniform distributions confined to the domain $\mathcal{D} = [-1, 1] \times [0, 1]$. The interaction between the pair $V = (V_1, V_2)$ and the outcome variable Y is given by the following rules: $V|Y = 0$ — black labels— has a uniform distribution over $\mathcal{D} \cap \mathcal{R}_0$, with $\mathcal{R}_0 = \{(v_1, v_2) : v_2 > |v_1|\}$. On the other hand, $V|Y = 1$ —red labels— has a uniform distribution over $\mathcal{D} \cap \mathcal{R}_1$, with $\mathcal{R}_1 = \{(v_1, v_2) : v_2 \leq |v_1|\}$.

Figure 1 shows the scatterplots obtained by simulating observations in accordance to the four schemes above for sample sizes $n_0 = n_1 = 40$. Note that if the points are projected on each one of the axes, both categories of the outcome do overlap; however, if both variables of the pair are considered together the classes are clearly separated. Therefore, the discrimination comes from the bivariate interaction between them. This is the reason why we call this type of interaction a *weak marginal / strong bivariate interaction.*

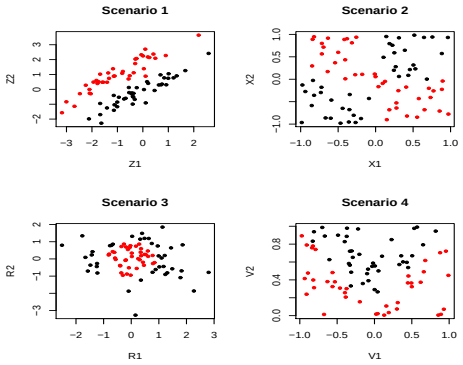


Figure 1: Weak marginal / strong bivariate signals. Sample sizes $n_0 = n_1 = 40$

2.2 The peaking phenomenon

We consider the previous synthetic scenarios and generate samples of sizes $n_0 = n_1 = 40$. For each scenario, we add j independent noisy features, $j = 1, 2, \ldots, 100$, with standard normal distribution and estimate the error rate for the following four classification rules: Adaboost [8], Random Forests (RF) [6], a support vector machine (SVM) with polynomial kernel [14] and QDA [12] . The error rate is estimated by 10-fold cross validation. The results are shown in the plots of figure 2.

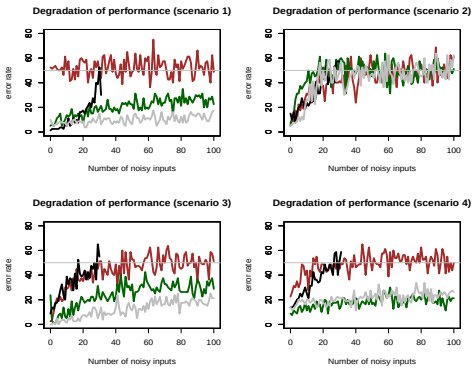


Figure 2: Scenarios $1, 2, 3, 4$ from left to right and top to bottom for the classifiers: SVM with polynomial kernel (brown), Random Forests (green), QDA (black) and Adaboost (gray)

Note that the error rate of the polynomial kernel in scenario 1 is high, around 0.5, when only the variables of the pair (Z_1, Z_2) are used as predictors. This ugly performance is related to the type of polynomial kernel we used, a second order polynomial kernel, which is well suited for tracing nonlinear quadratic patterns but poor for identifying linear class

separation patterns as in scenario 1; see the improvement of SVM classifier in scenarios 2, 3 and 4 where it makes a better job in catching the non linear interaction patterns (X_1, X_2), (R_1, R_2) and (V_1, V_2) describing the separation between the classes.

In addition, we can observe that the error rate deteriorates as the number of inputs increases; this shows the peaking phenomenon for weak marginal / strong bivariate interactions. QDA resistance to peaking compares to RF and Adaboost, with the exception of scenario 3, when $p < 20$; see that the error rates are nearly similar, specially for a number of inputs under 10 or 15. In addition, QDA has the appealing feature of requiring a low computational cost for training the classifier; this fact is crucial in the design of our search strategy since the algorithm will explore the input space in an almost exhaustive way by fitting thousands of times the design classifier.

2.3 Comparative study of design classifiers for a benchmark of classification rules

The CMA package [16] from Bioconductor project repositories, www.bioconductor.org, provides an interface for the analysis of genomic data. One of the utilities of CMA is the possibility to carry out a comparative study of the performance of classifiers for a benchmark of classification rules.

In this section we revisit the effect of the peaking phenomenon for a selection of classifiers from the CMA package: k-nearest neighbors (knn) and neural networks (nnet) [13], diagonal (DLDA), linear (LDA) and quadratic (QDA) discriminant analysis as in [12], partial least squares with lda, logistic regresion and RF variants (pls_lda, pls_lr, pls_rf) as in [17], PAM classifier (scDA) as introduced in [18], random forests [6], the componentwise boosting (compBoost) introduced in [5], the ElasticNet [20] and two versions of the SVM (svm, svm2) with second order polynomial and radial kernels respectively. The error rate was estimated by 10-fold cross validation.

The function `compare` gives a picture of how these classifiers compare one with each other. For each one, it displays the boxplot of the error rate over the 10 validation sets.

The data sets were generated by drawing $n_0 = n_1 = 40$ samples from scenarios 1, 2, 3 and 4. The boxplots below give a glance of the performance of the classifiers. The experiment was made for $p = 2$ (the signal alone), $p = 10$ (signal and 8 noisy inputs) and $p = 20$ (signal and 18 noisy inputs) features.

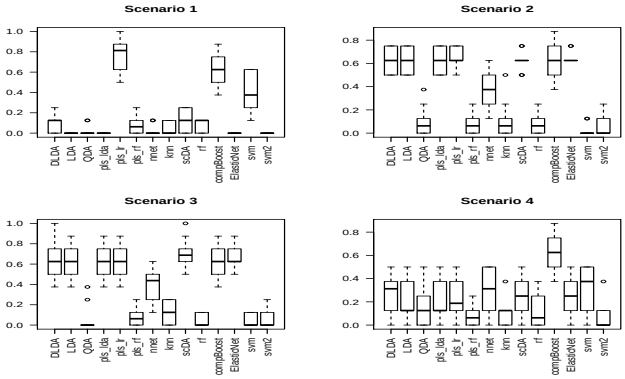


Figure 3: Number of inputs $p = 2$

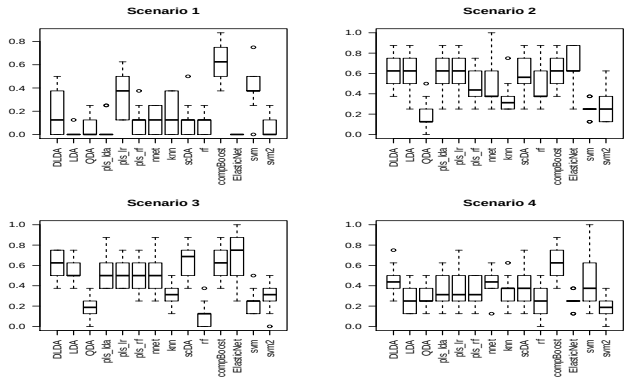


Figure 4: Number of inputs $p = 10$

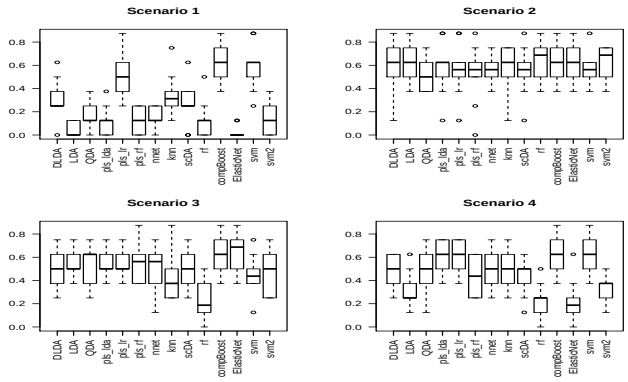


Figure 5: Number of inputs $p = 20$

We can see that QDA outperforms the remaining classification rules in almost all the scenarios. It is of special interest the XOR interaction pattern as pointed out in [7]; in this case, the simulations above and many others not reported here have shown that the performance of all the classifiers deteriorates when the number of inputs reaches $p = 20$. It is worth noting that for p under 10 (see figure 4) the most resistant classifier to peaking for the XOR signal is QDA; meanwhile, for $p = 20$ all the classifiers are highly affected by peaking in the XOR scenario (see scenario 2 in figure 5). So we conclude that QDA is a good candidate for designing a search strategy that uncovers this type of interaction patterns.

2.4 A closer look at QDA

In this section we explore in detail the resistance of QDA classification rule to peaking by means of a simulation study.

Recall that binary QDA is concerned with the discrimination between two p-dimensional multivariate normal class conditional populations $N(\mu_1, \Sigma_1)$ and $N(\mu_2, \Sigma_2)$, where μ_1 and μ_2 are the mean vectors and Σ_1 and Σ_2 are the covariance matrices. The decision boundary corresponding to QDA classification rule is given by

$$\frac{1}{2}x'(\Sigma_1^{-1} - \Sigma_2^{-1})x + (\mu_2'\Sigma_2^{-1} - \mu_1'\Sigma_1^{-1})x + k + \log\left(\frac{\pi_2}{\pi_1}\right) = 0,$$

with $k = \frac{1}{2}\log\left(\frac{|\Sigma_1|}{|\Sigma_2|}\right) + \frac{1}{2}(\mu_1'\Sigma_1^{-1} - \mu_2'\Sigma_2^{-1})$ and π_1, π_2

the a priori class probabilities. When we are dealt with unbalanced classes $\pi_1 = \pi_2 = 0.5$.

When the sample estimates $\hat{\Sigma}_1$, $\hat{\Sigma}_2$, $\hat{\mu}_1$, $\hat{\mu}_2$ of the covariances and the means are plugged in the expression above, we obtain the QDA design classifier.

The decision boundary of QDA defines an hyperquadric whose shape depends on the elements involved in the difference of inverses $\hat{\Sigma}_1^{-1} - \hat{\Sigma}_2^{-1}$, more specifically on the product of its eigenvalues. This yields to elliptical, hyperbolic, parabolic or linear boundaries. Thus, the variety of patterns recognized by QDA is rich enough to consider it a good classification rule for pattern discovery.

We now carry out a simulation experiment in order to study its resistance to peaking for weak marginal / strong bivariate patterns.

QDA resistance to peaking

We have drawn 80 observations ($n_0 = n_1 = 40$) according to patterns in scenarios 1, 2, 3 and 4, along with 80 cases from $p - 2$ independent standard normal variables, which are uninformative inputs for class prediction. On the order hand, we generated $n_0 + n_1 = 80$ samples from p independent standard normal variables and obtain a data set with only noisy inputs. The error rate of QDA was estimated by 10-fold cross validation for both data sets. We repeated the experiment $B = 100$ times in order to get both populations of error rates: with the signal and with only noisy features. We have considered a number of $p = 2, 5, 10, 15, 20, 30$ predictors.

The boxplots of figures 6 and 7 show the results of the simulations.

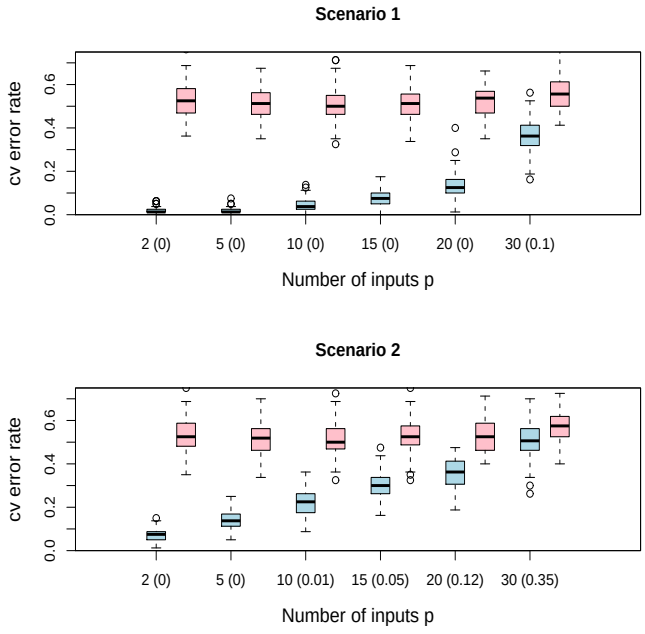


Figure 6: With signal (blue). Only noise (pink)

In parenthesis the amount of overlap between both populations is shown for each p; it is given by the well known measure $\Phi(-\Delta/2)$, with Φ the distribution function of the standard normal variable and Δ the Mahalanobis distance.

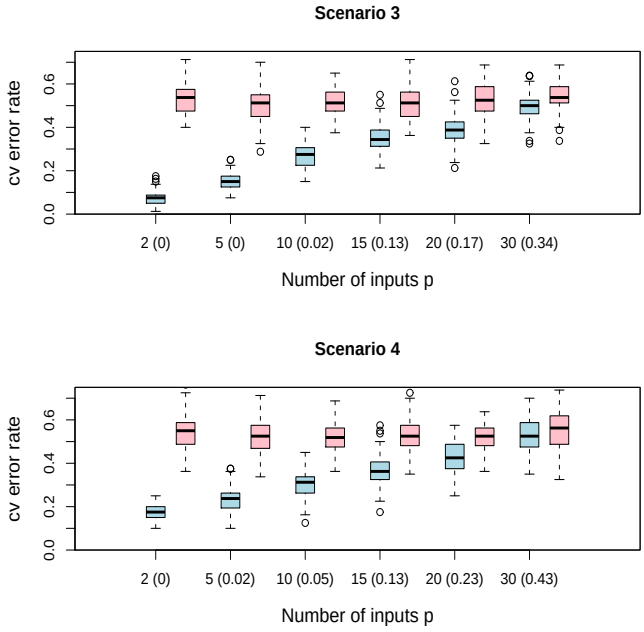


Figure 7: With signal (blue). Only noise (pink)

Note that the amount of overlap between both populations is always less than 5% when the number of predictors is smaller than 10, which means that QDA is able to distinguish between chunks of inputs with a weak marginal / strong bivariate signal and chunks with only noisy features, provided that the size of the chunk is not greater than 10. As p increases, the amount of overlap becomes larger; therefore QDA would be unable to catch the signal and might confuse it with the noise.

3. BIVARIATE INTERACTION DETECTOR ALGORITHM

The results of the previous simulation study show that QDA resistance threshold to peaking can be set at $p = 10$ (or at most $p = 15$), when we are concerned with the detection of weak marginal / strong bivariate interactions in high dimensional data sets. This statement is crucial and puts the basis for the design of a search strategy for this type of interaction patterns. The rationale behind this strategy is as follows.

The naive solution for detecting this type of interactions would explore the input space in an exhaustive way by fitting a QDA classifier to each pair of variables; a high accurate classification would be highlighting the presence of a signal. Obviously, this alternative is time consuming prohibitive as would require a total of $p(p-1)/2$ QDA fits; for example, if $p = 2000$ then 1999000 fits are needed.

Our search strategy proceeds in a nearly exhaustive way by dividing the input space in small blocks of inputs of a specified size *bsize*. As we know that QDA is resistant to peaking for a number of inputs between $p^* = 10$ and $p^* = 15$, we could take *bsize* such that $2 * bsize \leq p^*$; in this way we are protecting ourselves against the danger of peaking when two blocks of features are matched and the QDA classifier

with all the inputs contained in the matching is fit. Once QDA classifiers are obtained for all the possible matchings of blocks, we know that for a matching containing a bivariate interaction pattern, the classifier will give a very low error rate; meanwhile, for a block matching with only noisy inputs we will obtain a high error. Thus, we can construct a ranking of block matchings, with the top ranked matchings containing the bivariate interaction patterns, and the matchings at the bottom of the ranking carrying on only noisy features. Now, at a second stage we can restrict the search to the subset of features belonging to the top ranked matchings of blocks. For example if we confine the search to the $2 * bsize$ inputs of the first block matching, we would need to explore $bsize \times bsize$ interactions in order to find out which one of them is responsible for the observation of such a low error rate in the QDA; usually this search is very low time consuming since $bsize$ is never greater than 7.

Searching in the input space in a blockwise manner has an enormous advantage with respect to the exhaustive search; for example, if $p = 2000$ and we take $bsize = 5$, we would obtain 400 blocks; so the search would need only 79800 QDA fits, much less than the 1999000 fits of the naive solution.

The following algorithm summarizes the steps of the previous search strategy.

QDA bivariate interaction detector algorithm

Let $\mathcal{F}$ be the set of predictors. Set the value for $bsize$ (usually $bsize = 5, 6, 7$)

Step 1. Divide $\mathcal{F}$ in blocks of $bsize$ inputs

Step 2. For every pair (i, j) of blocks, match i and j. Fit a QDA classifier and compute $err(i, j)$ cv error rate

Step 3. Rank the block matchings in accordance to $err(i, j)$

It is recommended to carry out a first screening step that filters the strong marginal inputs highly correlated with the outcome before applying the algorithm. Recall that the search strategy was designed to uncover weak marginal / strong bivariate interactions which are difficult to detect by traditional sequential search procedures (see [3] for a detailed explanation of this fact).

3.1 A simulation study for synthetic data

Let $n_0 = n_1 = 40$ be the class sizes. The cases were drawn from p-dimensional random vectors, (Z_1, Z_2, E), (X_1, X_2, E), (R_1, R_2, E) and (V_1, V_2, E) corresponding to scenarios 1, 2, 3 and 4, with $E = (E_1, \ldots, E_{p-2})$ a vector of independent noisy standard normal variables which were added to the signal. For $p = 200$ the signal represents 1% of the 200-dimensional input space.

We have applied QDA interaction detector algorithm with $bsize = 5$ to the previous synthetic scenarios and have obtained a ranking of block matchings; this ranking is a useful tool that allows to restrict the search for the hidden interaction patterns by exploring its top ranked positions.

Figure 8 displays the heat map of errors of QDA classification rule for all the bivariate interactions obtained from the first position of the ranking of block matchings. Light yellow and orange shades represent a high error rate and the red color represents a low error rate.

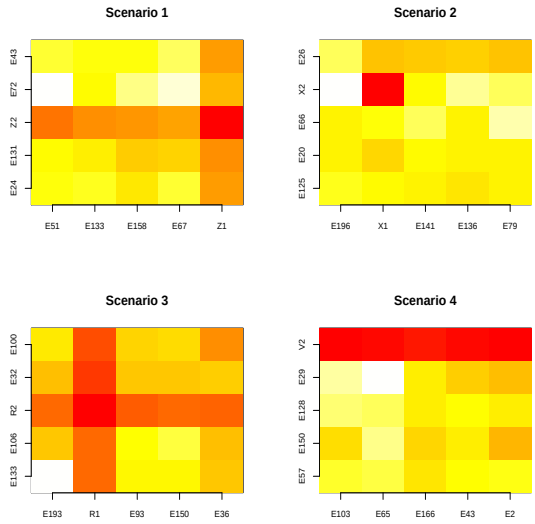


Figure 8: Heat map QDA error matrix

Note that the algorithm was able to locate interaction patterns (Z_1, Z_2), (X_1, X_2) and (R_1, R_2) at the fist position of the ranking of block matchings. The red hot squares of scenarios 1, 2 and 3 in figure 8 highlight the weak marginal / strong bivariate interaction hidden in the matching. The results are not so optimistic for the V-shaped pattern, where only V_2 input came out (see the first row of red squares in scenario 4). A reasonable explanation of this fact is the not so weak marginal effect of the components of the V-shaped pattern if we compare it with the components in (Z_1, Z_2), (X_1, X_2) and (R_1, R_2).

3.2 An application to a microarray experiment

The colon cancer data set is a publicly available experiment which can be obtained from the package colonCA of R project (www.r-project.org). Gene expression levels for 2000 genes across 40 tumor and 22 normal tissue samples were collected with Affymetrix oligonucleotide arrays [1]. The data were preprocessed using a log transformation and standardization across genes.

Random Forests (RF) outlier detector utility identified cases $18, 20, 52, 55$ and 58 as outliers. These were previously identified in [2] as aberrant observations and will be removed from the analysis.

Screening stage

The table of variable importance of RF identifies the most influential genes for class prediction. We took as a measure of importance the mean decrease Gini score; we utilized the values $ntree = 5000$ for the number of trees in the forest and the default for the number of eligible splitters $mtry$. Figure 9 shows the screeplot of variable importance. After a deep decay, we find a long flat behavior; the change of the pattern in the curve is placed at the elbow located at position 100 of the ranking.

We pick up the first one hundred genes of RF table of variable importance. The list has a great agreement with other previous selections in the literature, in particular with

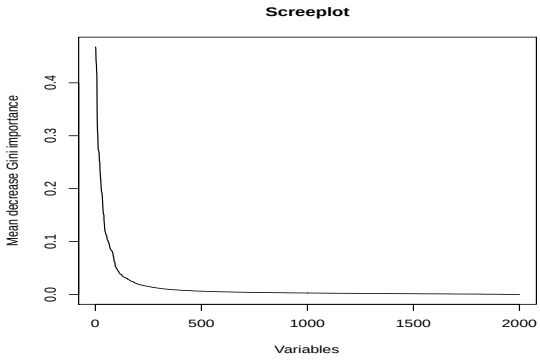


Figure 9: Screeplot of variable importance. Elbow at position 100

that one in [4]. The following table shows the identifiers of the genes of the list (four biomarkers identified as *control* are not reported).

M76378	M63391	M76378	M36634	R87126	J02854	Z50753
M76378	H43887	T92451	J05032	R36977	X12369	X63629
T71025	H40095	Z49269	R44301	M22382	X14958	U25138
R78934	H06524	T86473	H77597	H64489	M64110	X12671
Z49269	X86693	L05144	U19969	M26697	T40454	H20709
X54942	T51534	X16356	X70326	R42501	X87159	D25217
Z24727	R08183	L07648	H08393	U31525	M36981	M26383
X74295	T51571	R48303	T95018	T67077	M80815	U22055
T86749	R46753	X07290	T51539	T60155	U17899	U32519
D31716	H20426	D16294	U09564	R28373	R64115	X12466
R44418	X53743	U14631	X53461	R37276	D31885	X56597
T96873	X15882	T94350	X12496	D59253	D29808	R75843
L41559	T40645	M69135	U26312	T51858	R60883	R84411
Z25521	M26683	D42047	D15049	D14662		

Table 1: Subset of genes obtained from RF screening step

We will put them aside and retain the remaining ones for the application of QDA interaction detector algorithm.

Application of QDA interaction detector

After putting aside the biomarkers identified in the screening step and eliminating a few duplicated columns, we end up with a data set containing 1891 inputs along with the binary outcome. We now apply QDA interaction detector algorithm in order to uncover the weak marginal / strong bivariate signals; these patterns will not be detected by screening approaches like the previous one because they are based on rankings of individual variable importance.

We set $bsize = 5$. Figure 10 displays the heat map plots of the error rate given by QDA classification rule for all the bivariate associations obtained by pairwise matching of the variables belonging to the six top positions of the ranking of block matchings.

The heat map plots reveal several interesting interactions among inputs which were considered as uninformative by RF ranking of variable importance at the initial screening step. Four bivariate interactions are standing out; they correspond to the interaction of the genes at columns: $(G1334, G1573)$,

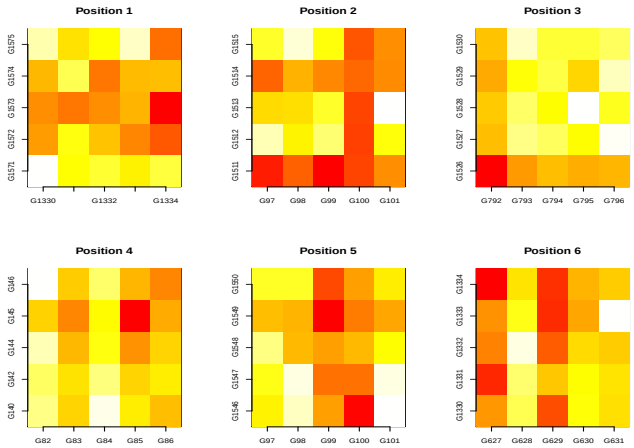


Figure 10: Heat maps of QDA error rate for the bivariate interactions obtained from the six top positions of the ranking of block matchings

$(G792, G1526)$, $(G85, G145)$ and $(G99, G1549)$ which come from the first, third, fourth and fifth positions of the ranking of block matchings. These pairwise associations correspond to the following gene to gene interactions: $(H72234, D29641)$, $(R88740, H05899)$, $(T68848, H48072)$ and $(D45887, H11084)$. The scatterplots in figure 11 contain the type of bivariate interaction pattern displayed by each gene to gene association.

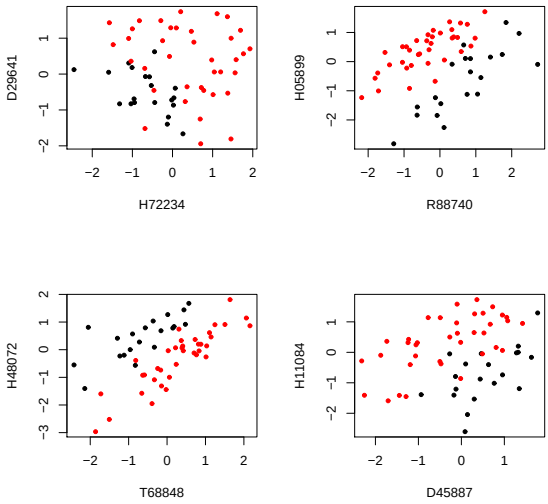


Figure 11: Sscatterplots for gene interactions

Note that none of the genes in the scatterplots exhibit univariate differential expression since both classes of the outcome do overlap for the projections on the axes. However, if both genes are considered together, they discriminate the binary outcome; such discrimination stems from the bivariate association between them. These are four examples of weak marginal / strong bivariate gene interaction patterns uncovered by our algorithm.

4. SUMMARY AND CONCLUSION

This paper has explored the peaking phenomenon for weak marginal / strong bivariate interactions. We have seen that the performance of the design classifier deteriorates as the number of predictors increases. A comparative study for a benchmark of classification rules concluded that QDA has appealing properties in regards to both the resistance to peaking and the computational cost involved in fitting the classifier; this is the reason why we have chosen QDA for the design of the search strategy that uncovers the weak marginal / strong bivariate signals hidden in data. The search takes advantage of the resistance of QDA to peaking by dividing the input space in small blocks of predictors. The algorithm matches all the blocks and computes the cv error rate of a QDA classifier with predictors the inputs in the matchings; the algorithm looks for matchings with high predictive accuracy, as they are expected to contain a hidden bivariate signal. Once they are located at the top positions of the ranking of block matchings, it is relatively easy and low time consuming to explore all the bivariate interactions between the predictors within a matching in order to highlight the interaction responsible for observing the low error rate given by the QDA rule trained with the inputs in the matching.

The algorithm was applied both to artificial data and to a real microarray gene expression experiment, the colon cancer data set. The application to real data has led to promising results providing gene interactions that exhibit bivariate differential expression but are not differentially expressed when considered marginally. The results show the usefulness of QDA interaction detector algorithm, which is expected to become an efficient tool for biologists and bioinformaticians for the discovery of new gene to gene interactions.

QDA interaction detector algorithm was implemented using R 2.10.1 code (www.r-project.org).

The proposed method has been developed for binary classification; the analysis for multi-class problems is a natural extension for conducting future research efforts. Some research regarding the computational cost involved in the QDA interaction detector algorithm is also an issue for further improvements.

5. REFERENCES

[1] U. Alon, N. Barkai, D. Notterdam, K. Gish, S. Ybarra, D. Mack, and A. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues by oligonucleotide arrays. *PNAS*, 96:6745–6750, June 1999.

[2] C. Ambroise and G. J. McLachlan. Selection bias in gene extraction on the basis of microarray gene-expression data. *PNAS*, 99(10):6562–6566, May 2002.

[3] J. M. Arevalillo and H. Navarro. Using random forests to uncover bivariate interactions in high dimensional small data sets. In *StReBio '09: Proceedings of the KDD-09 Workshop on Statistical and Relational Learning in Bioinformatics*, pages 3–6, New York, NY, USA, 2009. ACM.

[4] A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer, and Z. Yakhini. Tissue classification with gene expression profiles. *Journal of Computational Biology*, 7:559–583, 2000.

[5] P. Bühlmann and B. Yu. Boosting with the l_2-loss: Regression and classification. *Journal of the American Statistical Association*, 98:324–339, 2003.

[6] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, October 2001.

[7] M. Dettling, E. Gabrielson, and G. Parmigiani. Searching for differentially expressed gene combinations. *Genome Biology*, 6:R88, September 2005.

[8] Y. Freund and R. E. Schapire. Experiments with a new boosting algorithm. In *ICML*, pages 148–156, 1996.

[9] D. Geman, C. d'Avignon, D. Q. Naiman, and R. L. Winslow. Classifying gene expression profiles from pairwise mrna comparisons. *Statistical Applications in Genetics and Molecular Biology*, 3(1), 2004.

[10] J. Hua, Z. Xiong, and E. R. Dougherty. Determination of the optimal number of features for quadratic discriminant analysis via the normal approximation to the discriminant distribution. *Pattern Recognition*, 38(3):403–4212, 2005.

[11] J. Hua, Z. Xiong, J. Lowey, E. Suh, and E. R. Dougherty. Optimal number of features as a function of sample size for various classification rules. *Bioinformatics*, 21(8):1509–1515, 2005.

[12] G. J. Mclachlan. *Discriminant Analysis and Statistical Pattern Recognition*. John Wiley & Sons, Cambridge, MA, USA, 1992.

[13] B. D. Ripley. *Pattern Recognition and Neural Networks*. Cambridge University Press, 1996.

[14] B. Scholkopf and A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

[15] C. Sima and E. R. Dougherty. The peaking phenomenon in the presence of feature-selection. *Pattern Recognition Letters*, 29:1667–1674, 2008.

[16] M. Slawski, M. Daumer, and A.-L. Boulesteix. Cma - a comprehensive bioconductor package for supervised classification with high dimensional data. *BMC Bioinformatics*, 9, 2008.

[17] K. Strimmer, A. laure Boulesteix, and A. laure Boulesteix. Partial least squares: A versatile tool for the analysis of high-dimensional genomic data. In *Briefings in Bioinformatics*, pages 32–44, 2007.

[18] R. Tibshirani, T. Hastie, B. Narasimhan, and G. Chu. Class prediction by nearest shrunken centroids, with applications to dna microarrays. *Statistical Science*, 18(1):104–117, 2003.

[19] L. Xu, A. C. Tan, D. Q. Naiman, D. Geman, and R. L. Winslow. Robust prostate cancer marker genes emerge from direct integration of inter-study microarray data. *Bioinformatics*, 21(20):3905–3911, 2005.

[20] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B*, 67:301–320, 2005.

Protein Subcellular Localization Extraction and Prediction from PubMed Abstracts

Yifeng Liu,* Zhaochen Guo, Grzegorz Kondrak
Department of Computing Science
University of Alberta
Edmonton, Alberta, Canada, T6G 2E8
{yifeng, zhaochen, kondrak}@cs.ualberta.ca

ABSTRACT

Predicting protein subcellular localization is an essential step for annotating novel protein sequences. When protein sequences are deposited into UniProtKB, they are often associated with PubMed abstracts, and the abstracts can provide additional information to predict the protein subcellular localization. Our work focuses on extracting and predicting protein subcell labels from a query protein's associated PubMed abstracts. We explore two categories of methods for this task: match-and-resolve and supervised classification. In the match-and-resolve approach, we first match the original PubMed abstracts as well as the recognized biomedical named entities with GeneOntology (GO) terms and synonyms; we then resolve the matched terms among the GO hierarchy to their corresponding subcell labels. In supervised classifications, we classify proteins based on features extracted from abstracts: bag-of-words and MeSH terms from abstracts, as well as GO terms extracted using the GOPubMed algorithm. In general, supervised classification outperforms the match-and-resolve approach. However, supervised classification is limited by the availability of training data as well as the size of the feature space, while match-and-resolve is always applicable to proteins annotated with PubMed abstracts.

Keywords

protein annotation, subcellular localization prediction, biomedical text mining, associative classifier, suport vector machine

1. INTRODUCTION

High throughput genome sequencing produces an explosion of biological data; however, human effort in annotating newly discovered sequences grows at a much slower speed. Increasingly, the amount of biological data challenges our ability to assimilate them; biologists therefore resort to computational methods for extracting useful information from data rich but information poor protein sequences.

An important question in annotating novel protein is to predict a protein's *subcellular localization*, the location where a protein functions within a living cell. Predicting a protein's subcellular localization (subcell label) helps biologists elucidate protein functions and facilitates such biomedical applications as protein purification and drug target discovery. Many protein sequences in UniProtKB lack subcellular localization annotation but contain references to PubMed abstracts. It is likely that during the investigations for these proteins, biomedical researchers discover and express the knowledge about subcellular localization in the associated abstracts. However, protein annotators still need to go through all the related references to "rediscover" the query protein's actual subcell label. Such task is tedious and time consuming; automating the process of assigning labels to novel proteins (followed by human verification) would greatly increase the efficiency of protein annotation for biomedical research.

Our work focuses on extracting and predicting subcell labels from a protein's related PubMed entries. We tackle the problem with two different approaches: *match-and-resolve* and *supervised classification*. In both approaches, we use the GeneOntology (GO) extensively both as a biological thesaurus and a well-organized biomedical concept hierarchy. Given a query protein, we first retrieve its associated titles, abstracts and MeSH (Medical Subject Heading) terms from PubMed, and then extract various features from the retrieved data and classify proteins to subcell labels using machine-learned classifiers. We can also find the mentioned GO terms from the PubMed titles and abstracts using direct string matching, named entity recognition (NER)-based matching or the GO term extraction method [4]. After extracting the mentioned GO terms, we can map them to subcell labels by *subcell label resolution* in the GO hierarchy. Figure 1 presents a graphical summary of our approaches.

This paper is organized as follows. We first review the related work in Section 2. We then describe our methodologies in Section 3, and present the results for different approaches in Section 4. We discuss some interesting issues in Section 5, and finally we conclude our findings in Section 6.

2. RELATED WORK

Identifying the localization of proteins is key to understanding their function within cells. However, given the increasing number of discovered proteins, human annotation

*To whom correspondence should be addressed.

BIOKDD '10, July 25, 2010, Washington, DC, USA

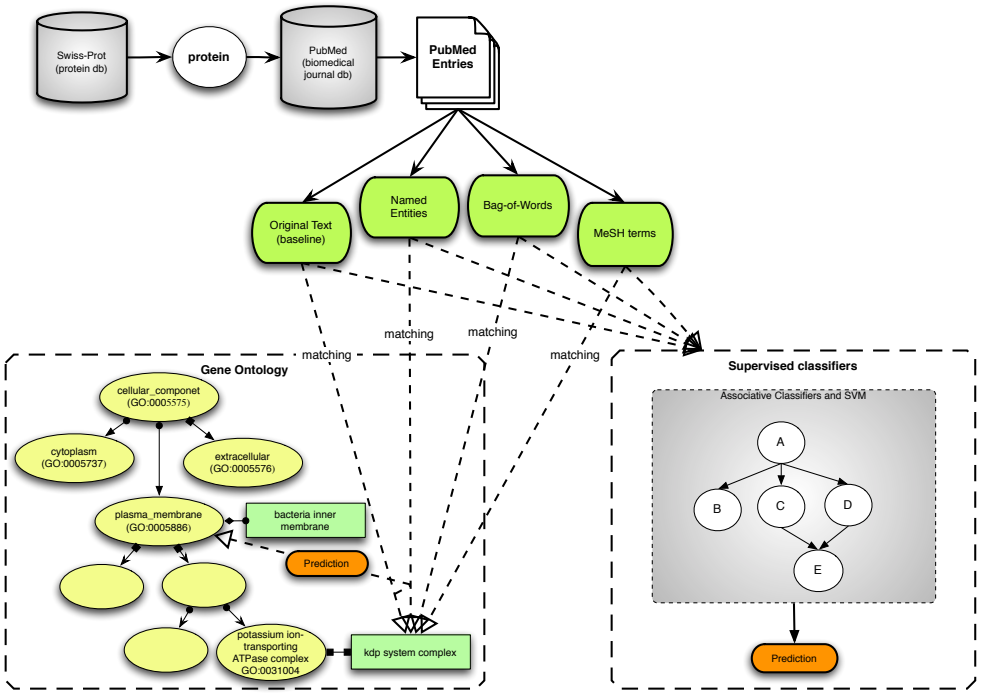


Figure 1: Overall structure of our subcellular localization extraction and classification system.

alone is not sufficient. In the past two decades, biomedical researchers proposed many computational methods to predict protein subcell labels based on primary sequence, amino acid composition and UniProt annotation texts [11]. With the increasing number of digitalized biomedical publications, many recent approaches were proposed to mine useful information about proteins (*e.g.* subcellular localization, GO terms) from biomedical text repository (*e.g.* PubMed) [13] [6].

A straightforward method for protein localization prediction is to first exact the gene and protein names related to the query protein, and then derive the localization from these extracted entities using an ontology knowledge base. A number of systems have been developed for this task — interested readers can consult [2] for a more detailed introduction.

Instead of direct matching, other approaches have been proposed to predict protein localization based on a labeled training set. Höglund *et al.* [7] predicted protein localization by Support Vector Machine (SVM) classifiers based on *most distinguished* terms extracted from PubMed abstracts. They achieved 72%-76% accuracy for their plant and animal datasets using only text features. Fyshe *et al.* [6] also predicted subcellular localization by binary SVM classifiers but based on bag-of-words features weighted by *tf.idf*. Fyshe *et al.* improved the classification results by adding GO synonyms and ancestral GO terms to the bag-of-words features (*Synonym Resolution* and *Term Generalization*). They achieved 94%-96% accuracy on the *Proteome Analyst* datasets. Fyshe *et al.* also proposed a PubMed abstract filtering strategy based on an abstract's referencing protein's subcellular localizations. For more details about abstract filtering see [6]. GOPubMed organizes PubMed entries using the GO hierarchy. Delfs *et al.* [4] extracted the mentioned GO terms from PubMed abstracts by first matching general GO terms and further refining and expanding the matching in the GO hierarchy.

A number of supervised classification methods have been employed for the protein localization prediction task, including SVM, Neural Networks and Bayesian Networks [11]. Particularly noteworthy is *associative classification*, a relatively novel classification method developed by the data mining community [5]. Associative classifiers are considered to be more transparent than SVM, and more accurate and efficient than decision trees. Vadhi and Zaïane used associative classifiers to identify extracellular plant proteins employing sequence-based classification features called the "*partition-based subsequences*"; they achieved 98.83% F-measure on their plant dataset [8].

Our approaches differ from the above approaches in the following aspects. First, in addition to bag-of-words, we also use extracted biomedical named entities and extract GO terms as classification features. Second, we explore the match-and-resolve approach by directly mining the GO terms and resolving subcell labels. Third, we apply associative classification to the problem of protein subcellular localization with the potential for high accuracy and transparent explanation for predictions. Fourth, unlike Fyshe *et al.* who use binary SVM to classify whether the query protein belongs to a single subcellular localization, we use multi-class SVM to classify the query protein as one of many possible subcellular localizations. Finally, we do not rely on feature expansion (*e.g. Synonym Resolution* or *Term Generalization*) for our bag-of-words features.

3. METHODS

In this section, we introduce the bioinformatics databases and datasets used in our experiments and present methods for feature extraction from text and supervised classification.

3.1 Databases

We use three bioinformatics databases extensively in our project:

- **UniProtKB** [14], a protein sequence and annotation database containing over 10 million novel protein sequences and 0.5 million high quality human curated annotations.
- **PubMed** [12], an online biomedical literature database, containing over 14 million newly published and legacy biomedical journal abstracts.
- **GeneOntology** (GO) [3] provides a controlled vocabulary and concept hierarchy of biological terms to reflect the working knowledge of current biomedical discoveries.

All protein annotations, including their associations with PubMed entries, were extracted from SwissProt (part of UniprotKB) version 51.3. PubMed titles, abstracts and MeSH terms were downloaded from the NCBI PubMed website. The GO hierarchy was constructed using the GeneOntology OBO flat-file version 1.2.

3.2 Feature Extraction

We extract four types of features from PubMed entries: bag-of-words, MeSH terms, biomedical named entities and mentioned GO terms.

Bag-of-words

Given a query protein, we concatenate titles and abstracts for all associated PubMed entries into a body of text. We remove standard English stop words and tokenize the text by white space. We then stem each text token using the Porter stemmer[1]. The stemmed tokens form the bag-of-words features for the query protein.

We weight each feature according to the following methods: for associative classification, features are weighted by their presence or absence, *i.e.* a feature receives weight 1 if it is present and weight 0 if it is absent; for SVM, features are weighted with presence/absence and their *tf.idf* (Term Frequency - Inverse Document Frequency) value. *tf.idf* is defined as:

$$tf.idf_{i,j} = \frac{tf_{i,j}}{\sum_k tf_{k,j}} \log \frac{N}{df_i} \quad (1)$$

where $tf_{i,j}$ is the frequency of term i in the PubMed abstracts of protein j. $\sum_k tf_{k,j}$ is the total number of terms in all PubMed abstracts k referenced by protein j. N is the total number of proteins in the dataset, and df_i is the number of proteins with term i in their referencing PubMed abstracts.

Biomedical Named Entities

Biomedical named entities are specialized terms used in the biomedical literature. They may be directly or indirectly related to a protein's subcellular localization, and can be used to help with subcell label prediction. Similar to bag-of-words feature generation, we concatenate related titles and abstracts of each protein into a body of text, and then recognize biomedical named entities using the Named Entity Resolution toolkit in LingPipe[2]. In supervised classifications, named entities are weighted by their presence or absence.

MeSH Terms

Tokenized MeSH terms are simply extracted from PubMed entries without modification. MeSH term features are weighted by their presence or absence.

Mentioned GO Terms

Using the hierarchical structure of the GeneOntology, we can infer the localization labels of a given GO term by recursively examining the GO term's ancestors in the ontology hierarchy. Finding related GO terms for each protein helps localization prediction. Mentioned GO terms are GO terms that match text segments in PubMed titles and abstracts. We implemented the algorithm proposed in GOPubMed [4] for extracting mentioned GO terms for both subcell label resolution and associative classification. In supervised classifications, GO terms are weighted by their presence or absence.

3.3 Supervised Classification

For supervised classification, we used two types of associative classifiers (CMAR [9] and CPAR [15]) and linear kernel SVM. Associative classification is a novel supervised machine learning method that combines *frequent item set mining* and *association rule generation*. An associative classifier finds the features that often co-occur with class labels, and generates classification rules mapping features to class labels. The resulting rules may be pruned to reduce the model size and to increase classification speed as well as accuracy. Associative classification has recently gained popularity thanks to its efficiency and prediction transparency. The technical details of associative classification are beyond the scope of this paper; interested readers are encouraged to consult the survey [5] for more details. SVM classifiers are also popular for classification tasks, and can handle large feature spaces. We include SVM classification results for comparison to associative classifications.

For our experiments, we adapted the source code of the CMAR (*Classification based on Multiple Association Rules*) and CPAR (*Classification based on Predictive Association Rules*) implementations created by Frans Coenen from the University of Liverpool[3]. We used the LIBSVM package by Chang *et al.*[4] for SVM classification. All cross validation results in this paper were optimized with the screening of optimal parameter settings within our limit of computational power. For CMAR, we varied *support* from 0.1%, 1% and 5% to 95% with 5% increments and *confidence* from 10% to 90% with 10% increments. For CPAR, we used only the top 5 rules for classification ($K = 5$) and varied the *minimum best gain* from 0.1 to 0.7 with 0.1 increment and *gain similarity ratio* from 0.7 to 0.99 with 0.05 increments. For LIBSVM,

[1] `http://tartarus.org/~martin/PorterStemmer/`

[2] LingPipe: `http://alias-i.com/lingpipe/`

[3] The CMAR and CPAR source codes are available at `http://www.csc.liv.ac.uk/~frans/KDD/Software/`.

[4] LIBSVM `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

we tried *linear*, *polynomial* and *radial basis function* kernels and found that *linear* kernel was the most robust with the highest accuracy. Using *linear* kernel, we screened for optimal settings of the cost (C) and kernel (γ) parameter using LIBSVM's parameter screening utility.

3.4 GO Term Matching

For extracting mentioned GO terms, we match text segments from PubMed abstracts with GO terms using three different methods: direct string matching, the GOPubMed algorithm, and matching based on named entity recognition (NER).

Direct String Matching (baseline)

Direct string matching is used as our baseline. It finds the GO terms by matching the protein's associated PubMed abstracts with GO term descriptions and synonyms. Instead of performing a character-by-character or word-by-word matching, we build an inverted index to speed up the matching process. The first step is to build an inverted index over all words in PubMed abstracts. Then we go through the entire GO flat-file database and find all the proteins whose abstracts contain the GO terms (the actual matching process). Finally, we re-organize the matching results and retrieve all the GO terms for each protein.

GOPubMed Term Matching

GOPubMed term matching is an approach proposed by Delfs *et al.* [4] in the GOPubMed project. Actual texts seldom match the GO terms perfectly. For example, PubMed entry 1274796 contains text "cAMP-dependent kinase", which corresponds to the GO-term "cAMP-dependent protein kinase activity". In order to handle such cases, GOPubMed first matches the terms from the rightmost (the most general) words, and finds short GO terms as seeds. These seeds are expanded using regular expressions, and then used to locate more specific and longer GO terms. For "cAMP-dependent kinase", we first find the GO term "kinase activity", and expand it using regular expression pattern "cAMP-dependent **.*** kinase activity", which matches the maximal GO-term "cAMP-dependent protein kinase activity". We re-implemented the GOPubMed term matching algorithm for our experiments.

NER-based Matching

Most text segments are unrelated to GO terms, and may not be useful for matching GO terms. Based on this intuition, we explore the NER-based matching approach to extract GO terms. NER-based matching first extracts the biomedical named entities using a named entity resolution method, and then matches the extracted named entities against the GO terms. In our experiments, we used the LingPipe NER module, which had been trained on the GENIA biomedical corpus[5].

3.5 Subcell Label Resolution

After matching GO terms, we resolve them among the GO hierarchy. More specifically, *subcell label resolution* means that given a GO term, we recursively examine its ancestors until we reach one or more of the *target GO terms* and predict the corresponding class label, or reach the root of the GO hierarchy and make no predictions. Table 1 lists the target GO terms and their corresponding class labels used in our subcell label resolution.

Table 1: Target GO terms for subcell label resolution.

Class Label	Target GO Terms	Organism Types
cytoplasm	GO:0005737	all
endoplasmic reticulum	GO:0005783	AN
extracellular	GO:0005576	all
golgi	GO:0005794	AN, FU
lysosome	GO:0005764	AN
mitochondrion	GO:0005739	AN
nucleus	GO:0005634	AN, PL, FU
peroxisome	GO:0005777	AN, PL, FU
plasma membrane	GO:0005886	AN, PL, FU, GN
vacuole	GO:0005773	FU, PL
inner membrane	GO:0005886 GO:0009276	GN
outer membrane	GO:0019867	GN
periplasm	GO:0042597	GN

Table 2: Proteome Analyst dataset statistics.

Organism Type	Class	Protein	PubMed	Word Count
AN	9	12,261	53,549	11 million
PL	9	3,387	6,782	1.5 million
FU	9	2,651	15,752	3.2 million
GP	3	2,594	3,981	0.9 million
GN	5	6,440	15,598	3.4 million

In the example shown in Figure 1, suppose a text segment matches a GO synonym "kdp system complex"; we first relate the GO synonym to its corresponding GO node "potassium ion-transporting ATPase complex" (GO:0031004), and then recursively examine the ancestral nodes for GO:0031004 until we reach one of the defined target nodes, "plasma membrane" (GO:0005886), which implies that the protein is localized in "plasma membrane". If we reach the root of the GO hierarchy without passing through any target nodes, we would make no prediction for the query protein.

3.6 Datasets

We evaluate our approaches using the Proteome Analyst [10] datasets created by Fyshe *et al.* (2008)[6]. Table 2 summaries the datasets, showing the number of classes, proteins, retrieved PubMed abstracts, and total abstract word counts for each organism type. In the rest of this paper, organism types are referenced by their corresponding abbreviations: Animal as AN, Plant as PL, Fungi as FU, Gram-positive Bacteria as GP, and Gram-negative Bacteria as GN.

3.7 Evaluation Metric

We evaluate the classification results of associative classifiers and SVM using stratified 10-fold cross validation. We evaluate the performance of match-and-resolve methods and supervised classifications using *precision*, *recall*, *F-measure* and *percent accuracy* as defined below:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad \text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}},$$

[5] `http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/`

[6] The datasets are available at `http://webdocs.cs.ualberta.ca/~bioinfo/nlp/`.

Table 3: Overall F-measure values for the match-and-resolve approach with different GO term matching methods. Best results for each organism type (excluding upper bounds) are shown in bold; we follow this convention throughout the paper.

Organism Type	Direct Match. (baseline)	NER-based Matching	GOPubMed Extraction
AN	0.212	0.183	**0.298**
PL	0.562	0.354	**0.586**
FU	0.273	0.202	**0.387**
GP	0.158	0.037	**0.782**
GN	**0.274**	0.205	0.161
Average	0.296 (±0.156)	0.196 (±0.112)	**0.443** (±0.244)

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}},$$

where TP stands for *True Positive*, FP stands for *False Positive*, FN stands for *False Negative* and TN stands for *True Negative*.

For evaluating whether the differences between alternative methods or feature sources are statistically significant, we apply two-sample *t-test* (95% confidence) with Welch correction. In the remainder of this paper, "significant" should be understood as "statistically significant". We provide *p-values* for *t-tests* where the results are significant.

4. RESULTS

In this section, we present the results for both match-and-resolve and supervised classifications.

4.1 Match-and-resolve

Table 3 shows the overall F-measure values using various match-and-resolve approaches on different datasets. Baseline F-measure is very low for most organism types. A careful examination of the results reveals that average baseline precision is 0.466, while the average recall is only 0.224. The low recall is partly caused by the "no prediction rate", which is the percentage of proteins without any predictions. The average "no prediction" rate is 62%. We found that the higher the "no prediction" rate, the lower the recall.

To our surprise, the result corresponding to NER-based matching is much worse than the baseline, which can be explained by two factors. First, the training corpus covers only part of GO terms, while the accuracy of the NER approach is not high by itself (state-of-the-art method achieves roughly 0.75 F-measure); therefore NER errors are propagated to our matching process. Second, extracted named entities do not have word contexts that may help with term matching. For example, for the GO term "protein amino acid phosphorylation", the extracted named entity "amino acid" does not match it. This is confirmed by the high "no prediction" rate of 79.3%. As we expected, the GOPubMed algorithm performs better than the other two approaches for most organism types. The overall "no prediction" rate for GOPubMed algorithm is 52%, well below the rate of the other two approaches. However, the results corresponding to the GOPubMed method are not significantly better than either direct string matching or the NER-based matching.

4.2 Supervised Classification

Table 4 shows the overall F-measure values for associative classification with CMAR. The classification results with annotation GO terms (the GO terms extracted from the query protein's UniProt annotation) represent the upper bound of performance, since annotation GO terms are just class labels in disguise. Among all the features we extracted from PubMed entries, MeSH terms achieve the best performance for AN, PL and FU, while bag-of-words features achieve the best performance for GP and GN. For GP, the difference between MeSH and bag-of-words is arguably small; for GN, bag-of-words is better than MeSH by 0.065 in terms of F-measure. We found that neither bag-of-words nor MeSH terms are significantly better than other feature sources. Neither bag-of-words features nor MeSH terms achieve significantly worse results than the upper bounds. On the other hand, both named entity ($p = 0.022$) and GOPubMed ($p = 0.048$) features achieve significantly worse results than the upper bounds.

Table 5 shows the overall F-measure values for associative classification with CPAR. For CPAR, bag-of-words features achieve the best performance for PL, GP and GN, while MeSH terms achieve the best performance for AN. Named entity features outperformed other features types in FU. Similar to the results with CMAR, we found that neither bag-of-words nor MeSH terms are significantly better than other features sources. Unlike classification results with CMAR, none of the four feature types are significantly worse than the upper bounds.

Table 6 shows the overall F-measure values for SVM classification results with various features. For bag-of-words, we have two different weighting schema: binary and *tf.idf* as described in Section 3; other feature types are weighted using the binary schema. Bag-of-words features outperform all other feature types, but the differences in performance are not significant. Similar to the result with CPAR, none of the feature types are significantly worse than the upper bound.

4.3 Performance Comparison

Table 7 shows a comparison between the best performance of our subcell label extraction and predictions methods. We also compare our results to the best performance from Fyshe *et al.* (2008) with bag-of-words features plus feature expansions but without any abstract filtering. Figure 2 shows a graphical view of the comparison. We found that when using simple bag-of-words features *without* any feature expansion, multi-class SVM slightly outperformed the results from Fyshe *et al.* (2008) *with* sophisticated feature expansion in AN, PL and FU; for GP and GN, multi-class SVM is only slightly worse than the results of Fyshe *et al.*. The differences in performance between multi-class SVM and the method of Fyshe *et al.* are not significant.

Comparing the performance of SVM and associative classifiers, we found that for all organism types, SVM classifiers outperformed associative classifiers, although for PL and GP, associative classifiers achieved very similar accuracy. The differences between CMAR, CPAR and multi-class SVM are *not* significant. In summary, associative classifiers are almost as good as SVM on certain datasets. Comparing the performance of supervised classifications and the match-and-resolve approach, we found that for all organism types, supervised classification methods outperformed match-and-resolve methods. All supervised classification methods we used in this paper significantly outperformed match-and-resolve methods (CMAR $p = 0.048$, CPAR $p = 0.031$, SVM $p = 0.014$).

Table 4: Overall F-measure values for CMAR with various features.

CMAR	Bag-of-words	MeSH	Named Entities	GOPubMed GO terms	Annotation GO terms (upper bound)
AN	0.413	**0.723**	0.593	0.407	0.858
PL	0.739	**0.821**	0.722	0.766	0.920
FU	0.469	**0.527**	0.477	0.459	0.773
GP	**0.947**	0.936	0.890	0.945	0.993
GN	**0.741**	0.676	0.580	0.488	0.951
Average	0.662 (±0.219)	**0.737** (±0.154)	0.652 (±0.159)	0.613 (±0.232)	0.899 (±0.086)

Table 5: Overall F-measure values for CPAR with various features.

CPAR	Bag-of-words	MeSH	Named Entities	GOPubMed GO terms	Annotation GO terms (upper bound)
AN	0.640	**0.717**	0.638	0.550	0.873
PL	**0.851**	0.838	0.844	0.767	0.933
FU	0.434	0.529	**0.580**	0.559	0.696
GP	**0.923**	0.898	0.915	0.916	0.998
GN	**0.830**	0.807	0.826	0.743	0.960
Average	0.736 (±0.198)	0.758 (±0.144)	**0.761** (±0.144)	0.707 (±0.154)	0.892 (±0.119)

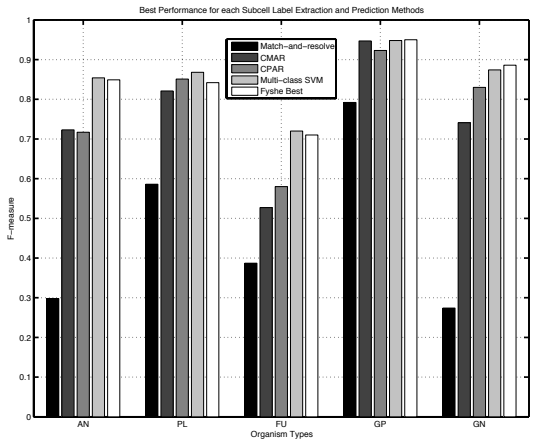


Figure 2: Best performance for each subcell label extraction and prediction methods for each organism types.

5. DISCUSSION

In this section, we discuss the results of both supervised classification and match-and-resolve approaches.

5.1 Supervised Classification

In addition to employing GO terms present in the annotation or extracted from PubMed abstracts as classification features, we experimented with generalizing the current collection of GO terms by adding all ancestral terms (*Term Generalization*). Term Generalization was proposed by Fyshe *et al.* for expanding the feature set of classification instances; we use the technique to increase the likelihood of proteins with same subcell label sharing common GO terms. For example, protein $P1$ is associated with GO term "extracellular matrix" (GO:0031012), while protein $P2$ is associated with GO term "extracellular space" (GO:0043245). "Extracellular matrix" and "extracellular space" are sibling GO terms under a common parent "extracellular region part" (GO:0044421). We hypothesized that without Term Generalization, such terms would have no GO terms in common, thus preventing the associative classifiers from recognizing their association. To our surprise, we found that generalizing GO terms all the way to the GO root node actually *decreases* the F-measure of association classification with extracted GO terms by $0.01 - 0.07$. We speculate that Term Generalization adds too many common GO terms to the feature set, which makes proteins with different labels share common GO terms, and therefore degrades the classification accuracy. In future work, we plan to verify this hypothesis by limiting the generalization through adding only the parent or "grand-parent" terms instead of all ancestral terms.

5.2 Feature Selection for Bag-of-words

The idea of *association rule mining* in associative classification was initially developed for *market basket analysis* in data mining. Associative classifiers were not designed to handle large feature space. Their classification performance depends on the discovery of strong associations between features and class labels, which in turn depends on the result of *frequent item set* mining. When the feature space contains thousands or millions of features, the frequent item set mining procedure becomes computationally expensive or even intractable. Therefore, associative classifier is not particularly suitable for feature types with large feature space such as features generated by the bag-of-words or named entity approaches. However, when the feature space is small enough, as in GP, associative classifier yields comparable performance with SVM.

In order to reduce the feature space for the bag-of-words approach, we use the following feature selection method. We first calculate the $tf.idf_{i,j}$ values for each term i and for each referencing protein j in the bag-of-words feature space. As shown in the definition, a term could have different *tf.idf* values when referenced by different proteins. We sum the $tf.idf_{i,j}$ values for the same term i under all referencing proteins j to produce the term ranking score $tf.idf_i$:

$$tf.idf_i = \sum_j tf.idf_{i,j} = \left(\sum_j \frac{tf_{i,j}}{\sum_k tf_{k,j}} \right) \log \frac{N}{df_i} \qquad (2)$$

We then sort the terms (features) according to their ranking score in non-increasing order and select the topmost $x\%$ features with the highest $tf.idf_i$ ranking score. The intu-

Table 6: Overall F-measure values for SVM Classification with various features

SVM	Bag-of-words binary weighted	Bag-of-words tf.idf weighted	MeSH	Named Entities	GOPubMed GO terms	Annotation GO terms (upper bound)
AN	0.793	**0.854**	0.812	0.774	0.703	0.894
PL	0.827	**0.868**	0.861	0.834	0.802	0.933
FU	0.550	**0.720**	0.603	0.553	0.608	0.709
GP	0.945	**0.948**	0.947	0.946	0.957	0.997
GN	0.842	**0.874**	0.849	0.849	0.830	0.964
Average	0.791 (±0.146)	**0.823** (±0.083)	0.814 (±0.128)	0.791 (±0.147)	0.780 (±0.132)	0.899 (±0.113)

Table 7: Best performance for each subcell label extraction and prediction methods, with comparison to the best result from Fyshe *et al.* 2008 using the same set of PubMed abstracts without any abstract filtering.

Organism Type	Match-and-resolve	CMAR	CPAR	Multi-class SVM	Fyshe *et al.*
AN	0.298	0.723	0.717	**0.854**	0.849
PL	0.586	0.821	0.851	**0.868**	0.842
FU	0.387	0.527	0.580	**0.720**	0.710
GP	0.792	0.947	0.923	0.948	**0.950**
GN	0.274	0.741	0.830	0.874	**0.886**
Average	0.467 (±0.219)	0.752 (±0.154)	0.780 (±0.134)	**0.853** (±0.083)	0.847 (±0.088)

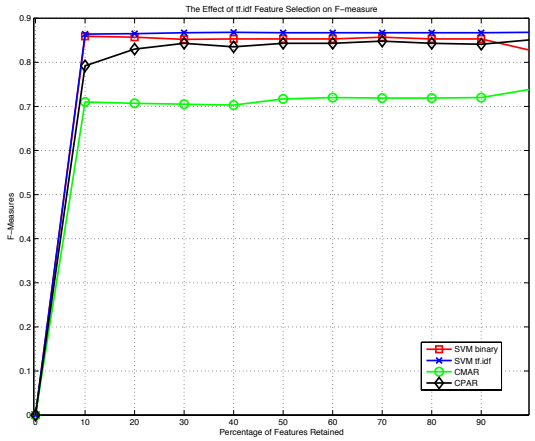


Figure 3: The effect on overall F-measure values for different percentages of features retained using sum *tf.idf* feature selection on the plant organism type.

ition behind the $tf.idf_i$ ranking score is that if a given term (feature) is referenced by many proteins (popular: high overall *term frequencies* $\sum_j \frac{tf_{i,j}}{\sum_k tf_{k,j}}$) or referenced by only few proteins (discriminative: high *inverse document frequency* $\log \frac{N}{df_i}$), or both, then such a term is important to retain as a popular and discriminative classification feature.

In order to experimentally verify the above feature selection method, we varied the percentage x of selected features from 0% to 100% in 10% increments for PL with bag-of-words features. The effect of varying x on the overall F-measure values is illustrated in Figure 3. We were surprised to find that retaining the topmost 10% or 20% features was sufficient to achieve similar performance as using all features. For other organism types with bag-of-words features, we observed similar trends. The overall trend of the lines was flat, indicating that varying the number of selected features did not greatly improve or severely degrade performance. With the above ranking score, we were able to effectively reduce the running time of supervised classification without significant performance degradation.

5.3 Match-and-resolve

We found that removing the default subcell label when more specific subcell labels are predicted improves the performance of subcell label resolution. For example, in animal, plant or fungi cells, organelle "mitochondrion" is physically surrounded by the major subcellular fluid "cytoplasm"; hence in the GO hierarchy, GO node "mitochondrion" (GO:005739) is logically part of "cytoplasm" (GO:0005737). Any protein with prediction "mitochondrion" is also labelled with "cytoplasm". However, according to protein annotation conventions, a protein is labeled as cytoplasmic protein only when it is localized in "cytoplasm" but outside all membrane-bounded organelles, such as "mitochondrion". Therefore, removing the default label "cytoplasm" when a protein is also predicted as "mitochondrion" is deemed appropriate; the label "cytoplasm" is only retained if it is the only predicted label. This subtle but important observation reduces false positives for the default label "cytoplasm", hence increasing the F-measure for PL from 0.383 to 0.586 with GOPubMed GO term extraction. In general, removing default labels increases the overall F-measure by 0.04−0.20.

We evaluated the effectiveness of *subcell label resolution* separately to investigate its impact on the final performance of the match-and-resolve approach. We found that the resolution step is very effective by itself. Using the GO terms present in a protein's Swiss-Prot (part of UniProtKB) annotation, we performed *subcell label resolution* to see whether we can recover the protein's subcellular localization; this method is very similar to the match-and-resolve approach except that the GO terms are given in the query protein's annotation instead of being extracted from PubMed abstracts. We achieved average overall accuracy 95.84% and average F-measure 0.7904 in resolving subcell labels with annotation GO terms. The result indicates that if perfect GO terms are found, we can effectively resolve them to their true corresponding subcell labels; such effectiveness is due to the use of the GO concept hierarchy in resolving subcell labels.

Nevertheless, we did not observe high F-measure values in our experiments for most match-and-resolve methods, indicating that our matching methods could not, in most cases, capture the perfect GO terms. In fact, the inability to rediscover the annotation GO terms from PubMed entries may

not necessarily be attributed to the defects of the matching approaches — the annotation GO terms in UniProtKB are assigned based on a protein's human annotation, which may or may not reflect the content of the protein's associated PubMed entries. Delfs *et al.* [4] show that only 56% PubMed abstracts contain one or more GO terms and on average only 1.76 terms are found per abstract. However, our experiments do not allow us to conclude with certainty whether the poor results are due to defects in our matching algorithms or to the absence of suitable GO terms in PubMed abstracts.

Since the match-and-resolve approach did not achieve good results in GO term extraction, we also tried predicting GO terms from text features such as bag-of-words and MeSH terms. As expected, the classification results were also poor due to the large number of class labels (GO terms). In conclusion, extracting GO terms from PubMed abstracts remains a difficult task.

5.4 Future Work

Increasing the number of training instances would likely improve the localization prediction accuracy. While the amount of labeled training data is scarce and unlabelled data is abundant, we can use the *co-training* [1] technique to annotate unlabelled data using high accuracy prediction methods before training our supervised classifiers. We will further evaluate our approaches with other publicly available datasets, and compare our performance with other state-of-the-art systems such as MultiLoc [7].

6. CONCLUSION

We have described experiments with two different categories of methods for discovering protein subcellular localization from their associated PubMed abstracts: supervised classification and match-and-resolve. In supervised classification, multi-class SVM outperformed associative classifiers CMAR and CPAR for all organism types, although the differences were not statistically significant; multi-class SVM also outperformed, for certain organism types, methods proposed by a previous study with sophisticated feature expansion. We have shown that associative classifiers can approach the accuracy of SVM when the feature space is small. In the match-and-resolve approach, subcell label resolution is effective if perfect GO terms are given; however, the difficulty of extracting GO terms from PubMed abstracts limits the accuracy of the match-and-resolve approach. All supervised classification methods outperformed match-and-resolve methods with various feature sets. However, unlike match-and-resolve methods, supervised classifiers are limited by the availability of training data. Finally, we proposed an effective feature selection method for supervised classification with bag-of-words features; by retaining the topmost 20% features ranked by overall *tf.idf* values, we preserve classification accuracy while greatly improving classification efficiency.

7. ACKNOWLEDGMENTS

The authors would like to thank Sittichai Jiampojamarn for help with biomedical named entity recognition, Xiaodi Ke for help with associative classification, and the Proteome Analyst Research Group for sharing with us their dataset.

8. REFERENCES

[1] A. Blum and T. M. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT*, pages 92–100, 1998.

[2] A. M. Cohen and W. R. Hersh. A survey of current work in biomedical text mining. *Briefings in Bioinformatics*, 6(1):57–71, 2005.

[3] T. G. Consortium. `http://www.geneontology.org/`.

[4] R. Delfs, A. Doms, E. Kozlenkov, and M. Schroeder. GoPubMed: ontology-based literature search applied to geneontology and PubMed. In *In Proceedings of German Bioinformatics Conference. LNBI*, pages 169–178. Springer, 2004.

[5] H. Fadi Thabta. A review of associative classification mining. *The Knowledge Engineering Review*, 22(01):37–65, 2007.

[6] A. Fyshe, Y. Liu, D. Szafron, R. Greiner, and P. Lu. Improving subcellular localization prediction using text classification and the gene ontology. *Bioinformatics*, 24(21):2512–2517, 2008.

[7] A. Hoglund, P. Donnes, T. Blum, H.-W. Adolph, and O. Kohlbacher. Multiloc: prediction of protein subcellular localization using n-terminal targeting sequences, sequence motifs and amino acid composition. *Bioinformatics*, 22(10):1158–1165, 2006.

[8] V. Jazayeri and O. R. Zaiane. Plant protein localization using discriminative and frequent partition-based subsequences. *4th Workshop on Mining Complex Data, in conjunction with IEEE International Conference on Data Mining, Pisa, Italy*, December 15 2008.

[9] W. Li, J. Han, and J. Pei. Cmar: Accurate and efficient classification based on multiple class-association rules. In *Proceedings IEEE International Conference on Data Mining*, pages 369–376, 2001.

[10] Z. Lu, D. Szafron, R. Greiner, P. Lu, D. S. Wishart, B. Poulin, C. Macdonell, and R. Eisner. Predicting subcellular localization of proteins using machine-learned classifiers. *Bioinformatics*, 20(4):547–556, 2004.

[11] K. Nakai and P. Horton. *Computational Prediction of Subcellular Localization*, volume 390 of *Protein Targeting Protocols*. Humana Press Inc., Totowa, NJ, 2 edition, 2007.

[12] PubMed. `http://www.ncbi.nlm.nih.gov/pubmed/`.

[13] H. Shatkay, A. Hoglund, S. Brady, T. Blum, P. Donnes, and O. Kohlbacher. Sherloc: high-accuracy prediction of protein subcellular localization by integrating text and protein sequence data. *Bioinformatics*, 23(11):1410–1417, 2007.

[14] UniProtKB. `http://ca.expasy.org/`.

[15] X. Yin and J. Han. Cpar: Classification based on predictive association rules. In *SIAM International Conference on Data Mining*, 2003.

The Dynamics of the MAPK Network

Ioana Policeanu
Computer Sciences
Florida Tech
Melbourne, Florida, USA
ipolicea@fit.edu

Ronaldo Menezes
Computer Sciences
Florida Tech
Melbourne, Florida, USA
rmenezes@cs.fit.edu

ABSTRACT

Mitogen-activated protein kinase (MAPK) is one of the most important and highly studied signal transduction molecules, being often associated with cancer and auto-immune diseases. The network which carries its name connects cell-surface receptors to specific transcription factors and other regulatory proteins that are responsible for numerous cellular activities including growth, proliferation, cell differentiation and survival. Recent advances in the field of Network Sciences improved our understanding of networks as frameworks to model interactions, such as the ones encountered in the MAPK network. However, even with the development of high-throughput techniques, the complexity of biological networks presents researchers with various challenges, such as sampling from sub-networks and analyzing incomplete datasets. In this paper, we made use of the directed MAPK network reconstructed starting from 1980 data, and the updates recreated every four year following that. Once the present state was reached, we compared all reconstructions to instances from a similar undirected network, as well as a randomized network with the same number of nodes and edges as the directed one, which served as a control. Based on the results of our experiments we concluded that MAPK displays scale-free and small-world characteristics only with newer instances, when the majority of the MAPK network components had been discovered. By analyzing the evolution of the MAPK network over the years and observing the topology and structure transformations that occurred in the network, we were able to pinpoint not only the type of dynamic network MAPK encompasses and its most significant proteins that can be targeted for drug treatments, but also the important fact that completeness might not be achieved in certain networks. This is a crucial aspect when examining biological networks, since scientists assume that the latest maps available are complete and error-proof.

Keywords

MAPK, scale-free networks, signal transduction

1. INTRODUCTION

During the last twenty years, several related intra-cellular signaling cascades have been elucidated; these are collectively known as MAPK (Mitogen-activated protein kinase) signaling cascades. The MAPKs are a group of proteins that are activated in response to a variety of extracellular stimuli, and have been amongst the most studied signal transduction molecules known to have kept constant shape and function over time, in multicellular organisms [4]. The MAPK network connects cell-surface receptors to specific transcription factors and other regulatory proteins that are responsible for numerous cellular activities including growth, proliferation, cell differentiation and survival. Because of its central role in signal transduction when improperly activated, the MAPK network has been repeatedly implicated in the pathogenesis of cancer and autoimmune diseases, leading to its selection as target for drug development. Several inhibitors of the MAPK network have now entered clinical trials, especially for malignant melanoma, Rheumatoid arthritis, Crohn's disease, psoriasis, Parkinson's disease, Alzheimer's disease and hearing loss [3]. However, even with the advances in the development of high-throughput techniques that has led to an explosion of available data, MAPK ability to coordinate and process a variety of inputs from different growth-factor receptors into specific biological responses is still not well understood. As a network, it is surprising that even with the advances in the field of Network Sciences, little analysis has been performed to verify the properties of the MAPK network; these properties can reveal important features of the network that may be used to further advance the understanding of signaling cascades. One way to investigate these global property statistics of the MAPK network is by comparing it to other observed setups, such as the neural network, or the social network and the Internet. The comparison between these multiple-setting networks can provide us with an indication of the robustness of MAPK when looking at the common properties it shares with real-world networks [5, 22].

Recent studies have suggested that biological networks may share the same properties as the real-world networks and do not match those that had been traditionally utilized as modeling tools in networks, such as random graphs [11, 1, 19]; rather they express characteristics of scale-free networks[5], where certain nodes in the network (proteins in the case here) act as hubs connected to a large number of lower degree nodes, thus following a power-law distribution. This leads to the assumption that these hub-proteins may be more important for an organism's survival than those of lower degree. In other studies, it was inferred that bio-molecular networks have small average path lengths between pairs of nodes in comparison to the network's total number of nodes, giving them small-world properties [22]. The average path length in a network serves as an indicator of how readily the information can be transmitted though it. Thus, the small-world properties observed in biological

BioKDD'10 July 25-28, 2010, Washington, DC, USA

networks suggest that such networks are efficient in the transfer of information because only a few interactions are required for any one protein in the network to influence the characteristics or behavior of another [3]. If we can show that the MAPK network has scale-free attributes such as a power-law degree distribution with a relatively small exponent, as well as small-world characteristics, particularly short-path lengths, then targeting hubs may prove to be a sufficient condition for the inhibition of this pathway.

Together with the potential benefits of applying graph theoretical methods in molecular biology, the complexity of the networks encountered in biology presents network researchers with numerous challenges, such as the inherent variability of data, the high likelihood of data inaccuracy with high levels of false-positive and false-negative errors [21], and the need to incorporate dynamics and network topology in the analysis of biological systems. Furthermore, the data analyzed has most often been inferred from sampled sub-networks [3], rather than complete networks. While some studies have pinpointed that the statistical properties of interaction networks may be robust with respect to variations from one data set to another, the impact of sampling and incomplete information on the identified degree distribution is an important issue yet to be grasped [20, 22]. The precise impact of sampling on the results and techniques published in the recent past needs to be well understood, if these are to be reliably applied to real biological data. Some authors [10] have addressed this issues and came up to the relevant conclusion that sub-networks sampled from a scale-free network are not in general scale free, and that it is possible for a sampled sub-network of a network with skewed degree distribution to appear to be scale-free. These represent just a few of the problems scientists have to take into consideration and address when dealing with real biological networks. In this paper, we show that having incomplete information about the MAPK network leads to some variation in analyzing the global property statistics.

2. METHODS AND MATERIALS

2.1 Reconstruction of the MAPK Network

2.1.1 KEGG

The data used in our study was downloaded from the Kyoto Encyclopedia of Genes and Genomes (KEGG) database [16, 23]. Serving as a crucial tool for biological analysis and interpretation of large-scale datasets, KEGG is an integrated database resource encompassing 16 main databases, broadly categorized into systems information, genomic information, and chemical information [16].

The molecular network, shown in the stored pathway on the KEGG server side, is a graph consisting of nodes (proteins, genes, small molecules) and edges (interactions, reactions, relations-activations or phosphorylation) connecting the nodes. The KEGG graphs can be scrutinized to answer biological relevant questions, such as what nodes are crucial targets in finding new cures and treatments for severe or crucial diseases.

For our study, the most recent MAPK pathway, seen in Figure 1, was obtained from the KEGG database and the name of each protein involved in this signaling pathway was recorded in a file. A small script written in Perl was used to retrieve information related to the discovery date (year) of each enzyme. This was necessary for the reconstruction procedure, since KEGG database repository does not include a versioning system that could allow us to access older instances of the network.

2.1.2 R

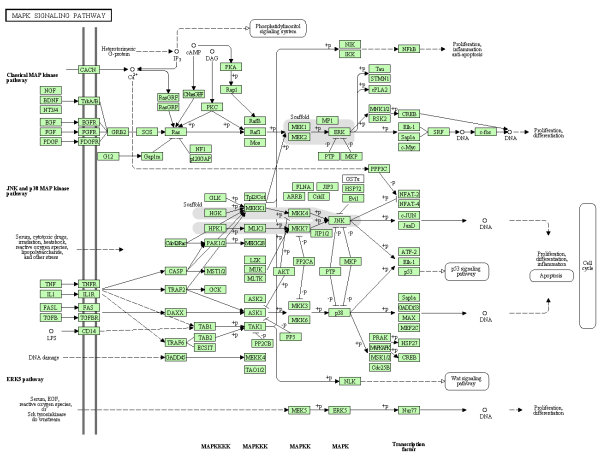


Figure 1: The most updated KEGG PATHWAY entry for the MAPK Signaling Pathway obtained from the KEGG database.

Once we imported the data from KEGG, we analyzed it with R [15]. R is a high-level interpreted language in which one can easily and quickly prototype new computational methods. R is gaining widespread usage within the bioinformatics community and many other bioinformatics projects and researchers have found R to be a good language and toolset to work with. The infrastructure in R that is used to support remote robustness analysis could be implemented in other languages such as Perl and Python [8]. Through its flexible data handling capabilities and well-documented application programming interface (API), it is easy to link it to other applications, such as databases, web servers, numerical or visualization software [15].

2.1.3 Bioconductor

There were many existing, well-tested and high-quality implementations of graph algorithms, but the one that provides a large collection of software for the analysis of functional genomic data and biological networks, is Bioconductor [8]. Bioconductor is an international open source and open development software project for the analysis and comprehension of genomic data whose main engine is R. The software is organized into functions and packages, and runs on all major computing platforms.

R and Bioconductor include powerful tools for graph algorithms and operations, including graph, Rgraphviz and RBGL which are infrastructure packages. Using Bioconductor, we were able to reconstruct yearly instances of the MAPK network, starting from 1980, when only around ten proteins had been discovered. Since we had already created a file containing the discovery time of each protein, it was relatively straightforward to create yearly network instances. The proteins that were described for the first time in a particular year were included in the network, but not necessarily connected right away. One assumption made was that a protein was connected to the existing network only if the subsequent publications following its discovery were at most two years apart. For example, *Tau* was first mentioned before 1980, but the following publication naming it dates from 1991. This suggests that *Tau* was discovered before 1980, but the scientific community did not link it to the MAPK pathway until 1991. A possible explanation could be that since the MAPK network became one of the most studied signaling pathways starting with 1990, when technological progress made possible the massive exploration of the molecu-

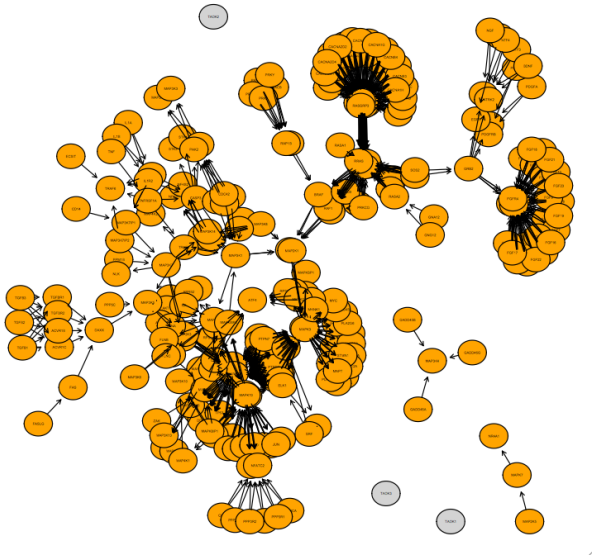


Figure 2: The same MAPK signaling pathway as in Figure 1, represented in R and Bioconductor using the KEGGgraph package and Rgraphviz package.

lar world, most if not all of the proteins involved were either discovered when analyzing the MAPK cascade or were connected to the MAPK network through newly seen interactions only after that year.

The current directed MAPK pathway displayed in Figure 1 was converted to the graph format and shown in Figure 2. For analytical purposes, an undirected network and a randomized network were created additionally. The *ugraph* function allows us to remove the directionality of the initial network, which then allowed the comparison between the two networks. This is important when analyzing the degree distribution of the node proteins. In directed networks, proteins will be measured based on their in-degree, as well as out-degree, which, in scale-free cases follow a power-law distribution with a negative exponent, when plotted against frequency. Biologically, the out-degree (number of out-going edges) reflects the regulatory role, while the in-degree (number of in-going edges) suggests the subjectivity of the protein to intermolecular regulations [15, 23]. In undirected networks where there is no directionality, the mean degree distribution is computed instead. While the undirected network can share similarities between some of the global statistical results, such as the betweenness centrality coefficient, it is necessary to analyze a randomized version of the MAPK instances in order to attest that these results are robust with respect to protein-protein interactions. If the directed MAPK network displays the same properties as its randomized version, then the analysis performed on the directed network would yield similar results even when the links between these proteins are mixed randomly. This would mean that the MAPK directed network is random and not scale free. The randomized network for this project was build by specifying the number of nodes and edges existing in the largest directed network, thus starting with a randomized graph for the 2010 MAPK network. For each of the earlier instances, we removed the nodes that were not present in the previous directed version of that instance from the randomized network, while keeping the same number of edges as in the directed version. This subtraction of nodes was performed until the number of nodes and edges from the randomized instance was equal to the number of nodes and edges in the directed instance.

2.1.4 Gephi

Although most of the network analysis has been performed using R and Bioconductor, we were unable to retrieve the power law exponent and the average path length. Therefore, another powerful tool for network analysis, Gephi [7], was used. Similar to R, Gephi is an open source software that specializes in assaying graphs and networks. Gephi contains numerous modules that import, visualize, spatialize, filter, manipulate and export all types of networks information, especially when dealing with the study of dynamic networks.

2.2 Network Measurements

Once all the instances of the directed, undirected and randomized networks were obtained, a thorough diagnosis of each network was calculated to verify the assumption that the MAPK signaling pathway follows the laws of scale-free and small-world networks. Since all instances describe interactions between proteins as nodes, the degrees of nodes constitute an important measure when analyzing networks. Thus, the degree distribution together with the density, and average degree were investigated. The degree distribution $P(k)$ gives the fraction of proteins with k interactions in the network.

We also computed the average path length between all possible pairs of proteins and the diameter of the network, which is a measure of maximum shortest path between any given two proteins. The average path length estimates how easily information can flow from one node to another and represents one of the most important factors when looking for small-world networks. When observing biological networks, the small world property infers that these type of networks are efficient in the transfer of biological information [17]. The clustering coefficient C is another important measure which is needed to characterize small-world networks, as well as to quantify the likelihood of neighboring nodes to be linked. It can be defined according to the formula

$$C = \frac{2w}{n(n-1)},$$

where w represents the number of links between n neighbors [6]. The presence of a large clustering coefficient is usually a good indicator of interaction between neighboring nodes and attests to the presence of scale-free topologies.

With regards to the existence of hubs, whenever highly connected nodes (hubs) appear, the network displays scale-free topology. This type of network has a power-law distribution given by

$$P(k) \approx k^{-\lambda},$$

and is extremely robust against failures of random components. The λ exponent determines the importance of hubs within these networks; the smaller the λ value is, the larger the fraction of nodes connected to hubs is, and the more important the hubs are [6]. Therefore, in order to determine which nodes act as representatives for the MAPK network, betweenness centrality coefficient (BCC) is needed. However, Bioconductor reports the relative betweenness centrality coefficient (RBCC) that is calculated from scaling the BCC by a factor of $(n-1)(n-2)/2$, where n represents the number of nodes in the network.

The interactions between nodes can also be measured by the presence of cliques, subsets of nodes where each node is connected to every other node. In biological networks, cliques can be seen

as groups of interrelated proteins that influence the activity of one another. Finding protein cliques can be extremely important in medicine, especially when targeting certain proteins which are responsible for fatal diseases. Therefore, this study assessed the size maximum clique as well as the number of these cliques.

3. EXPERIMENTAL RESULTS

The reconstruction of the MAPK instances began with the year 1980 and ended in 2000, after which no new proteins have been added to the network. This can be seen in Figure 3, where the count for newly discovered proteins is plotted against time. Most proteins in the MAPK network were identified between 1988 and 1998, during a ten-year time span. As stated before, yearly instances were considered and rebuilt, however the analysis for this project took into account every four-year change beginning with 1980. The rationale behind this fact is that certain instances contain no changes when compared to the previous states, thus their analysis would have yielded redundant results. Another particularity is that this project considered the MAPK interaction with other molecular pathways, which is described in this study as the instance for 2010. This extra instance accounted for the hypergraph of the MAPK network and presented additional information regarding the properties of this network. Overall, seven instances were taken into account, for the year 1980, 1984, 1988, 1992, 1996, 2000, and 2010.

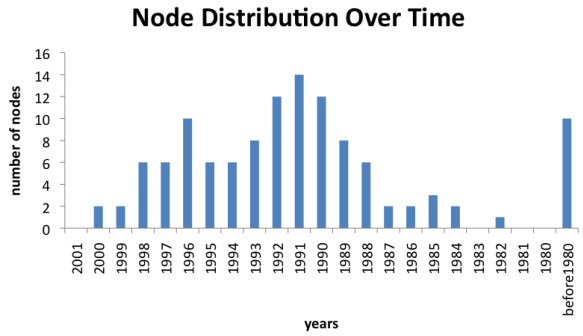


Figure 3: Node distribution plotted against time. Each column of the graph represents the number of newly discovered proteins that are part of the MAPK network.

Table 1: General information about each of the seven network instances. Values for the average degree are rounded to 3 decimal points.

Year	1980	1984	1988	1992	1996	2000	2010
Number of nodes	10	13	26	72	102	124	265
Number of edges	8	11	19	91	148	168	876
Average degree	0.8	0.846	0.731	1.264	1.451	1.355	3.306

Using the KEGGgraph package we analyzed each of the seven instances. A summary of the findings is shown in Table 1. Originally, we started with a network having 10 proteins linked by 8 interactions, which developed into a graph representation with 265 proteins connected by 876 interactions. The average degree was also computed, as the number of edges divided by the total number of nodes. The MAPK hypergraph displayed the highest value of 3.30, an expected result since the number of edges in the graph is much higher than the number of nodes.

A much more detailed analysis, containing the results of all the statistical tests mentioned in Section 2.2 is revealed in Table 2. The mean degree and the max k-clique results in the directed network are not available, because the Bioconductor packages *graph* and *RGBL* are only able to calculate them for undirected networks. For the same reason, the number of k-cliques is omitted from the MAPK directed network. However, we calculated the in-degree and out-degree for the MAPK directed network, although the results were not included in the paper, and found that only a small fraction of the nodes have high in-degree and high out-degree.

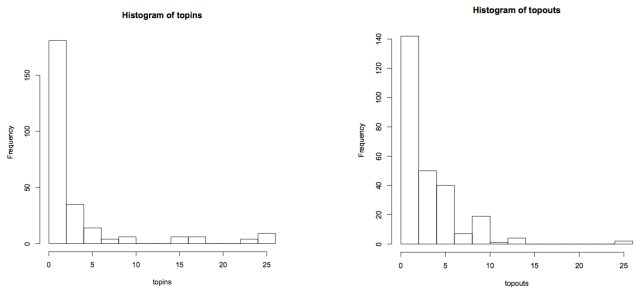


Figure 4: Example of in-degree and out-degree node distribution for the MAPK network from 2000.

Figure 4 shows an example of the in- and out-degree for the 2000 network instance; note the power-law distribution of these degrees. From the total number of proteins, only one or two display both high in- and out-degree. One explanation could be that certain proteins require a lot of factors to become active, while their neighboring enzymes become active under the influence of only one protein, but are capable to catalyze many others in return. This can be seen as a bottleneck effect.

As the size of the network increases, the clustering coefficient also increases, a property needed for the scale-free networks and small-world networks. While the clustering coefficient for the directed network has a constant value throughout time, the others fluctuate greatly. The undirected network displays high values for the first clustering coefficients, only to drop below 1 around the year 2000. Similarly to the undirected clustering coefficient, the values of the randomized coefficient for the earlier instances is much higher than for the newer instances. These results demonstrate that with the completeness of the MAPK network due to new biological discoveries and technical developments, the clustering coefficient for the directed network reaches higher values than both the undirected and randomized coefficient, property needed for the scale-free and small-world networks authentication. To augment these findings, Figure 5 describes the evolution of the clustering coefficient in all three type of networks.

Table 5: Results for the size of the giant connected component in all the networks studied as a ratio of the total number of nodes in the network (rounded to two decimal points).

Year	1980	1984	1988	1992	1996	2000	2010
Directed	0.5	0.46	0.35	0.92	0.90	0.95	0.96
Undirected	0.5	0.46	0.35	0.92	0.90	0.95	0.96
Randomized	0.8	0.38	0.57	0.92	0.94	0.89	0.94

Table 2: Statistical measurements applied on the MAPK directed networks. Mean degree is not measured as in a directed network we have in- and out-degrees. We also do not measure the clique size again due to the direction of the edges.

Year	1980	1984	1988	1992	1996	2000	2010
Density	0.1777	0.1410	0.0584	0.0356	0.0287	0.0220	0.0250
Diameter	2	3	3	8	12	11	15
Average Path	0.1333	0.1538	0.1151	0.3519	0.6955	2.8743	3.4534
Clustering Coefficient	0.0833	0.0512	0.0480	0.0735	0.0686	0.0856	0.1383
Degree Exponent	0.1667	0.8958	1.6326	2.9495	3.9546	3.4365	2.7610

Table 3: Statistical measurements applied on the MAPK undirected networks. These networks are build by considering all links in the directed network as undirected.

Year	1980	1984	1988	1992	1996	2000	2010
Mean Degree	1.4	1.5384	1.3846	2.3611	2.6470	3.7811	6.6113
Density	0.1555	0.1282	0.0553	0.0332	0.0262	0.0220	0.0250
Diameter	3	3	4	11	9	17	23
Average Path	0.4444	0.5769	0.3907	4.0207	3.6843	2.5436	0.6455
Clustering Coefficient	0.4166	0.2	0.1388	0.2168	0.1889	0.1051	0.0302
Max Clique	3	3	3	3	3	3	3
Number of Cliques	1	1	1	10	17	13	48
Degree Exponent	1.3247	1.4222	2.2589	2.2894	4.02081	4.6554	4.2303

In networks one also look at the existence of a giant component, which is the, the largest subgraph of connected proteins. In scale-free networks the emergence of giant components happen in a phase transition as we increase the number of edges. In general it is expected that the giant component would spam the entire network after this phase transition. Meaning that the network transits from a disconnected network into a highly connected one very rapidly. We have included the size of the giant component for each of the MAPK instances. From the results presented in Table 5, we can infer that the giant component encompasses almost the entire network for all three types of networks. This fact suggests that most elements are interacting with each other, justifying the shortest path length results.

Another observation related to the directed MAPK network can be made regarding the average path length between nodes. Unlike the undirected and randomized average path lengths that recorded large values from the earliest reconstructions, the directed network reported values between 0.11 and 0.35 until 1992. However, as the MAPK directed network developed, the average path length reached 3.45. This suggests that the directed network displays small-world properties having short average path length. For the calculations of the shortest path lengths between all pairs of nodes, the *RBGL* package was utilized. Although the results were not included in the paper, they prove the existence of shortest path lengths for all directed networks, starting with the oldest distances until reaching the hypergraph formation.

Moreover, nodes that occur on many short path between other nodes have high betweenness centrality and represent the hubs. To evaluate these hubs, we computed the betweenness centrality using the *RBGL* function, which is scaled by a factor of $(n-1)(n-2)/2$, where n represents the number of nodes in the network.

Table 6 displays the first three hubs in each type of network, for every instance starting with 1980 until 2010. Each column represents a time instance, which is partitioned in three parts, one for every type of network: directed, undirected and randomized. In each column subdivision, the most significant proteins were recorded in decreasing order based on the RBCC coefficient calculated using Bioconductor. Starting with the hypergraph from the right, $MAPK1$, $GRB2$, $MAP2K2$ are the common hubs for both directed and undirected networks, suggesting that these proteins have numerous interactions with their neighboring proteins. On the other hand, the randomized network shows a totally different organization, having no proteins in common with the other networks.

For the year 2000, $MAP2K2$ and $MAP2K1$ are still among the most important nodes, but $MEKK1$ has become the central protein in both directed and undirected network. The randomized network maintains its distinctive structure, differing not only from the other two values form the other networks, but also from the 2010 randomized hypergraph, without any overlapping hubs.

The same results as in 2000 can be drawn when analyzing the 1996 instance from Table 6. The directed and the undirected networks have similar hub proteins, however these hubs differ from the 2000 and 2010 reconstructions. This can be explained by the fact that in 1996, not all proteins had been discovered and added to the MAPK pathway. In this case, Ras is the most significant hub protein, but it still fails to appear in the randomized network, separating the randomized instance again from the other results.

In 1992, Ras was still the major contributor to the connectivity of the MAPK instance for both the directed and undirected networks, along with MOS,and JNK. However, none of these proteins appear as the hubs of the randomized network. For all the last three instances, $GRB2$ and JNK represent the common factors in both directed and undirected networks. It can be argued that since we are dealing with small networks, the hubs found in the older instances remain hubs for the later ones as well. This correlates to molecular biology, since it is true that both $GRB2$ and JNK play crucial roles in the dynamics of the MAPK network.

Table 4: Statistical measurements applied on the MAPK randomized network. These networks are generated using a randomization algorithm.

Year	1980	1984	1988	1992	1996	2000	2010
Mean Degree	1.6	1.6923	1.4615	2.5277	2.9019	2.7096	4.2719
Density	0.1777	0.1410	0.0584	0.0356	0.0287	0.0220	0.2867
Diameter	3	3	3	6	8	9	6
Average Path	0.4766	0.3452	0.3977	2.5347	3.4332	5.4456	2.6357
Clustering Coefficient	0.3	0.4285	0.2865	0.0533	0.0373	0.0548	0.0118
Max Clique	3	3	2	3	3	3	3
Number of Cliques	1	2	19	2	4	2	2
Degree Exponent	1.2317	1.2536	1.5348	1.5644	1.7563	1.8696	1.8010

Table 6: The hubs in the MAPK network based on the RBCC values. Each column represents a time instance starting with 1980, which is partitioned in three parts, one for every type of network: directed, undirected and randomized. In each column subdivision, the three most significant proteins were recorded in decreasing order, based on the RBCC coefficient calculated using Bioconductor.

Type of Network	1980	1984	1988	1992	1996	2000	2010
Directed	JNK	GRB2	MOS	MOS	Ras	MEKK1	MAPK1
	0.0833	0.0757	0.0797	0.3614	0.2568	0.4782	0.2685
	P53	JNK	GRB2	Ras	JNK	RRAS2	GRB2
	0.0833	0.0606	0.0652	0.3498	0.2400	0.2754	0.2467
	GRB2	P38	JNK	ERK	MOS	MAP2K2	MAP2K2
	0.0277	0.0606	0.0326	0.3173	0.2258	0.2357	0.2366
Undirected	JNK	GRB2	MOS	MOS	Ras	MEKK1	MAPK1
	0.0833	0.0757	0.0797	0.3614	0.2568	0.4782	0.2685
	P53	JNK	GRB2	Ras	JNK	RRAS2	GRB2
	0.0833	0.0606	0.0652	0.3498	0.2400	0.2754	0.2467
	GRB2	P38	JNK	ERK	MOS	MAP2K2	MAP2K2
	0.0277	0.0606	0.0326	0.3335	0.2258	0.2357	0.2366
Randomized	JNK	GRB2	GRB2	GADD45	Rap1	AKT	NFAT4
	0.4444	0.0530	0.1561	0.1500	0.1174	0.1208	0.1374
	PDGF	NGF	AKT	PTP	TAK1	ASK1	RasGFR
	0.1666	0.0303	0.1304	0.1440	0.1157	0.1141	0.1009
	Tau	MAPKAPK	RafB	Tau	MEKK23	ECSIT	PDGFR
	0.1666	0.0303	0.1245	0.1364	0.1018	0.1030	0.0992

4. CONCLUSION

Network theory represents an important and popular approach for the analysis of large-scale interaction networks [6]. The analysis of networks provides useful insight into the structure and properties of real networks, such as biological networks and social networks. However, the results of these assessments need to be considered thoughtfully, since they are generally based on the assumptions that they are error-free and complete. As stated in Section 1, biological data is usually sampled from a larger pool which does not necessarily guarantees a significance. More than that, various software yield different results in computing the same attributes. For example, the clustering coefficient obtained using the Bioconductor packages varies from the clustering coefficient computed by Gephi. Therefore consistency is essential when recording data.

In this study, we performed seven reconstructions of the directed MAPK signaling pathway over a period of 20 years, and hypothesized that some of these instances will display scale-free and small-world properties. With the help of R and Bioconductor, we were able to construct the instances of the network that were analyzed independently and compared to a similar undirected network, as well as a randomized network with the same number of nodes and edges as the directed one. Although the directed network revealed small clustering coefficients, between 0.04 and 0.05 in the beginning, the value of the later coefficients increased to 0.13, making the directed clustering coefficient larger than both the undirected and the randomized ones as the MAPK network reached completeness. We can observe the same manifestation for the power-law exponents of the MAPK directed network. Even though the values of the first exponents were all below 1, as MAPK evolved into the network known today, the power-law exponent stabilized itself between 2 and 3 in conformity with the scale-free requirements. For the average path length between nodes, its values appeared to be very small, between 0.11 and 0.69, for most instances and increased to 3.45 only in 2010. Relative to the undirected and the randomized path lengths, the result for 2010 directed network might appear large, but we must take into consideration the fact that it is the only network having directionality. Moreover, when comparing the average path length of the directed MAPK network to

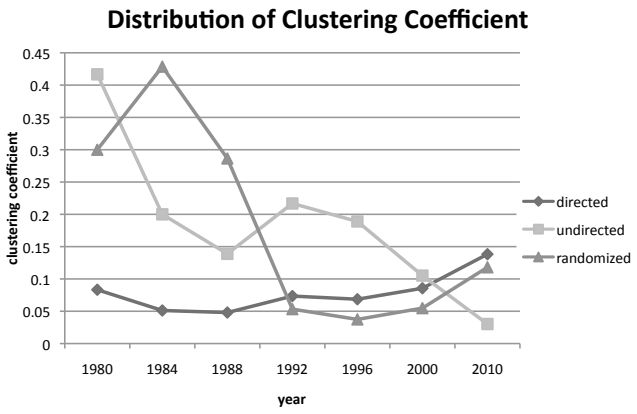


Figure 5: Time evolution of the clustering coefficient for the three types of networks, directed, undirected and randomized. While the clustering coefficient for the directed network has a constant value throughout time, the others fluctuate greatly. The undirected network displays high values for the first clustering coefficients, only to drop to values below 1 around the year 2000. Similarly to the undirected clustering coefficient, the values of the randomized coefficient for the earlier instances is much higher than for the newer instances. These results demonstrate that as MAPK becomes more complete due to new biological discoveries, the clustering coefficient for the directed network reaches higher values than both the undirected and randomized coefficient. This may indicate that the network is reaching a completion stage where it is more stable.

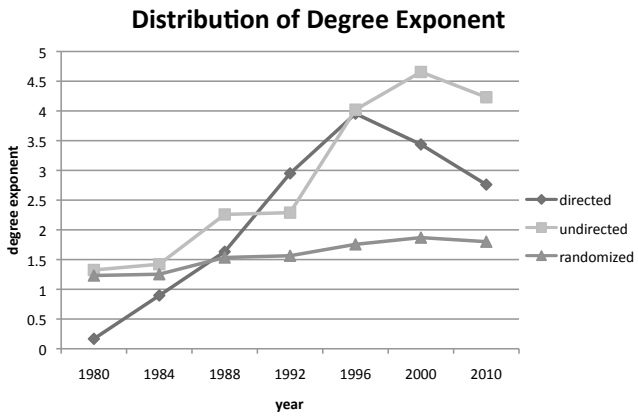


Figure 6: Degree distribution λ plotted against time for the directed, undirected and randomized network. Although the λ value for directed network oscillates in the beginning, it stabilizes itself between 2 and 3, which is in conformity with the scale-free λ values. The undirected network grows steadily throughout time from 1.32 to 4.23, a value outside the range for scale-free networks. In contrast, the value of the randomized λ is almost static, growing only from 1.23 to 1.80.

the average path length for the film actors network, 3.48 [2], and the Internet network, 3.31 [9], the results yield very similar values, suggesting that the path length for our network qualifies as a short path length and satisfies the condition for the small-world claim. Based on these results, the dynamics of protein networks establishes itself as a necessary factor when analyzing biological systems. We predicted that the older instances of the MAPK signaling pathway might not have been scale free, due to the fact that many of the proteins involved in the process had yet to be added to the network, and our hypothesis came true since some of the first instances (1984 and 1988) exhibit small clustering coefficients, as well as small degree exponent. However, with the discovery of the majority of MAPK network components, we were able to demonstrate that the more recent reconstructions display scale-free and small-world characteristics, as postulated. One interesting fact to observe is that proteins having high degree-in are not the same as the proteins having high degree-out; thus there is no correlation between in-degree and out-degree. A possible explanation is that in the MAPK cascade, certain proteins require the interaction of many other neighboring proteins, but they are responsible for activating only a few downstream proteins. These latter enzymes can play an important role in activating many other proteins, displaying a high out-degree, but a low in-degree.

On the other hand, the undirected network has one instance when it displays scale-free properties (1988). During this time, the power-law exponent is within 2 and 3, the average path length is small, having a value of 0.39, while the clustering coefficient is 0.13. With the addition of new proteins, the undirected network loses its scale-free structure to a more random one. For the randomized network, the power-law exponent is always below 2, reinforcing the idea that it is no coincidence that the directed MAPK network is scale-free and small-world.

Although other studies [14] have attempted to decipher the human MAPK pathway characteristics, they have only analyzed the latest instance of MAPK network. Even if their research comes to the conclusion that MAPK is indeed a scale-free network, the results obtained from this study vary from past experiments, even in the input data. One study contains a MAPK network with 148 nodes and 187 edges, while the MAPK retrieved from the KEGG database past updated on April 4, 2010, had 124 nodes and 168 edges. The discrepancy could come from the fact that their data was downloaded from another protein database. Therefore, as long as there is no standardization in the data, scientists will struggle with incompatible datasets, as well as sampling concerns.

4.1 Future Work

A better approach than having an undirected randomized equivalent is to examine the directed MAPK network relative to a directed randomized version. This way, directionality is preserved for both networks. To create such randomization, we can apply an edge-swapping algorithm to the directed MAPK network. This iterative algorithm would exchange interactions in a random manner: if an interaction exists between nodes v_i and v_j, as well as between v_m and v_n, then the edge from v_i would go to v_n and the edge from v_m would go to v_j, preserving the degree distribution between the nodes [13]. By generating such randomized networks we would improve the qualitative analysis on the global property statistics of the MAPK network.

One aspect that has been ignored in this paper is the presence of network motifs: small subgraphs that occur at significantly higher rates than expected by chance. Other studies [18] analyze the presence of 3- and 4-node motifs to determine when in time these motifs form; they suggest that directed triangular loops, feed-forward 3-node loops, bi-fan and two-path robust motifs are the most recurring 3-node and 4-node motifs [21]. Once found, these motifs can provide information about their function, since previous studies suggest that motifs found in transcriptional regulatory networks and neural networks are involved in processing information [21,

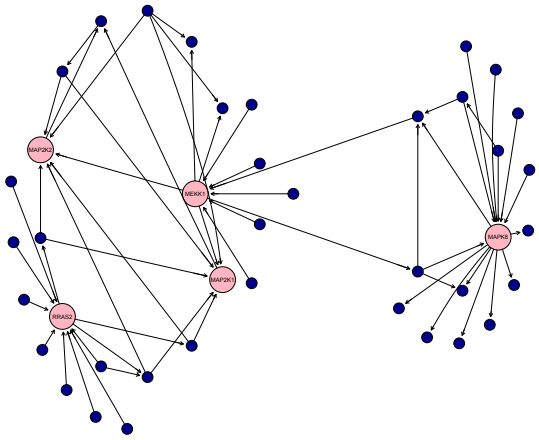


Figure 7: Example of nodes with the highest relative betweenness centrality in the MAPK network from 2000.

18]. We can then test to see if the implied motifs display signs of evolutionary conservation, in other words if they are represented by the oldest functional proteins discovered in most of these instances.

Another aspect we want to pursue is to compare the MAPK network to other important regulatory networks from the KEGG database. Other studies have already analyzed the MAPK pathway between various databases and found that the information encoded there differs significantly[13]. There should be no surprise that interaction networks found in various databases may contain more false positives and less untested relationships around the hubs than around less popular genes, since they are updated by scientists that assume accuracy of the data and results. Therefore, using the same database might yield concurrent data, in hope of finding similar properties among existing networks. Once the characteristics of these networks are investigated, we could then cluster them and study whether they perform similar functions. This analysis can be of great usage, especially in the medical and pharmaceutical fields. If certain proteins are targeted in known networks, then looking to test new drugs on other networks from the same cluster may become more straightforward.

Lastly, several other studies which analyze the dynamics of the network, can be performed using the framework presented here. The evolution of communities overtime and the nature of special features of these communities could be used to reveal unknown properties of the MAPK network. One thing to be kept in mind is that there is no agreed-upon community algorithm in Network Sciences. Recently, Fortunato [12] performed an extensive study on community algorithms that can help us identify which algorithm is more suitable for the case of MAPK.

5. REFERENCES

[1] R. Albert and A.-L. Barabási. Statistical mechanics of complex networks. *Rev. Mod. Phys.*, 74(1):47, Jan. 2002.

[2] L. A. Amaral, A. Scala, M. Barthelemy, and H. E. Stanley. Classes of small-world networks. *Proceedings of the National Academy of Sciences of the United States of America*, 97(21):11149–11152, October 2000.

[3] J. Avruch. Map kinase pathways: The first twenty years. *Biochimica et Biophysica Acta (BBA) - Molecular Cell Research*, 1773(8):1150 – 1160, 2007.

[4] S. Bandyopadhyay, R. Sharan, and T. Ideker. Systematic identification of functional orthologs based on protein network comparison. *Genome Research*, 16(3):428–435, 2006.

[5] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, Oct. 1999.

[6] A.-L. Barabási and Z. Oltvai. Network biology: understanding the cell's functional organization. *Nature Reviews Genetics*, 5(2):101–113, 2004.

[7] M. Bastian, S. Heymann, and M. Jacomy. Gephi: An Open Source Software for Exploring and Manipulating Networks. In *International AAAI Conference on Weblogs and Social Media*, pages 361–362, 2009.

[8] V. Carey, R. Gentleman, W. Huber, and J. Gentry. Bioconductor software for graphs. *Bioinformatics and Computational Biology Solutions Using R and Bioconductor*, pages 347–368, 2005.

[9] Q. Chen, H. Chang, R. Govindan, and S. Jamin. The origin of power laws in internet topologies revisited. In *IEEE INFOCOM*, volume 2, pages 608–617, 2002.

[10] A. Clauset and C. Moore. Accuracy and scaling phenomena in internet mapping. *Phys. Rev. Lett.*, 94(1):018701, Jan. 2005.

[11] P. Erdös and A. Rényi. On random graphs. *Publicationes Mathematicae*, 6:290–297, Dec. 1959.

[12] S. Fortunato. Community detection in graphs. *Physics Reports*, 486:75âĂŞ174, 2010.

[13] M. Futschik, A. Tschaut, G. Chaurasia, and H. Herzel. Graph-theoretical comparison reveals structural divergence of human protein interaction networks. *Genome Informatics*, 18:141–151, 2007.

[14] N. S. Gergana Bounova, Michael Hanowsky. MAPK signaling pathway analysis. Technical Report ESD.342 Final Project Report, MIT, 2006.

[15] W. Huber, V. Carey, L. Long, S. Falcon, and R. Gentleman. Graphs in molecular biology. *BMC Bioinformatics*, 8(Suppl 6):S8, 2007.

[16] M. Kanehisa, S. Goto, M. Furumichi, M. Tanabe, and M. Hirakawa. KEGG for representation and analysis of molecular networks involving diseases and drugs. *Nucleic Acids Research*, 38(Suppl 1):355–360, 2010.

[17] O. Mason and M. Verwoerd. Graph theory and networks in biology. *IET Syst. Biol.*, 1(2), Mar. 2007.

[18] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network Motifs: Simple Building Blocks of Complex Networks. *Science*, 298(5594):824–827, 2002.

[19] M. Newman. The structure and function of complex networks. *SIAM Review*, 45:167–256, 2003.

[20] M. Stumpf, C. Wiuf, and R. May. Subnets of scale-free networks are not scale-free: Sampling properties of networks. *PNAS*, 102(12):4221–4224, Jan. 2005.

[21] C. von Mering, R. Krause, B. Snel, M. Cornell, S. Oliver, S. Fields, and P. Bork. Comparative assessment of large-scale data sets of protein-protein interactions. *Nature*, 417(6887):399–403, May 2002.

[22] D. J. Watts and S. H. Strogatz. Collective dynamics of small world networks. *Nature*, 393:440–442, June 1998.

[23] J. D. Zhang and S. Wiemann. KEGGgraph: a graph approach to KEGG PATHWAY in R and Bioconductor. *Bioinformatics*, 25(11):1470–1471, 2009.

SVM-based classification and feature selection methods for the analysis of Inflammatory Bowel disease microbiome data

Nuttachat Wisittipanit
Department of Bioinformatics and Computational Biology, George Mason University, Manassas, VA 20110, USA
nwisitti@gmu.edu

Huzefa Rangwala
Department of Computer Science, George Mason University, Fairfax, VA 22030, USA
rangwala@cs.gmu.edu

Masoumeh Sikaroodi
Department of Environmental Science and Policy, George Mason University, Fairfax, VA 22030, USA
msikaroo@gmu.edu

Ali Keshavarzian
Department of Medicine, Section of Gastroenterology, Rush University Medical Center, Chicago, IL 60612, USA
Ali_Keshavarzian@rush.edu

Patrick Gillevet
Department of Environmental Science and Policy, George Mason University, Fairfax, VA 22030, USA
pgilleve@gmu.edu

Ece A. Mutlu
Department of Medicine, Section of Gastroenterology, Rush University Medical Center, Chicago, IL 60612, USA
Ece_Mutlu@rush.edu

ABSTRACT

Motivation: The human gut is one of the most densely populated microbial communities in the world. The interaction of microbes with human host cells is responsible for several disease conditions and of criticality to human health. It is imperative to understand the relationships between these microbial communities within the human gut and their roles in disease.

Methods: In this study we analyze the microbial communities within the human gut and their role in inflammatory bowel disease (IBD). The bacterial communities were interrogated using Length Heterogeneity (LH-PCR) fingerprinting of mucosal and luminal associated microbial communities during healthy and diseases states. We develop support vector machine based classification and feature selection techniques to differentiate between healthy controls and patients suffering from IBD. Moreover, we develop site-specific classifiers to analyze community differences on the inner lining of the intestine (called mucosa) and the fluid within the intestine (called lumen).We also determine differentially abundant features across the different samples.

Results: Using SVM-based classifiers with feature selection, we can distinguish the communities between the healthy controls and disease class patients. We also report differentially abundant features that exist between the different patient groups. The site-specific analysis provides an understanding of the microbial community differences between the lumen and mucosa of the healthy controls and patients suffering from IBD.

Categories and Subject Descriptors

1.5.2 [**Pattern Recognition**]: Design Methodology – *Classifier design and evaluation.*

Conference'10, July 25-28,2010,Washington, DC, USA

General Terms

Methodology

Keywords

Inflammatory Bowel Disease Classification, Microbiome, Microbial Abundance Profile, Feature selection, Support Vector Machine, Metastats

1. INTRODUCTION

The human gut is one of the most densely populated microbial communities known [1]. It is a nutrient-rich environment packed with up to 100 trillion microbes [2] which is ten times more than the total number of human body cells [3] and it is estimated that there are somewhere between 500 to 1000 different species living in our gut [4].

These microbes have a collective genome called the microbiome, which contains at least 100 times as many genes as a human genome [5]. These microbiomes encode many metabolic functions including the ability to extract energy and nutrients from our diet [6]. It has been hypothesized that these interactions between digestive tract epithelium with the microbiome are critical to human health. These interactions are involved with the immune system and its responses, metabolic regulation, and digestion [7]. In an abnormal condition such as the disease state, these interactions may be altered (dysbiosis) resulting in disrupted functionality.

In this paper, we present a computational pipeline to analyze the gut microbiome and correlate its composition to inflammatory bowel disease (IBD).

We interrogate the relative abundances of microbial components of the gut microbiome using Length Heterogeneity Polymerase Chain Reaction (LH-PCR). The LH-PCR method uses the first two variable regions of the 16S Ribosomal RNA (rRNA) genes for the identification of different microbial species [8]. 16S rRNA genes are marker genes that are highly conserved and provide an accurate identification of the bacteria family. We derive features from the LH-PCR data and use a supervised learning approach within the support vector machine (SVM) framework [9] to classify samples between the healthy controls and disease states. We identified a small group of significant features (amplicon lengths representing microbial content) to distinguish disease and healthy states at various intestinal sites. We also detected differentially

abundant features between the disease and healthy controls using a computational tool called Metastats [10].

The results demonstrate that we can distinguish the IBD samples from the healthy control samples. We were able to identify several differentially abundant features at the intestinal location where inflammation occurs in the disease state (IBD). Moreover, in the comparison of the communities between the mucosa (boundary of intestine) and the lumen (inner), more features were differentially abundant in the healthy control state than in the diseased state. This potentially suggests a role played by the bacteria across the intestine boundaries and alteration of communities due to IBD.

1.1 Inflammatory Bowel Disease

Inflammatory Bowel Disease (IBD) is a group of disorders that cause inflammation in the intestines. The inflammation may last a long time and is recurring or chronic in nature. Of all the disorders belonging to IBD, Crohn's and Ulcerative Colitis are the two most severe. Crohn's disease usually involved the inflammation of the lower part of small intestine called ileum. Ulcerative Colitis usually involves the top layer of the lining of the large intestine or colon.

Despite extensive research of IBD, the cause(s) of the disease remains unknown to this day. One factor that might be involved in the etiology (origination) of IBD is the microbial gut flora [11].

The mucosa is the boundary or the inner lining of the intestine and the lumen is the fluid inside the intestine. The microbial content and abundance within the mucosa and lumen can vary across the different intestine locations and may be altered during the disease state [25].

2. METHODS

In this study, we investigated the microbiome in disease and healthy control samples to determine if we can predict the disease class using features derived from LH-PCR. Two major types of analysis were performed on the datasets, the first one was the SVM classification and the second one was to use Metastats to find differentially abundant features between the communities. We also placed emphasis on understanding the differences between the mucosa and lumen across different locations of the intestine for patients suffering from Crohn's and Ulcerative Colitis versus healthy controls.

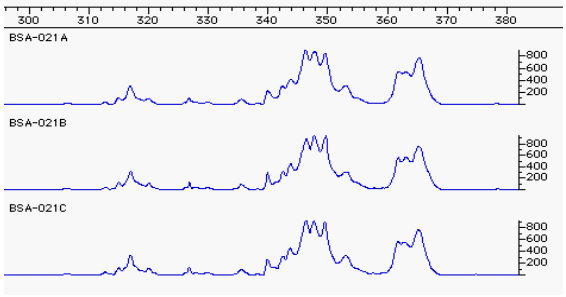


Figure 1. Examples of three LH-PCR electropherograms. The fragment or amplicon lengths (horizontal axis) and relative fluorescence intensity (vertical axis) of three samples are shown. Each peak of fluorescence intensity is proportional to the abundance of the amplicons associated with any given length or OTU.

2.1 Patient Sample Collection

The samples used in this study were collected at Rush University Medical Center along with clinical and demographic information (done by AK and EM). These samples were collected from the mucosa layer at the following intestinal sites: (i) ileum, (ii) colon consisting of ascending and transverse colon, (iii) sigmoid and the lumen fluid from IBD and healthy patients. The IBD suffering patients were further categorized into two diagnostic classes: (i) Crohn's and (ii) Ulcerative Colitis. The mucosal samples were taken using the pinch biopsy procedure whereas those from the lumen were from a fluid inside the intestine that was collected using a Luken trap.

2.2 LH-PCR Fingerprinting

The patient samples were used to generate an amplicon length heterogeneity profile [13]. LH-PCR uses the 16S Ribosomal RNA (rRNA) genes for the identification of different microbial species by using primers to amplify highly conserved regions that are interspersed with hyper-variable sequence regions. The highly conserved primers amplify a wide range of species [14] and have been shown to be useful for community analysis [8].

In this technique, total genomic DNA is extracted from a community of microbes and part of the 16S rRNA gene is amplified by PCR using a fluorescently labeled 27F primer (AGAGTTTGATCCTGGCTCAG) and an unlabelled 355R primer (GCTGCCTCCCGTAGGAGT) that target the first two hyper-variable regions of 16S rRNA genes. These labeled amplicons are separated by gel electrophoresis and detected by laser-induced fluorescence with an automated gene sequencer [15] yielding a profile of amplicon lengths or Operational Taxanomic Units (OTUs) associated with the various microorganisms in the sample where the height (intensity) of the peak is proportional to the abundance of the amplicons associated with any given length [13]. Thus, the LH-PCR method profiles a community based on the patterns of lengths of amplified products (amplicons) providing a rapid and cost-effective way to distinguish taxa in the communities although the OTUs do not actually identify individual species or genera [16]. The amplicon length distribution, in this study, was computed and filtered such that any OTU that has less than 1% abundance is removed from the analysis. Figure 1 shows examples of three LH-PCR electropherograms and their reproducibility. The features generated were the abundance of the different OTUs or different peak values at different lengths. The total number of unique features or peaks was 103 determined after a binning procedure that would allow shifted peaks to be aligned together.

2.3 SVM and kernel functions

Support vector machine (SVM) is a supervised learning framework and can build binary classifiers to distinguish the IBD-suffering patients from the healthy patients. SVMs have the ability to classify samples after being trained with a collection of known, labeled feature vectors (in this study the microbial abundance profiles is referred to as the feature vectors and the labels are diseases or controls).

Given a set of training diseased-state samples S^+ and a set of healthy control samples S^-, using large margin principles. SVM learns a classification function $f(X)$ of the form [9]:

$$f(X) = \sum_{X_i \in S^+} \lambda_i^+ K(X, X_i) - \sum_{X_i \in S^-} \lambda_i^- K(X, X_i), \quad (1)$$

where λ_i^+ and λ_i^- are non-negative weights that are computed during training by maximizing a quadratic objective function, and $K(.,.)$ is called the kernel function that is computed over the various training set and test set instances. A penalty parameter, C, is introduced during the learning phase which is part of the error term in the SVM and represents the rate at which the SVM 'learns' from the misclassifications.

Given Equation 1, a new sample X is predicted to be diseased or healthy depending on whether $f(X)$ is positive or negative, respectively. In addition, the value of $f(X)$ can be used to obtain a meaningful ranking of a set of instances, as it represents the strength by which they are members of the positive or negative class [9, 17].

Only one kernel functions was employed in this study, a radial basis function (RBF).
The RBF is defined by

$$K(X,Y) = \exp\left(-\gamma ||X - Y||^2\right), \gamma > 0. \quad (3)$$

For the RBF kernel function, there are 2 parameters; C *and* γ to be searched to find the optimum classifier. We also tested the performance of the linear kernel and noticed that the RBF kernel consistently outperformed the linear kernel. As such, we report results only for the RBF kernel.

2.4 Feature selection: Relief Algorithm

We performed feature selection to identify the relevant OTU features that have the ability to separate the disease sample from the healthy sample in order to improve our classifiers.

The algorithm we used was the Relief Algorithm [18]. The basic idea under Relief is that values of different class instances on a feature should be different and values of same class instances should be the same. Features satisfying this criterion are ranked higher in comparison to features failing to satisfy.

As shown in Algorithm 1, there are 2 parameters which are (1) k: the number of neighbors for each sample (2) m: The number of reference samples. Given p = the number of all attributes. W= weight vector with the number of dimensions equal to p, in each loop of m times loops, Relief randomly picks a sample X and finds nearest-hit (within the same class of X) and nearest-miss (within the different class of X) samples by using the p-dimensional Euclidian distance for selecting Near-hit and Near-miss and then update W.

2.5 Metastats Analysis

Metastats [10] is a computational tool used to detect differentially abundant features present in the disease samples compared to those in the healthy control samples.

Algorithm 1. Relief [18]

1: Relief(S, k, m):
2: S = total samples
3: k = number of neighbors for each sample
4: m = number of reference samples
5: p = number of total features
6: W = (0, 0, .. 0) with number of elements equal to p
7: Separate S into S^+ = {positive instances} and
8: S^- = {negative instances}
9: For i = 1 to m:
10: Pick at random an instance $X \in S$
11: Pick k positive instances closet to X, $P_u \in S^+$,
12: $u = 1$ *to* k
13: Pick k negative instances closet to X, $P_t \in S^-$
14: $t = 1$ *to* k
15: If X is a positive instance:
16: then $NH_u = P_u$, $NM_t = P_t$
17: else $NH_u = P_t$, $NM_t = P_u$
18: UpdateWeight(W, X, NH_u, NM_t)
19: UpdateWeight(W, X, NH_u, NM_t):
20: For i = 1 to p:
21: $W_i = W_i - \sum_{u=1}^{k}(X_i - NH_{ui})^2 + \sum_{t=1}^{k}(X_i - NM_{ti})^2$

Where X_i is value of feature i of X, NH_{ui} is the value of i of Near-Hit vector u and NM_{ti} is the value of i of Near-Miss vector t

It employs a false discovery rate to improve specificity in high-complexity environments and separately handles sparsely-sampled features using Fisher's exact test. For this study, it was used to identify OTUs whose presence or absence correlated to the disease states and also across the different intestinal locations.

The Fisher's exact test is an appropriate method for sparse datasets because it models the sampling process according to a hyper geometric distribution (sampling without replacement) [10]. Equation 4 shows the calculation of the exact p-value assuming that the null hypothesis is True (i.e. no differential abundance).

$$p\text{-value} = \frac{\binom{R_1}{f_{11}}\binom{R_2}{f_{21}}}{\binom{n}{C_1}} \text{ where,}$$

$$R_1 = f_{11} + f_{12}, R_2 = f_{21} + f_{22}, C_1 = f_{11} + f_{21},$$

$$n = f_{11} + f_{12} + f_{21} + f_{22}. \quad (4)$$

From equation 4, for each feature i, f_{11} is the number of observations of feature i in all individuals from the disease class, f_{21} is the number of observations that are not feature i in all individuals from the disease class, f_{12} and f_{22} are similarly defined for control. The Fisher's Exact Test is more suitable for the sparse datasets found in this study in contrast to a Student's t-test.

3. MATERIALS

3.1 Datasets

Using the LH-PCR procedure, feature matrices were generated for the different intestinal sites of patients suffering from Crohn's disease Ulcerative Colitis and

Table 1. Number of samples and features in all the datasets used in this study. These include Ileum, Colon (Ascending + Transverse Colon), Lumen and Entire sites (all samples).

Disease Class/Location	Ileum	Colon	Sigmoid	Lumen	Entire
Crohn's	74	108	101	35	318
Ulcerative Colitis	44	63	76	29	212
Healthy Control	63	67	107	53	290
Total	181	238	284	117	820

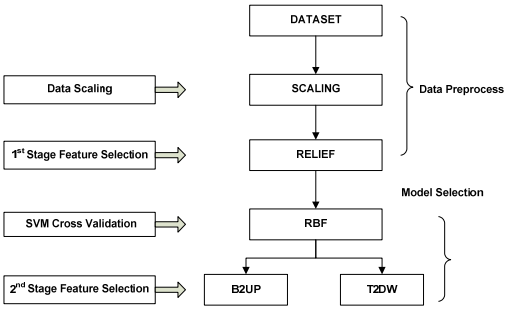


Figure 2. The process diagram for the SVM-based classification of IBD for a dataset having features selected from the 1st stage feature selection (Relief algorithm) and 2nd stage feature selection (one-by-one feature selection). All datasets begins with a raw dataset in 'Data Preprocess' step. Next the raw dataset is passed to Data Scaling (SCALING), then to the 1st Stage Feature Selection by the Relief algorithm (RELIEF). Then the dataset is sent to 'Model Selection' step. The 2nd stage feature selection (one-by-one feature selection method) is performed on the dataset with two approaches; bottom-up (B2UP) and top-down (T2DW) using SVM cross validation with RBF kernel function to select the features having the highest classification accuracy. After all the tests are done for one raw dataset, there will be two unique configurations and evaluation results. The configuration having the highest accuracy will be selected to show the results.

healthy control. The samples were taken from the mucosal layer of the ileum, colon, and sigmoid and within the lumen of the intestine. The total number of samples available for the different groups and locations are reported in Table 1. "Entire" refers to the collection of microbial samples collected from all the sites. All the samples have the same number of features of 103 OTUs.

3.2 Experimental Methodology

Figure 2 provides an overview of the different steps. The first step is denoted 'Data Preprocess' where the features are normalized (SCALING). The first stage feature selection is performed using the RELIEF algorithm (described in Section 2.4). The second step involves "Model Selection" where the different parameters are selected for the SVM classification along with a second-stage feature selection using either a bottom-up or top –down approach.

To evaluate the performance of the different feature selection techniques we report the classification accuracy for three cases: (i) using all the 103 features denoted as FS_FULL, (ii) first stage RELIEF-based feature selection denoted as FS_RELIEF and (iii) combination of RELIEF and SVM-based feature selection denoted as FS_BOTH

To detect differentially abundant features by Metastats, the experiments were performed on all the 103 OTU features.

3.2.1 Data Scaling

The intensity values of each feature (OTU) were scaled to have values in the range -1 to 1 by making the lowest value of each feature equal to -1 and the maximum equal to 1 and the other values were normalized within the range. The main advantage of data scaling is to avoid features in greater numeric ranges dominating those in smaller numeric ranges. Another advantage is to avoid numerical difficulties encountered during the calculation [19].

3.2.2 Model Selection

We used the RBF kernel for the SVM classification models. Selection of two parameters C and γ was done by a grid-search. The range of the grid-search for log base 2 of C was [-5, 15] with step increase of 2 and for log base 2 of γ was [-15, 3] with also the step increase of 2. For each of the parameter pair (C, γ), five-fold cross validation was performed. In the five fold cross validation, one-fifths of the data is held out for testing and the remained four-fifths is used for training. This is iterated five times and the evaluation metrics are averaged across the five iterations [20, 21]. After the grid search was done, the parameter pair (C, γ) at which the cross-validation had the highest accuracy was used to build the final classifier.

3.2.3 Feature Selection

In the first stage of feature selection, we used the Relief algorithm to select significant features from each dataset. We set parameters k: the number of neighbors for each instance equal to 30% of the total number samples and m: the number of reference examples equal to the number of all samples. The features of the dataset were ranked from the highest score to the lowest ones by the method. Then we selected 30 features within top-ranked scores.

For the second stage feature selection (FS_BOTH), we experimented with two approaches, a bottom-up approach and top-down approach which are a one-by-one feature selection methods going in opposite directions. Both used the features selected from the first stage feature selection. The method for the bottom-up approach could be briefly described as the following: the starting number of features is one and tested with SVM cross-validation process to determine which feature produces the best accuracy, stores that feature and, and removes it from the feature pool. In the next iteration, the process uses the previous best feature as the base and appends one feature from the pool and tests to find out which feature (appending to the previous base) produces the best accuracy, stores that in the feature set and removes that appending feature from the pool. The top-down approach is exactly opposite to the bottom up approach where features are removed one-by-one from the feature set. For both the approaches, the iteration was continued till the accuracy converged.

The SVM-based parameter selection and feature selection was performed using the ORANGE machine learning toolkit available at http://www.ailab.si/orange/

To determine the differentially abundant features we used the computational tool Metastats available at http://metastats.cbcb.umd.edu/software.html. We used all the 103 OTU features and analyzed the features with p-value less than or equal to 0.08 and 0.05 for the different combination of healthy and disease datasets across the mucosa and the lumen.

Table 2. Evaluation results of the three feature types for Crohn's of Entire, Ileum, Colon, Sigmoid and Lumen sites. Notes: POS/NEG = Number of positive/negative classes, NFS = Number of Features Selected, 1st NFS = Number of Features Selected in first stage, 2nd NFS = Number of Features Selected in second stage.

		Entire	Ileum	Colon	Sigmoid	Lumen
Class Type	Positive Negative	Crohn's Healthy Control	Crohn's Healthy Control	Crohn's Healthy Control	Crohn's Healthy Control	Crohn's Healthy Control
	POS/NEG	318/290	74/63	108/67	101/107	35/53
FS_FULL (All features)	Full features	103	103	103	103	103
	Accuracy (%)	50.17	56.88	45.14	52.42	35.29
	AUC	0.5	0.5	0.5	0.5	0.5
	Sensitivity	0.50	0.61	0.47	0.56	0.34
	Specificity	0.50	0.52	0.42	0.49	0.36
	F-Measure	0.51	0.60	0.52	0.54	0.30
FS_RELIEF (Relief-based Feature Selection)	NFS	30	30	30	30	30
	Accuracy (%)	72.85	70.82	48.00	50.46	49.87
	AUC	0.77	0.75	0.5	0.5	0.5
	Sensitivity	0.77	0.82	0.51	0.51	0.49
	Specificity	0.69	0.57	0.43	0.50	0.51
	F-Measure	0.75	0.75	0.55	0.50	0.44
FS_BOTH (Relief and One-by-one Feature Selection)	1st NFS	30	30	30	30	30
	2nd NFS	29	27	8	8	5
	Accuracy (%)	74.01	75.22	74.86	71.16	70.39
	AUC	0.77	0.77	0.78	0.74	0.64
	Sensitivity	0.78	0.86	0.88	0.67	0.37
	Specificity	0.70	0.62	0.54	0.75	0.92
	F-Measure	0.76	0.79	0.81	0.69	0.50

4. RESULTS

The patient samples were organized into two major groups for the classification and feature selection analysis. The first group was used for the comparison between the IBD (Crohn's or Ulcerative Colitis) and healthy control samples in the same sites. The second group was used for the comparison of the lumen and the mucosa samples from the ileum, colon and sigmoid. The experimental purpose for the first group was to see how good the methods were in distinguishing the disease and healthy control samples at the same site and which features were differentially abundant. Moreover, we wanted to test which site showed the best classification accuracy for each disease and compare the features that are differentially abundant.

The purpose for the second group was to test our hypothesis that in the homeostasis state (normal) there exist some bacteria in the mucosa that are different from those in the lumen. However, in the diseased state we expect dysbiosis (imbalance in bacterial distribution) where those bacteria in the lumen are also likely to be found in the

Table 3. Evaluation results of the three feature types for Ulcerative Colitis of Entire, Ileum, Colon, Sigmoid and Lumen sites. Notes: POS/NEG = Number of positive/negative classes, NFS = Number of Features Selected, 1st NFS = Number of Features Selected in first stage, 2nd NFS = Number of Features Selected in second stage.

		Entire	Ileum	Colon	Sigmoid	Lumen
Class Type	Positive Negative	Ulcerative Colitis Healthy Control	Ulcerative Colitis Healthy Control	Ulcerative Colitis Healthy Control	Ulcerative Colitis Healthy Control	Ulcerative Colitis Healthy Control
	POS/NEG	212/290	44/63	63/67	76/107	29/53
FS_FULL (All features)	Full features	103	103	103	103	103
	Accuracy (%)	50.79	53.16	55.38	47.03	40.51
	AUC	0.5	0.5	0.5	0.5	0.5
	Sensitivity	0.50	0.52	0.59	0.47	0.48
	Specificity	0.51	0.54	0.52	0.47	0.36
	F-Measure	0.46	0.48	0.56	0.43	0.36
FS_RELIEF (Relief-based Feature Selection)	NFS	30	30	30	30	30
	Accuracy (%)	72.92	52.25	48.46	64.52	41.18
	AUC	0.78	0.5	0.5	0.68	0.5
	Sensitivity	0.59	0.55	0.52	0.37	0.41
	Specificity	0.83	0.51	0.45	0.84	0.42
	F-Measure	0.65	0.48	0.50	0.46	0.33
FS_BOTH (Relief and One-by-one Feature Selection)	1st NFS	30	30	30	30	30
	2nd NFS	14	10	7	7	4
	Accuracy (%)	74.51	80.4	81.54	73.18	79.34
	AUC	0.75	0.77	0.83	0.65	0.74
	Sensitivity	0.54	0.55	0.78	0.46	0.55
	Specificity	0.90	0.98	0.85	0.93	0.92
	F-Measure	0.64	0.70	0.80	0.59	0.65

Table 4. Evaluation results (Accuracy, AUC and F-Measure) of the disease and healthy control classes between the mucosa and Lumen sites.

	Crohn's			Ulcerative Colitis			Healthy Control		
	Accuracy (%)	AUC	F-Measure	Accuracy (%)	AUC	F-Measure	Accuracy (%)	AUC	F-Measure
Ileum vs Lumen	77.27	0.72	0.85	74.93	0.6	0.81	74.18	0.7	0.77
Colon vs Lumen	79.09	0.63	0.25	73.39	0.52	0.24	73.53	0.65	0.51
Sigmoid vs Lumen	77.22	0.61	0.21	78.1	0.62	0.34	69.87	0.59	0.17
Average	77.86	0.65	0.44	75.47	0.58	0.46	72.53	0.65	0.48

mucosa. Therefore, the similarity of the microbial compositions between the mucosa and the lumen is likely to be higher in the disease state. In summary, we would like to see for which groups (healthy control or disease) the methods distinguish the samples better between the mucosa and the lumen. We also identify which features were differentially abundant between the mucosa and lumen in the different patient groups.

4.1 Disease-specific classifiers

Tables 2 and 3 report the results of the classification between Crohn's versus healthy controls and Ulcerative Colitis versus healthy control for all the different sites (colon, sigmoid and ileum) as well as the combination of all sites (ENTIRE). The tables show the classification and evaluation results done with the three feature types: (i) FS_FULL, (ii) FS_RELIEF and (iii) FS_BOTH. We report the number of samples within the pair of classes denoted by POS/NEG, the number of features from first stage feature selection (default at 30), the number of features from the second stage feature selection (2nd SFS). We report the accuracy of the SVM cross validation. We also present the area under receiver operating characteristic curve (AUC) [22, 23] to indicate how good the classifier is in distinguishing between each disease and the healthy control. AUC tells us the rate of true positive versus the false positive rate and helps in assessment of classifiers which deal with imbalanced distribution of class labels. We also show sensitivity (the proportion of actual diseased patients which are correctly identified), specificity (the proportion of healthy patients which are correctly identified) and F-measure (the harmonic mean of precision and recall) [24].

Table 2 shows that the classification accuracies for the FS_FULL are relatively low with an average accuracy of 47.98% compared to the FS_RELIEF with an average accuracy of 58.4% (a 21.72% increase from FS_FULL). However, the best results were from the FS_BOTH having an average accuracy of 73.13% (a 25.22% increase from FS_RELIEF). Of all the results for all sites except the Entire combination, the Ileum site has the highest accuracy in the FS_FULL, FS_RELIEF and FS_BOTH at 56.88%, 70.82% and 75.22%, respectively. This corresponds to the fact that Crohn's disease mostly occurs at the Ileum and the SVM classification distinguishes the disease from healthy control samples using features derived from this site.

Table 3 (Ulcerative Colitis) exhibits a similar trend as shown in Table 2 where there is much improvement of the average accuracies. The average accuracies for the FS_FULL, FS_RELIEF and FS_BOTH are 49.37%, 55.87% (a 13.17% increase from FS_FULL) and 77.79% (a 39.23% increase from FS_RELIEF), respectively. The colon site has the highest accuracy in the FS_FULL and FS_BOTH at 55.38% and 81.54% respectively. It is interesting to see that the test for colon site has the highest F-Measure value of 0.80. The observation made from Table 3, this also corresponds to the fact that Ulcerative Colitis mostly occurs at the colon.

4.2 Site-specific classifiers

Table 4 reports the evaluation metrics (Accuracy, AUC and F-Measure) of the SVM cross-validation tests with the FS_BOTH between the mucosa (ileum, colon and sigmoid) and the lumen. The tests were done between all the sample classes (Crohn's, Ulcerative Colitis and healthy control) to detect the mucosa from the lumen. Table 4 shows that the average accuracy comparing the mucosa and the lumen of healthy control group is slightly lower than those of the disease groups. From the results, it is not clear whether the classification between the mucosa and lumen of the healthy control samples is better that those of the disease samples or not.

4.3 Metastats-based Analysis

Tables 5 and 6 display the Metastats results of Crohn's and Ulcerative Colitis (versus healthy control) showing the features having p-value less than or equal to 0.08 using the Fisher's Exact Test. Features highlighted with * indicate a p-value of less than or equal to 0.05. Table 7 displays the Metastats results between the mucosa (ileum, colon and sigmoid) and the lumen showing only features having p-value equal or lower than 0.05 along with the site at which the feature is dominantly abundant.

Table 5 shows that for Crohn's disease there are three differentiating features detected in the ileum, whereas only

Table 5. Metastats analysis results showing features with p-value less than or equal to 0.08 of Crohn's disease versus healthy control for Entire, Ileum, Colon, Sigmoid and Lumen sites. Features having p-value less than or equal to 0.05 are marked with *. Bold features are the mutual features found in the 2nd NFS of FS_BOTH in Table 2.

Entire Crohn's		Ileum Crohn's		Colon Crohn's		Sigmoid Crohn's		Lumen Crohn's	
Features	p-value	Features	p-value	Features	p-value	Features	p-value	Features	p-value
None	None	358_31*	0.0016	359_45*	0.0246	None	None	337_36*	0.0481
		359_45*	**0.0019**	347_67	0.073				
		329_84*	0.0267						
		332_79	0.0641						
		346_79	0.0701						

Table 6. Metastats analysis results showing features with *p*-value less than or equal to 0.08 of Ulcerative Colitis disease for Entire, Ileum, Colon, Sigmoid and Lumen sites. Note that *p*-value less than or equal to 0.05 are marked with *. Bold features are the mutual features found in the 2nd NFS of FS_BOTH in Table 3.

Entire Ulcerative Colitis		Ileum Ulcerative Colitis		Colon Ulcerative Colitis		Sigmoid Ulcerative Colitis		Lumen Ulcerative Colitis	
Features	*p*-value	Features	*p*-value	Features	*p*-value	Features	*p*-value	Features	*p*-value
None	None	**353_26***	**0.0286**	**347_67***	**0.0116**	None	None	**335_21***	**0.0233**
		332_79	0.0587	**346_79***	**0.0166**			**359_45***	**0.0368**
		366_87	0.0698	**359_45***	**0.0231**				
		362_95	0.0698	**332_79***	**0.0336**				
		372_23	0.0698	**349_09***	**0.037**				
				351_17*	0.0414				

one at the colon and lumen. The results correspond to the fact that Crohn's disease mostly occurs at the Ileum. The results are similar with those from Table 2.

Table 6 shows that for Ulcerative Colitis, there are five features detected that are differentially abundant at the colon and only one at both ileum and lumen. The results correspond to the fact that Ulcerative Colitis disease mostly occurs at the colon. The results are similar with those from Table 3.

Table 7 shows that the Metastats results between the communities at the ileum mucosa and lumen of Crohn's and Ulcerative Colitis have no features detected to be differentially abundant while that of healthy control has three features namely 358_31, 341_03 and 348_37 with two of them being dominantly abundant at the lumen site. For the comparison between the colon mucosa and lumen, there was one differentiating feature between that of healthy Control (348_37 dominantly abundant at the lumen) while there are none for Crohn's and Ulcerative Colitis. Comparing the mucosa at Sigmoid and lumen of Healthy Control, there were four features detected to be differentially abundant namely 341_03, 348_37, 354_52 and 359_45 with two of them being dominantly abundant at the lumen site while there is one (335_21) for Ulcerative Colitis and none for Crohn's. This means the communities between the mucosa and the lumen are more similar in the disease state than in the healthy.

4.4 Mutual Features

Features that are common by the FS_BOTH procedure and Metastats analysis by Fisher's exact test at the same sample site are called mutual features. Those features were selected by SVM procedure to have the highest classification accuracy for each disease and also were detected by Fisher's exact test to be differentially abundant. At the ileum site, there is one mutual feature for Crohn's and one for Ulcerative Colitis. At the colon site, there is none for Crohn's and five for Ulcerative Colitis. Within the lumen there is none for Crohn's and two for Ulcerative colitis. However, there is not any mutual feature at the Sigmoid site. Table 5 and 6 shows these mutual features in bold.

5. CONCLUSION AND FUTURE WORK

In this study we developed a computational pipeline to characterize the microbial communities in the gut identified LH-PCR fingerprinting technique. Specifically, we trained SVM-based classification models to distinguish samples obtained from patients suffering from IBD i.e., Crohn's or Ulcerative Colitis versus the healthy controls. The samples were obtained from the mucosa or inner linings of different intestine locations as well as the lumen fluid well within the intestine.

Using feature selection approaches coupled with classification we were able to classify IBD samples from healthy controls. We demonstrated the improvement in accuracy when using the RELIEF based feature selection technique. Moreover, the results show that for Crohn's disease, the ileum site has the highest accuracy across the sites. These results correspond to the fact that tissue inflammation in Crohn's disease mostly occurs at the ileum of small intestine. Similarly for the Ulcerative colitis known to have the highest impact at colonic sites, we observe that using samples from the colon location can distinguish the disease from healthy controls. We showed the difference between the microbial communities present at the mucosa and those found in the lumen fluid.

Using the Metastats tool for comparing the healthy control samples at the mucosa and lumen versus the disease state showed that there was an alteration in the bacterial community during the disease state. In fact, during the disease condition the bacteria in the lumen would also be likely found in the mucosa. From the results, we can conclude that there exist significant OTUs or features that

Table 7. Metastats analysis results showing features with *p*-value less than or equal to 0.05 of the mucosa and Lumen sites. In the format of A | B | C, A = feature name, B = *p*-value, C = Site at which the feature is dominantly abundant.

	Crohn's	Ulcerative Colitis	Healthy Control
Ileum vs Lumen	None	None	341_03 | 0.0043 | Lumen
			348_37 | 0.0305 | Lumen
			358_31 | 0.0014 | Ileum
Colon vs Lumen	None	None	348_37 | 0.0034 | Lumen
Sigmoid vs Lumen	None	335_21 | 0.0336 | Lumen	341_03 | 0.0136 | Lumen
			348_37 | 0.0077 | Lumen
			354_52 | 0.0385 | Sigmoid
			359_45 | 0.0094 | Sigmoid

are differentially abundant between Crohn's, Ulcerative Colitis and Healthy control state at the inflamed disease sites and we can distinguish the disease from the control samples well at those location. These features are potential biomarkers for disease state.

In the future we would like to perform a similar analysis using the 16S sequence data as well as the available metagenomic data which would provide accurate identification and abundance measure of the microbial communities. This study is an example of understanding biological complexity and impact on human health studied using computational approaches.

AUTHOR'S CONTRIBUTIONS

AK and EM are responsible for the patient sample and clinical data collection. AK and EM are responsible for the clinical hypothesis and were supported by NIH 5R21DK071838-02. PG and MS designed and performed the microbiome analysis. NW and HR developed the computational pipeline and were supported by NSF IIS-0905117. HR and PG provided inputs regarding the computational experimental studies. NW, HR and PG analyzed the primary data. The manuscript was written by NW, HR and PG but was read and approved by all the authors.

REFERENCES

[1] Getting to know the tiny majority. *Nature Reviews Microbiology*, 6, 170, Mar 2008.

[2] R.E. Ley, D.A. Peterson, and J.I. Gordon. Ecological and Evolutionary Forces Shaping Microbial Diversity in the Human Intestine. *Cell Reviews*, 124(4):837-48, Feb 2006.

[3] P.J. Turnbaugh, R.E. Ley, M. Hamady, C.M. Fraser-Liggett, R. Knight, and J.I. Gordon. The Human Microbiome Project. *Nature*, 449:804–810, 2007.

[4] C.L. Sears. A dynamic partnership: Celebrating our gut flora. *Anaerobe*, 11:247–51, 2005.

[5] S. R. Gill. Metagenomic Analysis of the Human Distal Gut Microbiome. *Science*, 312:1355, 2005.

[6] H. Tilg, A.R. Moschen, and A. Kaser. Obesity and the Microbiota. *Gastroenterology*, 136(5):1476-83, 2009.

[7] S.E. Cheesman, and K. Guillemin. We know you are in there: Conversing with the indigenous gut microbiota. *Research in Microbiology*, 158(1):2-9, 2009.

[8] D. K. Mills, K. Fitzgerald, C. D. Litchfield, and P. M. Gillevet. A comparison of DNA profiling techniques for monitoring nutrient impact on microbial community composition during bioremediation of petroleum-contaminated soils. *Journal of Microbiological Methods*, 54:57-74, 2003.

[9] B.E. Boser, I.M. Guyon, and V.N. Vapnik. A training algorithm for optimal margin classifiers. *Proceedings of the fifth annual workshop on Computational learning theory*, 144-152, 1992.

[10] J.R. White, N. Nagarajan, and M. Pop. Statistical Methods for Detecting Differentially Abundant Features in Clinical Metagenomic Samples. *PLoS Comput Biol*, 5(4): e1000352, Doi:10.1371/journal.pcbi.1000352, April 2009.

[11] P. Chandran, S. Satthaporn, A. Robins, O. Eremin. Inflammatory bowel disease: dysfunction of GALT and gut bacterial flora (II). *The Surgeon*, 1:125-136, 2003.

[12] W. Strober. Unraveling Gut Inflammation. *Science*, 313:1052-1054, 2006.

[13] S. Komanduri, P. M. Gillevet, M. Sikaroodi, E. Mutlu, and A. Keshavarzian. Dysbiosis in pouchitis: evidence of unique microfloral patterns in pouch inflammation. *Clinical gastroenterology and hepatology*, 5:352-60, 2007.

[14] F. Schluenzen, A. Tocilj, R. Zarivach, J. Harms, M. Gluehmann, D. Janell, A. Bashan, H. Bartels, I. Agmon, F. Franceschi, and A. Yonath. Structure of functionally activated small ribosomal subunit at 3.3 angstroms resolution. *Cell*, 102(5):615-23, Sep 2000.

[15] N. Ritchie, M. Schutter, R. Dick, and David D. Myrold. Use of Length Heterogeneity PCR and Fatty Acid Methyl Ester Profiles To Characterize Microbial Communities in Soil. *American Society for Microbiology*, 66(4):1668-1675, April 2000.

[16] A.E. Bernhard, D. Colbert, J. McManus, K.G. Field. Microbial community dynamics based on 16S rRNA gene profiles in a Pacific Northwest estuary and its tributaries. *FEMS Microbiol. Ecol.*, 52:115– 128, 2005.

[17] H. Rangwala, and G. Karypis. Profile-based direct kernels for remote homology detection and fold recognition. *Structural bioinformatics*, 21:4239-47, 2005.

[18] K. Kira and L. Rendell. A practical approach to feature selection. *In D. Sleeman and P. Edwards, editors*, Proc. 9th Int'l Conf. on Machine Learning, pages 249, 1992.

[19] C.C. Chang, and C.J. Lin. LIBSVM: a Library for Support Vector Machines. *http://www.csie.ntu.edu.tw/~cjlin/libsvm*, 2001.

[20] B. Efron, and R. Tibshirani. An Introduction to the Bootstrap. *Chapman and Hall*, New York, 1993.

[21] C. Yang, D. Mills, K. Mathee, Y. Wang, K. Jayachandran, M. Sikaroodi, P. Gillevet, J. Entry, and G.Narasimhan. An ecoinformatics tool for microbial community studies: Supervised classification of Amplicon Length Heterogeneity (ALH) profiles of 16S rRNA. *Journal of Microbiological Methods*, 65(1):49-62, April 2006.

[22] T. Fawcett. An introduction to ROC analysis. *Pattern Recognition Letters*, 27:861-74, 2006.

[23] J.A. Swets. Signal Detection Theory and ROC Analysis in Psychology and Diagnostics, Collected Papers. *Lawrence Erlbaum Associates*, Mahwah, NJ, 1996.

[24] D.G. Altman, and J.M. Bland. Diagnostic tests. 1: Sensitivity and specificity. *BMJ*, 308(6943), 1994.

[25] H.M. Probert, and G.R. Gibson. Bacterial biofilms in the human gastrointestinal tract. *Curr Issues Intest Microbiol*, 3(2):23-7, 2002.

Analysis of Network Topological Features for Identifying Potential Drug Targets

Jintao Zhang
Center for Bioinformatics
University of Kansas
Lawrence, KS 66045
jtzhang@ku.edu

Jun Huan*
Department of Electrical Engineering and Computer Science
University of Kansas
Lawrence, KS 66045
jhuan@ittc.ku.edu

ABSTRACT

Identifying potential drug targets is a crucial task for drug discovery. Traditional *in silico* approaches utilize only protein sequence or structural information to predict whether a protein can be a drug target, and achieve limited success. Since proteins function in the context of interaction networks by interacting with other cellular macromolecules, analysis of topological features of proteins in such networks can reveal important insights on whether a protein can be a potential drug target. In this paper, we introduced ten new topological features extracted from human protein interaction networks. When designing these new features, we specially emphasized the roles of three disease-related groups of proteins: known drug targets, disease genes, and essential genes. Based on the topological feature set, we built supervised learning models using support vector machines, L1-regularized logistic regression, and k-nearest neighbors to predict whether testing proteins can be drug targets or not. We also analyzed the relevance of each feature to the probability of proteins being drug targets. We achieved up to 80% classification accuracy using tenfold cross validation, and yielded very stable results with a large number of random samplings. Our method can also be used to prioritize multiple candidate proteins according to their probability of being drug targets.

Categories and Subject Descriptors

H.2.8 [**Database Management**]: Database Applications-Data Mining

General Terms

Algorithm, Experimentation

*To whom correspondence should be addressed.

BIOKDD '10, July 25, 2010, Washington D.C., USA

Keywords

Human Protein Interactome, Network Feature, Machine Learning, Drug Target

1. INTRODUCTION

In the past decade, the success rate of drug discovery decreased while the corresponding R&D investment increased significantly [1, 17]. A significant number of drug failures were attributed to the utilization of inappropriate drug targets at the early preclinical stages [6]. Although great efforts have been exerted on drug research and development, a limited number of drug targets have been identified. In addition, the majority of drug targets came from a few gene families, e.g. about 60% current drug targets are membrane proteins [25], while in human genome only 15∼39% genes were predicted to contain transmembrane segments [2]. To this end, it will be a critical to predict whether a protein can be a potential drug target using *in silico* methods.

Genes in human genome have been classified into two classes: the "druggable" genome (genes that express proteins binding to drug-like molecules with potency greater than a threshold [12, 14], e.g. 10 μM) and the remaining "undruggable" genome. According to the estimation of Hopkins *et al.* [14], there are approximately 10% human genes that can potentially become drug targets. However, the boundary between the "druggable" and "undruggable" genome is ambiguous and dynamic, and highly depends on the screening libraries. Therefore, identifying novel drug targets, especially from the currently considered "undruggable" genome, could be a feasible solution to the current dilemma in drug discovery.

It is well known that proteins rarely function in isolation inside and outside cells, but rather behave as part of highly interconnected cellular networks [8, 21, 22]. It will be advantageous to investigate proteins in the context of human protein-protein interaction (PPI) networks, with the hypothesis that topological environment of drug targets should be distinct from that of non-drug-target proteins. For instance, Yildirim *et al.* [25] constructed a drug-target network and found that most of known drug targets formed a giant cluster in the human PPI network. They concluded that drug targets were usually not essential genes, but they are close to essential genes and disease genes in the network. Therefore, known drug targets, disease genes, and essential genes are special groups of proteins in the human PPI network, just like shining stars in the dark sky. It is the

motivation for us to propose network topological features of proteins regarding to their relationships to these special proteins.

Machine learning algorithms have been widely used in pharmaceutical and bioinformatics studies. By integrating novel topological features with advanced pattern recognition technologies, we may build highly accurate models to predict if a protein is likely to be a potential drug targets or. By analyzing the relevance of proposed features, we can prioritize a set of proteins according to their probability of being drug targets, thus design high-throughput screening to verify them more efficiently. The rest of this paper is organized as follows: In section 2 we will introduce related work that have been done on identifying potential drug targets using computational methods. In section 3 we discuss how we collect all data, how we proposed novel topological features, and the machine learning algorithms will use. We then present our experimental results and conduct reasonable interpretations in section 4. Finally we will draw some concluding remarks in section 5.

2. RELATED WORK

Various computational methods have been proposed and applied for identifying potential drug targets. Zheng *et al.* [26] reported properties of new druggable proteins based on analysis of approved drug targets, including membership of a target family, involvement in no more than two pathways, presence in no more than two tissues, and etc. In addition, Hajduk *et al.* [12] used the 3D structural information to predict whether a particular protein can bind with small, drug-like compounds. Although these methods achieved reasonable performance, they suffer from either poor generalization capability or limited availability of data such as protein 3D structures.

Supervised learning has also been widely used for drug target identification. For instance, Li *et al.* [18] predicted potential drug targets based on simple sequence properties such as hydrophobicity, polarity, and etc., and achieved about 80% accuracy using SVM cross validation on the 186 selected drug targets. In addition, Bakheet *et al.* [3] proposed a comprehensive list of properties of human drug target proteins, including EC numbers, Gene Ontology terms, Glycosylation, and etc., analyzed their correlation to drug targets, and also used them as features to predict potential drug targets. Recently, Zhu *et al.* [27] used five topological features extracted from human PPI networks to identify potential drug targets, and proposed a measure to rank proteins in the PPI network according to their potential of being drug targets.

If a protein is modulated by external stimulus, it is highly likely that its interacting partners, even the whole module, are also subject to the perturbation. Given a sufficiently complete network of high-quality PPI annotations, drug targets can be distinguished from non-drug-target proteins due to their distinct response to network perturbations, which is our basic assumption to use solely network topological features to predict potential drug targets. By investigating a set of 15 network features related to known drug targets, disease genes, and essential genes, we formalized the prediction of potential drug targets as a typical supervised learning problem. To this end, we applied sophisticated machine learning algorithms to analyze these network features and build accurate models to predict potential drug targets.

3. METHODS

3.1 Human Protein Interaction Network

To extract network topological features for proteins, we need a human protein-protein interaction (PPI) network with as accurate and complete as possible interaction annotations. UniHI (http://www.unihi.org) is a unified human PPI network containing over 250,000 human PPIs collected from 14 major PPI sources with careful data integration and literature curation [8, 9]. It also provides quality scoring systems for each data source. After careful curation, we obtained a human PPI network with 13,602 proteins as nodes and 157,349 PPIs as edges by removing redundant nodes and/or edges, merging duplicated nodes and/or edges, and excluding non-human proteins and other noises.

3.1.1 Approved Drug Targets

We obtained the gene symbols of the target proteins of all approved drugs from the DrugBank database [23, 24], and use official gene symbols to cross-reference proteins in the UniHI network and in DrugBank. We mapped the genes symbols of drug targets in DrugBank to the curated UniHI network, and excluded any drug targets that have some network topological features unavailable, resulting in a set of 1,092 drug targets for positive training samples and network feature calculation.

3.1.2 Human Disease Genes

We downloaded all "Genes Associated with Diseases" from the GeneCards database (http://www.genecards.org/), and used gene symbol mapping to identify 1,521 proteins as disease genes in UniHI. In addition, we removed any disease genes that have been approved drugs, and obtained a final set of 1,157 disease genes. They will be used to calculate important topological features for proteins in the UniHI network.

3.1.3 Human Essential Genes

A human gene was defined as "essential" if a knockout of its mouse ortholog confers lethality. To find human essential genes, we first extracted mouse essential genes from the Mouse Genome Informatics Database [11], and obtained 2,564 human essential genes through the human-mouse ortholog associations. Using gene symbol mapping we obtained 2,059 essential genes in the UniHI network, and finally used 1,759 of them after removing those that either have been used as drug targets or disease genes, or have some topological features unavailable.

3.1.4 Putative Non-drug-target Proteins

It is indispensable to have negative samples to build an accurate model, that is, we need some proteins that can be surely determined as not drug targets. This is technically difficult since no researcher is interested in validating that a protein is definitely not a drug target. To solve this dilemma, we simply excluded any proteins that have been used as drug targets, disease genes, and essential genes from consideration [3, 18, 27]. In addition, any proteins that have some topological features unavailable were also removed, resulting in a set of 9,674 proteins. In all experiments, we randomly selected a number of proteins from this set as our negative samples. It is sure that there will exist some false negatives, however, with random sampling the error rate is

Table 1: The 15 topological features extracted from the human PPI network.

Feature	Formula	Description
Degree	k_i	Number of direct links to node i
Clustering coefficient	$2n_i/k_i(k_i+1)$	n_i is the number of links among the k_i neighbors of node i
Topological coefficient	$\sum_j J(i,j)/k_i$	See text
Minimal SPL	$min_j(d_{ij})$	Minimal SPL to drug targets, disease, and essential genes
Mean SPL	$\sum_j d_{ij}/\|P\|$	Average SPL to drug targets, disease, and essential genes
Fraction of neighbors	k_i^p/k_i	Fraction of neighbors of node i as drug targets, disease, and essential genes
Characteristic distance	See text	Measure of Clustering of drug targets, disease, and essential genes

acceptable (<5%) considering that only less than 10% random proteins could be drug targets [14].

3.2 Network Topological Features

A network is an undirected acyclic graph consisting of a number of nodes and edges. A node can represent any object, and an edge connects two nodes and usually carries some physical meaning such as interaction, similarity, and etc. In this work, we proposed 15 topological features extracted from human PPI networks, including three general topological features: degree, clustering coefficient, and topological coefficient.

The "degree" (DEG) of a node is the number of edges connecting it to other nodes. The "clustering coefficient" [4] of a node is defined as $\boldsymbol{C}_i = 2\boldsymbol{n}/(\boldsymbol{k}_i{*}(\boldsymbol{k}_i\text{-}1))$, where n denotes the number of direct neighbors of a given node i, and $\boldsymbol{k}_i$ is the number of links among the n neighbors of node i. If the clustering coefficient (CLU) of a node equals 1.0, the node is at the center of a fully connected cluster called a clique. If the clustering coefficient is close to 0, the node is in a loosely connected region. We can calculate average clustering coefficient over nodes with the same degree, and then obtain the distribution of average clustering coefficient over node degrees. The average of $\boldsymbol{C}_i$ over all nodes of a network assesses network modularity. Finally, the "topological coefficient" (TPG) [22] $\boldsymbol{T}_i$ of a node i with $\boldsymbol{k}_i$ neighbors is computed as $\boldsymbol{T}_i = \sum_m \boldsymbol{J}(i,m)/\boldsymbol{k}_i$, where $\boldsymbol{J}$(i,m) is defined for all nodes m that share at least one neighbor with node i. The value $\boldsymbol{J}$(i,m) is the number of neighbors shared between the nodes i and m, plus one if there is a direct link between i and m. The topological coefficient is a relative measure for the extent to which a node shares neighbors with other nodes. Nodes that have one or no neighbors are assigned a topological coefficient of 0. Topological coefficients can be used to estimate the tendency of the nodes in the network to have shared neighbors.

3.2.1 Shortest Path Length-related Features

Next we defined six network features that are related to shortest path lengths (SPLs) between proteins and three pharmaceutically important sets of proteins/genes: approved drug targets, human disease genes, and essential genes. The shortest path length (SPL) between two nodes in a network is defined as the minimal number of consecutive edges between them. Given a protein in the UniHI network, the first three features are computed as its minimal SPLs to the nearest drug target (SPdt), disease gene (SPdg), and essential gene (SPeg), not including the protein itself. These features evaluate the minimal distance between a protein and pharmaceutically important proteins. In addition, the remaining three features are the mean SPLs between a protein and all drug targets (avSPdt), disease genes (avSPdg), and essential genes (avSPeg) in the UniHI network. These features evaluate the overall average distance between a protein and those important special proteins.

3.2.2 Characteristic Distance Features

The final six features are related to the clustering between proteins and the three pharmaceutically interesting sets of proteins. If we use a visual graph to view how the proteins distribute in the UniHI network, we can find the aggregation between proteins and drug targets (disease genes, and essential gens) are different from one to another. So the first three features will be defined as the fraction of approved drug targets (FRdt), disease genes (FRdg), and essential gens (FReg) in the direct neighbors of a given protein. For instance, if a protein has 15 direct neighbors (its degree = 15), 3 of them are known drug targets, 2 of them are disease genes, and 0 of them are essential genes, these three features will be calculated as 0.2 (3/15), 0.133(2/15), and 0.0 (0/15), respectively. These features provide a simple measure how the pharmaceutically important proteins are clustering around a protein.

To gain more understanding on the probability of proteins being potential drug targets, we developed a model to quantify the clustering of drug targets, disease genes, and essential genes surrounding other proteins. By computing the fraction $\boldsymbol{F}_i$ of drug targets, disease genes, and essential genes at each distance $\boldsymbol{d}_i$, we obtained the distribution of these three sets of proteins around each other protein in the network. We then defined the characteristic distance $\boldsymbol{D}_c^p$ as follows: $1/(\boldsymbol{D}_c^p)^2 = \sum_{i=1}^{n} \boldsymbol{F}_i/\boldsymbol{d}_i^2$, where n is the diameter of the network and p represents a protein from the three sets. The mechanism underlying this formula was from electrostatics and the Coulomb law. We can view each drug target (disease gene, and essential gene) as a "unit charge" that generated an electric field with field strength $\boldsymbol{F}_i/\boldsymbol{d}_i^2$, and all these electric fields accumulated at the position of a given protein. Therefore, the last three features are computed as the characteristic distance between a protein and all approved drug targets (CHdt), disease genes (CHdg), and essential genes (CHeg), respectively.

3.3 Machine Learning

Three types of classification algorithms were used in this work. Support vector machines [5] were applied with cross validation on the data sets to build models and select the best one. Logistic regression [15] was used to find the relative significance of each proposed feature. In addition, k-nearest neighbors method [10] was adopted to find the most similar drug target for each positively predicted protein.

Table 2: Summary of baseline results from experiments using 11 features.

Algorithm	Model Parameters	Training Accuracy	Testing Accuracy
SVMs	C=4.539+/-1.15 γ=2.406+/-0.804	0.727+/-0.017	0.691+/-0.030
L1-regularized Logistic Regression	λ = 0.0006+/-0.0011	0.750+/-0.014	0.757+/-0.028
k-Nearest Neighbors	k = 3	/	0.694+/-0.017

Table 3: Z-scores of L1-regularized logistics regression model coefficients

Network Feature	DEG	CLU	TPG	SPdg	SPeg	avSPdg	avSPeg	FRdg	FReg	CHdg	CHeg
Z-score	1.111	-3.282	-0.272	-3.018	-8.007	-3.090	3.450	-4.871	-1.028	-3.854	2.552

3.3.1 Support Vector Machines

Support Vector Machines (SVMs) is a widely used supervised learning algorithm with excellent performance on many applications in data mining, machine learning, bioinformatics, image recognition, and etc. Burges *et al.* [5] provided a comprehensive tutorial on SVMs. The basic ideas behind SVMs for binary classification are to find the best decision boundary between the positive and the negative data points by maximizing the margin between two parallel hyperplanes in the feature space, one for positive samples and the other for negative samples. These two parallel hyperplanes can rotate and translate in the opposite directions to maximize the margin between the two sets of data points, separating the feature space into three regions: the positive region, margin, and the negative region. When new testing samples are added to this feature space, they are classified based on their distance to the two hyperplanes. In this paper, we used SVMs to build highly accurate model for the baseline of comparison using tenfold cross validation and bootstrapping. The LIBSVM [7] implementation and RBF kernels were used in all experiments.

3.3.2 Logistic Regression

Logistic regression is a generalized linear model for pattern recognition. Such models include a linear exponential part followed by a "link function". First the linear function of input features is calculated and run through the logistic link function. The optimal coefficients of the linear function are learned from training data. Comparing with linear regression, logistic regression can be used to construct a model which estimates probabilities, e.g. for medical diagnosis and credit scoring. With proper regularization, the coefficients of a logistic regression model can be used to evaluate the relative significance of each feature. In this paper, we used a L1-regularized logistic regression (LLR) algorithm proposed by Boyd *et al.* [16] to run our experiments.

3.3.3 k-Nearest Neighbors

The k-nearest neighbors (kNN) algorithm is a simple classification method based on the votes of the k nearest neighbors of a given data point in the feature space. Given a training data set with class labels, the parawise distance (or similarity) between a testing data point and each training sample will be calculated and sorted, and the top k closet (or most similar) training samples are picked, and the majority of the k class labels will be assigned to the given testing data point. This method can be used to identify the most similar known drug target to a protein that was predicted as a "potential drug target" when k = 1.

4. RESULTS AND DISCUSSION

Based on the descriptions of the 15 network features proposed for each protein, they were computed for 1,092 known drug targets extracted from DrugBank, and 9,674 putative non-drug-target proteins in the UniHI database. The 1,092 known drug targets played two roles in the experiments: a) Four network features (SPdt, avSPdt, FRdt, and CHdt) were calculated using these drug targets; and b) they also served as training positive samples in cross validation.

4.1 Using No Drug-target-related Features

Due to the trick usage of known drug targets in feature extraction and cross validation, we first excluded these four drug-target-related features from consideration, and hence built the baseline for performance comparison using the remaining 11 features. With a highly unbalanced data set consisting of 1,092 positives and 9,674 negatives, we randomly selected 500 positive and 500 negative bootstrap samples to make a balance subset, on which 10-fold cross validation was performed for model selection and validation. The bootstrap sampling process was repeated for 100 times to obtain stable experimental results.

SVMs with RBF kernels, L1-regularized logistic regression, and kNN algorithms were used to classify proteins into two classes: drug targets or not drug targets. Each data set was split into two balanced disjoint subsets: 80% for training and 20% for testing. Tenfold cross validation was conducted on the training set to select the best model parameters ($\boldsymbol{C}$ and γ for SVMs, λ for logistic regression, and $\boldsymbol{k}$ for kNN) and calculate the training error, and then the best model was applied to the testing set to obtain the generalization error. The accuracies and optimal parameters for each classification method are summarized in Table 2, which showed that up to 75% accuracy was achieved by L1-regularized logistic regression using only 11 features.

In addition, the best models obtained from L1-regularized logistic regression using cross validation provide the coefficient of each feature in the linear function. From 100 cross-validation experiments, we obtained 100 sets of coefficients from the best models. According to Hastie *et al.* [13], the Z-score is defined as the ratio of the mean and standard deviation for each of the 11 features, and the results are shown in Table 3.

Generally a Z-score greater than 2.0 means the corresponding feature is significant to the model (confidence level 95%) [13]. From Table 3 we can find that the feature "degree" (DEG), "topological coefficient" (TPG), and "fraction of neighboring essential genes" (FReg) are trivial. Clustering coefficient (CLU), SPLs to disease genes (SPdg and avSPdg), fraction of neighboring disease genes (FRdg), and

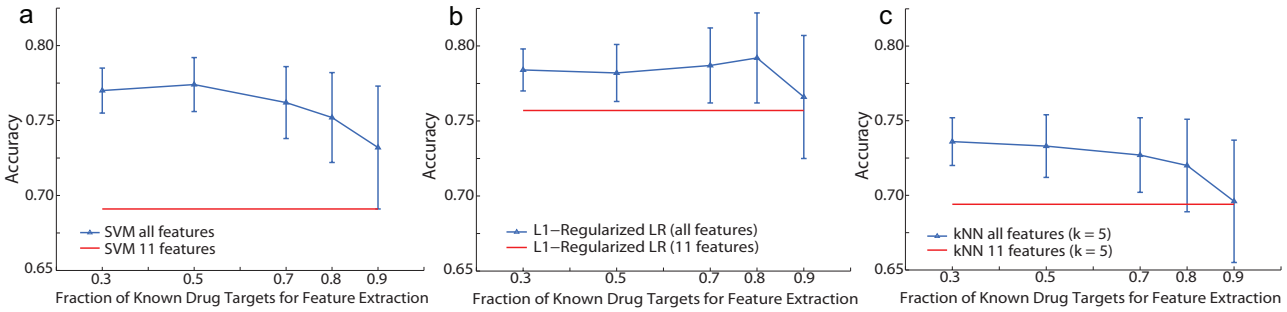


Figure 1: Summary of cross-validation accuracy using SVM, logistic regression, and kNN.

Table 4: Summary of results from cross validation experiments using different fractions of known drug targets for feature extraction. Standard deviation is computed over 200 data sets from random sampling

% Drug Targets	SVMs	LLR	kNN (k=5)
30	0.770 (0.015)	0.784 (0.014)	0.736 (0.016)
50	0.774 (0.018)	0.782 (0.019)	0.733 (0.021)
70	0.762 (0.024)	0.787 (0.025)	0.727 (0.025)
80	0.752 (0.030)	0.792 (0.030)	0.720 (0.031)
90	0.732 (0.041)	0.766 (0.041)	0.696 (0.041)

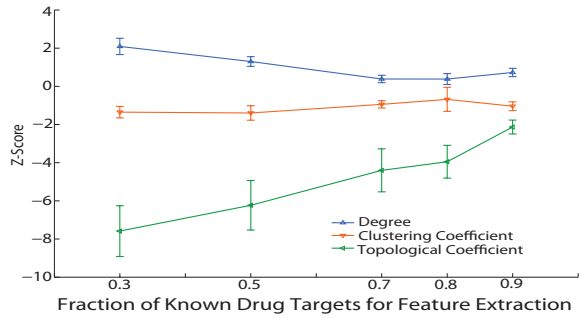


Figure 2: Z-scores of network features: degree, clustering coefficient, and topological coefficient.

characteristic distance to disease genes (CHdg) are features negatively correlated to the potential of proteins being drug targets, that is, the greater values mean lesser probable drug targets. especially for SPLs to essential genes (SPeg). Meanwhile, average SPLs and characteristic distance to essential genes are positively correlated to the probability of a protein being a drug target.

4.2 Model Selection and Assessment Using All Features

4.2.1 Model Selection and Assessment

When we take the four drug-target-related features into consideration, we need to split the 1,092 known drug targets into two sets: one for feature extraction, and the other using for positive training samples. Our experimental procedure is as follows: (1) we randomly selected 50% known drug targets for feature extraction, and the 15 features for each of the rest 50% known drug targets (546 positives) and each of the putative non-drug-target proteins (9,674 negatives) were calculated. (2) Since our data set is highly unbalanced and we have much more negatives than positives, we created a relatively balanced data set by randomly selecting 500 positives and 1,000 negatives (The number of negatives are always twice as many as positives). (3) For each such data set, we randomly selected 80% positives and 80% negatives as a training set, and the rest 20% data as a testing set. (4) Tenfold cross validation was applied to the training set to select the best models. We used grid search to find the best combination of model parameters, e.g., $\boldsymbol{C}$ and γ for SVM RBF kernels, λ for L1-regularized logistic regression, and $\boldsymbol{k}$ for kNN. (5) We tested the best models using the testing set to obtain the testing accuracy. (6) We repeated the random sampling in step 2 and the following experimental steps for 20 times to achieve stable prediction accuracy, and also repeated the random partitioning of the known drug targets in step 1 and the following steps for ten times. Therefore we conducted 200 experiments for each partitioning (i.e., 50% for above). Finally, we did the same experiments as described above using different percentages of known drug targets for feature extraction, such as 30%, 70%, 80%, and 90%. Note that the number of positives selected in step 2 were changed accordingly since the total numbers of positive samples available were different. All resulting accuracies and standard deviations obtained using SVMs, LLR, and kNN are summarized in Table 4.

From Table 4 we can find that the best accuracy obtained using SVMs, L1-regularized LR, and kNN (k=5) is 0.774, 0.792, and 0.736, respectively. To visualize the trend in the results more clearly and compare them with baseline results, we plotted the results using dot line in Fig. 1 with different colors, and the baseline results are represented with red horizontal lines. Overall, by introducing the four drug-target-related features, the cross validation accuracy obtained using SVMs, LLR, and kNN increased by 8.4%, 3.5%, and 4.2%, respectively. With up to 80% accuracy, the best models from L1-regularized logistic regression are considered meaningful for predicting potential drug targets.

Fig. 1 demonstrated that as the fraction of known drug targets used for feature extraction increases, the accuracy first slightly fluctuates (increases for SVMs and LLR, and decreases for kNN), and then decreases significantly. The main reason for this pattern is that less training data were used to learn models when more fraction of drug targets was used for feature extraction. In addition, the standard deviation bars enlarged dramatically when the fraction of drug targets for feature extraction increased, because less train-

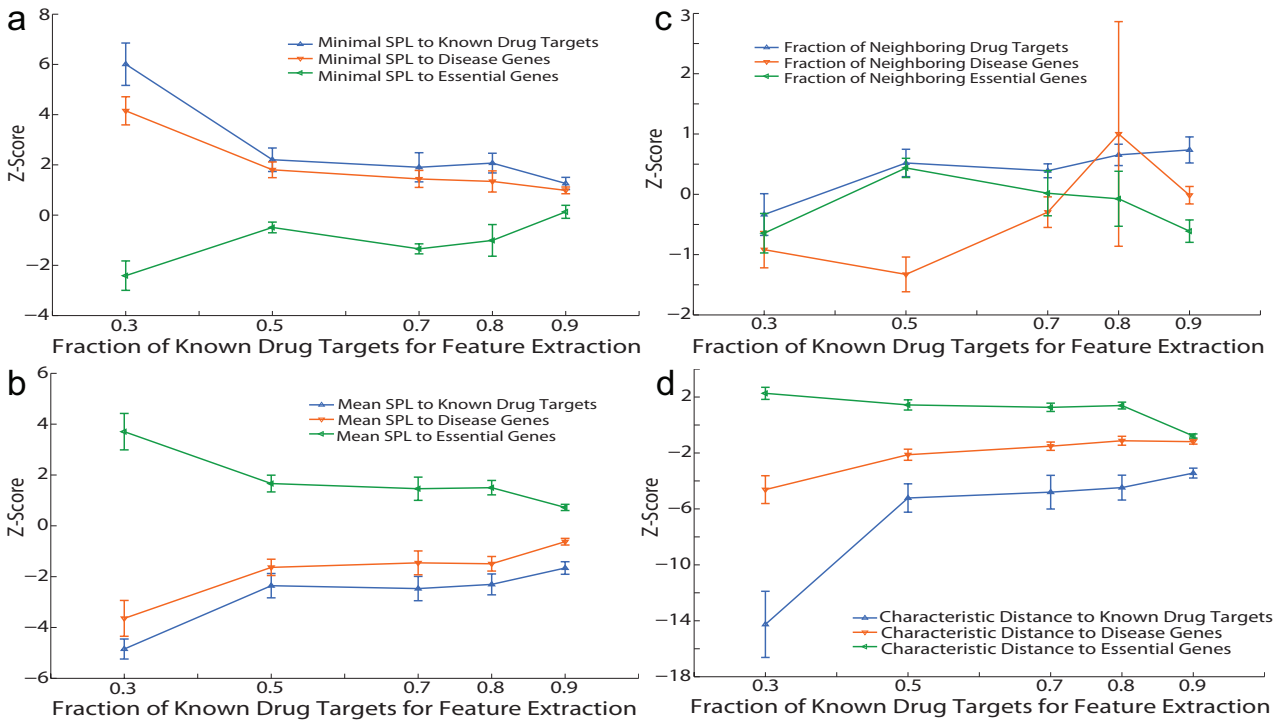


Figure 3: Summary of the Z-scores of four sets of topological features: a) Minimal SPL, b) Mean SPL, c)Fraction of neighboring special proteins, and d) characteristic distance. Each figure contains three features related to known drug targets, disease genes, and essential genes, respectively.

ing samples were used and 10 random samplings were not enough to lower the variance. There is a tradeoff between the number of training samples and the number of drug targets used for feature extraction, so the optimal partition is about 40%-60%.

4.2.2 *Relevance of Network Features*

Our L1-regularized logistic regression models provided not only cross validation accuracy, but also the coefficients of all involving features. At each partitioning we had ten random data sets, we obtained the Z-score for each data set from the 20 random samplings by calculating the ratio of the mean to the standard deviation. The Z-scores for three general network features: degree, clustering coefficient, and topological coefficient, were shown in Fig. 2. Mostly the Z-scores of feature "degree" and "clustering coefficient" are within [-2.0, 2.0], and hence have only marginal significance to the prediction results. However, topological coefficient shows significantly negative correlation to prediction results, that is, the greater the topological coefficient is, the less likely it will be a potential drug target.

The remaining 12 features make four groups, and the three features in each group are related to drug targets, disease genes, and essential genes, respectively. The mean Z-score and standard deviation of each feature was plotted in Fig. 3, and some interesting pattern can be discovered. First, fractions of neighboring drug targets, diseases genes, and essential genes are three weak predictors (marginally significant) since their Z-scores are all within [-1.0,1.0] (Fig. 3c). It is somehow intuitive since considering only direct neighbors in the PPI network is superficial.

The six SPL-related features (Fig. 3a and b) revealed interesting observations. The minimal and mean SPLs to known drug targets are both highly significant predictors, but with opposite inferences: a protein with greater minimal SPL and shorter mean SPL to known drug targets have higher probability to be potential drug targets. Observations on disease genes are quite similar, but SPLs to disease genes have only marginal significance. For essential genes, the significance is also very marginal, although they tell that a protein with shorter minimal SPL and greater mean SPL to essential genes has higher chance to be potential drug targets.

Moreover, the characteristic distance to known drug targets (Fig. 3d) was found the most significant predictor since its Z-scores are very negative, showing that proteins with shorter characteristic distance to all known drug targets have higher chance to be drug targets, which is intuitive because shorter characteristic distance means topologically more similar. Similar to SPL, characteristic distance to disease genes was similar but marginally important predictors. Characteristic distance to essential genes was unsurprisingly marginally correlated to prediction results.

Finally, it is noticeable that Z-scores at fraction of known drug targets equal to 0.3 are more significant than other values. The reason is that when small number of known drug targets were used for feature extraction, the advantages of the four drug-target-related features were weakened, and hence all the 15 features are all important for building models.

Table 5: Comparison of the performance of our method and the method by Zhu *et al.* Standard deviation is computed over 100 data sets from random sampling

% Drug Targets for	SVMs		LLR		kNN (k= 5)	
Feature Extraction	Our method	Zhu *et al.*	Our method	Zhu *et al.*	Our method	Zhu *et al.*
30	0.794+/-0.015	0.695+/-0.017	0.795+/-0.014	0.722+/-0.016	0.758+/-0.017	0.677+/-0.021
50	0.789+/-0.018	0.696+/-0.024	0.790+/-0.018	0.729+/-0.019	0.752+/-0.016	0.664+/-0.023
70	0.794+/-0.022	0.691+/-0.028	0.797+/-0.019	0.736+/-0.026	0.774+/-0.020	0.665+/-0.033
80	0.796+/-0.031	0.680+/-0.034	0.800+/-0.026	0.747+/-0.028	0.772+/-0.028	0.667+/-0.040
90	0.770+/-0.041	0.666+/-0.039	0.786+/-0.041	0.723+/-0.039	0.754+/-0.041	0.654+/-0.050

Table 6: Comparison of the performance of our method and the method by Li *et al.* Standard deviation is computed over 100 data sets from random sampling

% Drug Targets for	SVMs		LLR		kNN (k= 5)	
Feature Extraction	Our method	Li *et al.*	Our method	Li *et al.*	Our method	Li *et al.*
30	0.754+/-0.022	0.710+/-0.042	0.775+/-0.022	0.692+/-0.033	0.723+/-0.020	0.651+/-0.037
50	0.753+/-0.025	0.700+/-0.032	0.771+/-0.026	0.699+/-0.019	0.713+/-0.023	0.660+/-0.024
70	0.731+/-0.031	0.696+/-0.013	0.765+/-0.030	0.691+/-0.012	0.703+/-0.030	0.649+/-0.016
80	0.737+/-0.041	0.688+/-0.009	0.794+/-0.036	0.679+/-0.012	0.710+/-0.037	0.645+/-0.009
90	0.732+/-0.046	0.669+/-0.015	0.739+/-0.055	0.666+/-0.008	0.689+/-0.050	0.622+/-0.017

4.2.3 Comparison with Previous Work

Zhu *et al.* [27] proposed a SVM classification method to predict potential drug targets using five network topological features: degree, clustering coefficient, 1N index, shortest distance to drug targets, and average distance to drug targets, which are included in our 15 feature set. To compare the performance of our feature set with Zhu *et al.* [27], we applied a similar cross-validation experimental procedure: a) Partitioned the set of drug targets into two parts: one for feature extraction, and the other used for training; b) Randomly selected 30%, 50%, 70%, 80% and 90% of drug targets for feature extraction, and repeated the random sampling for 10 times; c) Randomly selected a balanced data set consisting of twice as many negative as positive samples, and repeated the random sampling for 10 times. We used grid search to select the best parameters C and γ for RBF kernels.

The results for our method and Zhu *et al.*'s method [27] were listed in Table 5 for comparison. For each partitioning of drug targets, our best five features (characteristic distance to known drug targets and to disease genes, minimal and mean SPL to known drug targets, and topological coefficient) outperformed the method by Zhu *et al.* for 9.3-11.6% using SVMs, 5.3-7.3% using L1-regularized logistic regression, and 8.8-10.9% using kNN at k = 5. A simple explanation of the superiority of our feature set was that our method systematically not only integrated information from both drug targets and disease genes, but also integrated information from many drug targets and/or disease genes into one feature. Our experimental results demonstrated that disease gene information did help identify potential drug targets.

In addition, we also compared our feature set with the protein sequence features used by Li *et al.* [18]. We downloaded all protein sequences from UniProt (http://www.uniprot.org), mapped all human proteins onto the UniHI network, and obtained 1,075 drug target sequences, and 7,099 putative non-drug-target sequences. We then applied Needleman-Wunsch global alignment algorithm [20] to calculate the pairwise sequence identities for each of the sequence sets, and iteratively removed protein sequences to cull all sequences in each set with a given identity threshold (e.g. 30% in this work). Eventually we achieved 660 drug target sequences and 5,006 non-drug-target sequences, and in each set no pairwise sequence identity are greater than 30%. We then calculated exactly the same 146 protein sequence features as in Li *et al.* [18] using the online server PROFEAT by Chen *et al.* [19].

In the experiments, both methods only used the 660 positive and 5,006 negative samples to conduct cross validation at different sizes of training set. We assured that the numbers of training samples for both methods are very close to each other. At each size of training set, we randomly selected 50 data sets with approximately twice as many negative as positive samples, and the final accuracy and standard deviation were obtained by averaging results over the 50 experiment, as shown in Table 6. Clearly our feature set outperformed the 146 sequence features for 3.5-6.3% using SVMs, 7.4-11.5% using L1-regularized logistic regression, and 5.3-7.2% using kNN at k=5. The best accuracy was 79.4% using our method, but 71% using the sequence features.

5. CONCLUSIONS

In this paper, we proposed a set of 15 topological features extracted from human PPI networks, applied sophisticated machine learning algorithms such as SVMs, logistic regression, and kNN to construct highly accurate models using these features to predict whether a human protein can be a drug targets or not, and achieved excellent performance with up to 80% prediction accuracy. In addition, we analyzed the correlation of each topological feature to the probability of being drug targets for human proteins by calculating the Z-scores of the model coefficients obtained by L1-regularized logistic regression, and found that some topological features were highly important to the druggability of a protein, such as characteristic distance to drug targets, shortest and average distance to drug targets and disease genes, and topological coefficients. Finally, we compared the performance of our feature set with two previous work, and observed that our method outperformed them for 5-11% higher accuracy. Analysis demonstrated that the superiority of our fea-

tures was originated from on highly integrated information from many drug targets and from both known drug targets and disease genes simultaneously. Our feature extraction only rely on the interacting profiles of proteins, and doesn't need any additional information such as protein sequences or 3D structures, therefore they can by easily applied to many other applications.

6. ACKNOWLEDGMENTS

This work has been supported by the KU Specialized Chemistry Center (NIH U54 HG005031) and NSF grant IIS 0845951.

7. REFERENCES

[1] C. P. Adams and V. V. Brantner. Estimating the cost of new drug development: Is it really $802 million? *Health Aff*, 25(2):420–428, 2006.

[2] M. Ahram, Z. I. Litou, R. Fang, and G. Al-Tawallbeh. Estimation of membrane proteins in the human proteome. *In Silico Biology*, 6:0036, 2006.

[3] T. M. Bakheet and A. J. Doig. Properties and identification of human protein drug targets. *Bioinformatics*, 25(4):451–457, 2009.

[4] A. Barabasi and Z. Oltvai. Network biology: understanding the cell's functional organization. *Nat. Rev. Genet.*, 5:101–113, November 2004.

[5] C. J. Burges. A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery*, 2(4):121–167, 1998.

[6] S. P. Butcher. Target discovery and validation in the post-genomic era. *Neurochem. Res.*, 28(2):367–371, February 2003.

[7] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[8] G. Chaurasia, Y. Iqbal, C. Hanig, H. Herzel, E. E. Wanker, and M. E. Futschik. UniHI: an entry gate to the human protein interactome. *Nucl. Acids Res.*, 35:D590–594, 2007.

[9] G. Chaurasia, S. Malhotra, J. Russ, S. Schnoegl, C. Hanig, E. E. Wanker, and M. E. Futschik. UniHI 4: new tools for query, analysis and visualization of the human protein-protein interactome. *Nucl. Acids Res.*, 37:D657–660, 2009.

[10] T. M. Cover and P. E. Hart. Nearest neighbor pattern classification. *IEEE Trans. on Info. Theo.*, 13(1):21–27, 1967.

[11] J. T. Eppig, C. J. Bult, J. A. Kadin, J. E. Richardson, J. A. Blake, and the Mouse Genome Database Group. The Mouse Genome Database (MGD): from genes to mice–a community resource for mouse biology. *Nucl. Acids Res.*, 33:D471–475, 2005.

[12] P. J. Hajduk, J. R. Huth, and C. Tse. Predicting protein druggability. *Drug Discovery Today*, 10(23-24):1675 – 1682, 2005.

[13] T. Hastie, Tibshirani, Robert, and J. Friedman. *Elements of Statistical Learning: Data Mining, Inference, and Prediction, 2nd ed.* Springer, 2009.

[14] A. L. Hopkins and C. R. Groom. The druggable genome. *Nat. Rev. Drug Discov.*, 1(9):727–730, 2002.

[15] D. W. Hosmer and S. Lemeshow. *Applied Logistic Regression, 2nd ed.* Wiley-Interscience Publication, New York, 2000.

[16] K. Koh, S.-J. Kim, and S. Boyd. An interior-point method for large-scale l1-regularized logistic regression. *J. Mach. Lear. Res.*, 8:1519–1555, 2007.

[17] I. Kola and J. Landis. Can the pharmaceutical industry reduce attrition rates? *Nat. Rev. Drug Discov.*, 3(8):711–716, 2004.

[18] Q. Li and L. Lai. Prediction of potential drug targets based on simple sequence properties. *BMC Bioinformatics*, 8(1):353, 2007.

[19] Z. R. Li, H. H. Lin, L. Y. Han, L. Jiang, X. Chen, and Y. Z. Chen. PROFEAT: a web server for computing structural and physicochemical features of proteins and peptides from amino acid sequence. *Nucl. Acids Res.*, 34:W32–37, 2006.

[20] S. B. Needleman and C. D. Wunsch. A general method applicable to the search for similarities in the amino acid sequence of two proteins. *J. Mol. Biol.*, 48(3):443–453, 1970.

[21] J.-F. Rual and *et al.* Towards a proteome-scale map of the human protein-protein interaction network. *Nature*, 437:1173–1178, October 2005.

[22] U. Stelzl and *et al.* A human protein-protein interaction network: A resource for annotating the proteome. *Cell*, 122(6):957 – 968, 2005.

[23] D. S. Wishart, C. Knox, A. C. Guo, D. Cheng, S. Shrivastava, D. Tzur, B. Gautam, and M. Hassanali. DrugBank: a knowledgebase for drugs, drug actions and drug targets. *Nucl. Acids Res.*, 36:D901–906, 2008.

[24] D. S. Wishart, C. Knox, A. C. Guo, S. Shrivastava, M. Hassanali, P. Stothard, Z. Chang, and J. Woolsey. DrugBank: a comprehensive resource for in silico drug discovery and exploration. *Nucl. Acids Res.*, 34:D668–672, 2006.

[25] M. A. Yildirim, K.-I. Goh, M. E. Cusick, A.-L. Barabasi, and M. Vidal. Drug-target network. *Nat. Biotech.*, 25(10):1119–1126, 2007.

[26] C. J. Zheng, L. Y. Han, C. W. Yap, Z. L. Ji, Z. W. Cao, and Y. Z. Chen. Therapeutic Targets: Progress of Their Exploration and Investigation of Their Characteristics. *Pharmacological Reviews*, 58(2):259–279, 2006.

[27] M. Zhu, L. Gao, X. Li, and Z. Liu. Identifying drug-target proteins based on network features. *Sci. in China Ser. C: Life Sci.*, 52(4):398–404, 2009.

Alignment-free Sequence Analysis Using Extensible Markov Models

Rao M. Kotamarti
Southern Methodist University
Dallas, Texas, USA
mallik@kotamarti.com

Margaret H. Dunham
Southern Methodist University
Dallas, Texas, USA
mhd@lyle.smu.edu

ABSTRACT

Profile models based on Hidden Markov Models (HMM) for sequence studies have gained visibility among researchers. While the mathematical foundation, the proven algorithms such as Viterbi, Forward and Backward algorithms have certainly provided a rigorous probabilistic platform, the requirement of classic alignment has ensured an extremely high time complexity. We propose the use of another kind of Markov model called Extensible Markov Models (EMM) to create profile architectures that are more efficient in space and time complexity than their HMM counter parts. EMM efficiency comes from an alignment-free paradigm through use of an improved statistical signature form of sequences. The EMM aproach is based on the use sliding p-mers that count every possible p-mer pattern along equal sized segments of a sequence which are then clustered into Markov states. The resulting count vectors shift the position based letter-by-letter sequence analysis problem for phylogenetic trees, classification and search to a more efficient numerical vector space. Using adapted Karlin-Altschul statistics from the Basic Local Alignment Search Tool (BLAST) literature, the EMM based sequence classification also computes a p-value for statistical significance. We present a comparison between profiles generated using profile HMM and EMM.

Categories and Subject Descriptors

H.4 [**Bioinformatics**]: Techniques; D.2.8 [**Phylogenetics and Comparative Genomics**]: Models

General Terms

Profile Markov Models, HMM, EMM

1. INTRODUCTION

Profile, as used today in Bioinformatics, usually refers to the conserved residue distribution in consensus columns of a multiple sequence alignment [8]. BLAST PSI [1]and ProfileHMM employ profiling in order to improve accuracy of homology prediction. But, this generalization does not include long range correlations as found in the RNAs due to Watson-Crock complementarity for which a more involved form called *covariance models* are defined [9]. Several comparative studies have shown that profile methods outperform sequence methods in predicting homology [18]. However, profile methods based on HMMs run slower than BLAST PSI. Since there appears to be a tradeoff between accuracy and response time, it is at the discretion of the researcher as to which method to use depending on the research needs.

Blaisdell [3] demonstrated the information theoretic basis of using p-mers of sufficient width to uniquely represent sequences in a clustering approach without using alignment. Vinga [21] presented an excellent review of alignment-free approaches ranging from the use of distance methods to complexity theory. Alignment-free analysis is unsusceptible to translocation of sequence fragments during genetic recombination, while this still remains a concern in alignment approaches.

In our work we use *profile* in a more general sense, as a summary of the important statistical properties of one or many sequences. This view allows for a broader consideration of sequence data within a single or multiple sequences. Extended Markov Models (EMM) [6] are different from Hidden Markov Models in that EMM states are not hidden and they are representative of similar sequence fragments found in one or more related sequences. In addition, due to the clustering and independence of positional constraints, EMM representations of sequences and sequence communities are more compact than HMMs. Successful application of EMMs in phylogeny and classification have been reported earlier in [16, 15]. With roots in data mining, EMM techniques involve learning from stream data and EMMs were successfully applied to outlier detection [13] and future event predictions [5].

Markov models are predominant in profile representations of sequences. The profile Hidden Markov Model [17] takes a multiple sequence alignment generated elsewhere (for example, ClustalW [20]) and build detailed 'tri-state per position' models. Inheriting the states *Match*, *Insert* and *Delete*, the profileHMM assumes a comprehensive Markov model as shown in Figure 1. Once configured, dynamic programming algorithms, Viterbi and Forward, are available for scoring a sequence against a model [7].

Since an EMM state involves several contiguous positions of a sequence, it is possible to apply a parametric model us-

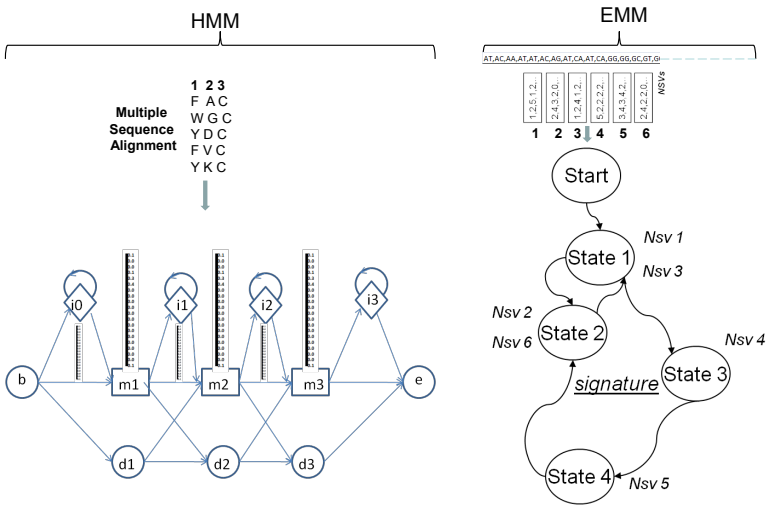


Figure 1: The profile HMM and EMM. These are examples of profile Hidden Markov Model (HMM) and Extensible Markov Model (EMM). The HMM is representing three consensus columns. The probability distributions for insertion and match states are shown. A multiple sequence alignment is used as input. The resulting profileHMM consists of three states for each (consensus) position in the alignment along with the emission probabilities for the match and insertion states. The arcs among states represent the transition probabilities. The EMM is representing a single sequence with 5 equal sized segments. Numerical Summary Vectors (NSV) constitute the numerical representations of equal sized segments along a molecular sequence which are used in building an EMM signature. Signature building starts with a start state; as each NSV is processed, it is compared to the existing states of the model. If the NSV is not found to be close enough (per a Squared Euclidean threshold) as in the case of NSV 1, a new state (1) is created with the new NSV as its first cluster member; otherwise, the new NSV (as in the case of NSV 3) is simply added to the matching cluster state node (state 1). When all NSVs are processed, the model building process is finished.

ing Karlin-Altschul statistics which allows assessing statistical significance also. In fact, EMM signatures of sequences allow comparison, classification and search more efficiently than the HMM counter parts.

The rest of the paper is organized as follows: Section 2 compares the model building and the models generated by profileHMM and EMM along with sequence scoring methods; section 3 describes experiments with DNA and RNA sequences; sections 4 and 5 finally conclude with a summary of results.

2. PROFILE MODELS: HMM AND EMM

An example of both models is shown in Figure 1 which differentiates the alignment based profileHMM and the alignment-free EMM.

2.1 Model Building

For building HMMs, in particular the profileHMMs, multiple sequence alignment (MSA) is performed first using well known tools such as ClustalW [20]. The original paper for profileHMMs by Krogh [17] and some subsequent literature employ profileHMM itself to progressively build and improve multiple sequence alignment; however, more recently, this functionality has been replaced by more optimal techniques such as *ClustalW* that are freely available. The MSA is input to build profileHMMs which take the form as shown in Figure 1. In profileHMMs, consensus columns are mapped into Match states M and additional states such as Insert I and Delete D states are used. Since a consensus column may use different symbols, a probability distribution is specified representing the emission probabilities, so called in the HMM nomenclature. Since insertion regions can be of different lengths and occur between consensus columns, they too are associated with a probability distribution for the residues that occur in those regions as shown in the figure. The transition probabilities are associated with transitions among the three states though typically transitions between Insertion and deletion in either direction are rare. Each consensus position within a sequence community uses three states and two lists of distribution probabilities. The time complexity for building profileHMMs including the pre-requisite multiple sequence alignment is $O(M^2+N^2LogN)$ where M is the number of sequences and N is the size of a sequence.

Building EMMs does not require multiple sequence alignment, but does require counting the occurrences of words or p-mers within each sequence where p is typically 2 or 3. Counting is done by sliding a p-mer window over the sequence and counting the corresponding p-mer letter pattern. If $p = 3$ is selected, there would be $4^3 = 64$ different combinations for a DNA sequence where the alphabet consists of 4 letters $A, C, Tand\ G$. Actually for an EMM, a sequence is first divided into equal segments of size k and the count vector of occurrences is created for each segment. Such vectors are called Numerical Summarization Vectors (NSVs) in EMM nomenclature [15]. The NSVs thus prepared for each sequence in a group of sequences are organized into a Markov model as shown in Figure 2. The nodes of an EMM represent clusters of related segments in NSV form. As such, each new NSV is first compared to each node's cluster to determine its placement. In case, such placement is not possible, i.e, no cluster is a reasonable choice based on a squared Euclidean distance threshold, a new node is created starting yet an-

other cluster. During the building process, transitions are counted between nodes which will subsequently be used for scoring.

During the building process of a profile EMM, a Log-Odds Score matrix of dimension 4^pX1 is created where p is the value corresponding to p-mer used. For example, if 3-mer is used, i.e., if $p = 3$, the score matrix of size $4^p = 64X1$ is created where each entry represents one of the 3-mer variations. Kotamarti et al describe the algorithms for creating the LOD score matrices for EMM in [16]. The score matrix is used in adapting Karlin-Altschul statistics during homology assessment for a query sequence of interest. Since clustering is employed, the number of states and therefore the space used by an EMM tends to be a lot smaller than for profileHMMs.

The time complexity for building the EMMs for sequence analysis is $O(NM/K)$ where M is the number of sequences, N is the size of a sequence and K is the number of equal sized segments in a sequence. For Profile HMM the space complexity is O(MN). For Profile EMM it is O(K) where K is the number of clusters. In the worst case this is O(N).

2.2 Model analysis

ProfileHMM captures and retains the alignment inherited from the multi-sequence alignment that was provided as input and adds a probabilistic framework for consensus columns and gaps. ProfileEMM captures and retains the alignment-free summary representation and adds a Log-ODDs Score table to use in scoring. In short, profileEMM consolidates multiple sequence statistical signatures into a compact model for further processing.

In addition, the profileEMM naturally reflects the most frequently used sequence segments and transitions as shown in Figure 3. As may be seen in the figure, profileEMM captures the conserved segments and transitions regardless of where the segments occur along a sequence or across a community of related sequences. Since segment sizes are usually much smaller than the sequences, longer stretches of similar sequence fragments are shown as a chain of frequently occurring nodes and transitions.

ProfileHMMs make use of the Viterbi algorithm for analysis. Since Viterbi algorithm is known for generating the *survivor or the most likely path*, it could possibly generate a useful presentation; however, due to the underlying alignment basis used, the presentation would not be able to reflect long range correlations or translocated yet conserved stretches of sequences. To the best of our knowledge, there has been no attempt to use Viterbi for such analysis for profileHMMs.

Unlike ProfileHMMs, profileEMMs are used to represent both DNA and RNA. Straight alignment column-wise without consideration to base-pairing could reject valid homologous sequences. As such profileHMM is not used for representing RNA. A variant that borrows much from profileHMM and adds more complexity to the state framework called *covariance models* is used to model RNA. But, the processing complexity limits the length of RNA to less than 500 [7, 9]. On the other hand, profileEMM is usable for RNA as empirically established by [15]. The statistical signature appears to more than adequately handle the base-pairing induced changes and estimate the classification correctly [16].

Yet another difference lies in how the proteins are represented in profileEMM. Amino acid representations are typ-

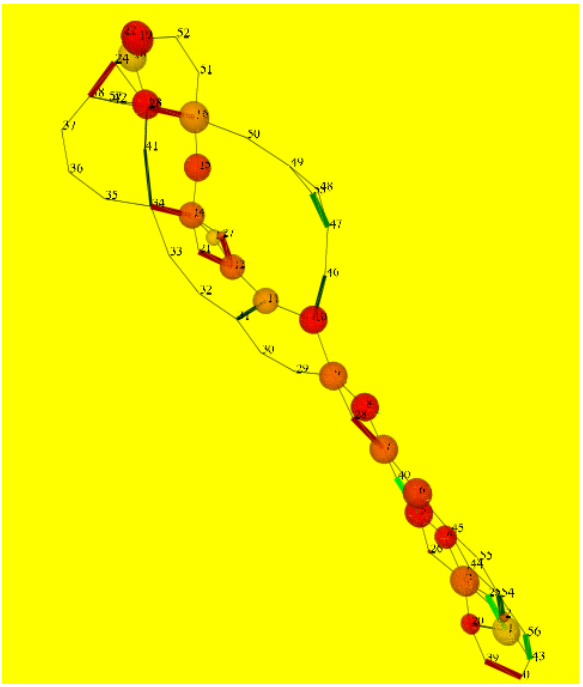


Figure 3: ProfileEMM visualization. This profile EMM visualization for the genus EColi captures the summary of 146 sequences. The graph reflects the most frequently occurring sequence segments and transitions between segments. The least frequently occurring nodes and transitions are small and thin. Size of the nodes in the graph indicates the conserved segments of sequences which may occur at different points across different sequences. Color and width of arcs represents the relative occurrence of transitions between the segments of certain composition. The conserved regions and transitions are thus captured regardless of the location of the segments.

ically used in profileHMMs though straight DNA can also be handled. Due to the sliding p-mer approach used in profileEMM, the native DNA sequence form is used for representing the protein signatures. Since protein sequence analysis deals with motifs that can occur any where and be of variable length, the NSV generation for a protein profileEMM differs from RNA profileEMM as follows:

1. If native sequence in the form of Amino acids, convert to the most likely DNA.
2. For a protein profileEMM, divide the sequence into maximum overlapping equal sized segments where as for a RNA profileEMM, divide the sequence into non-overlapping equal sized segments. Use smaller segment sizes <20 to make it useful for motif finding from protein profiles.
3. For a protein profileEMM, use 3-mers as they tend to signify codon paradigm.

Figure 4 shows a profileEMM for a select globin family of two proteins, both of which have similar primary and tertiary structure (amino acid sequence and folding). These proteins all incorporate the globin fold, a series of eight alpha helical segments. The table 1 summarizes the differences in model building and representations between profileHMM and profileEMMs.

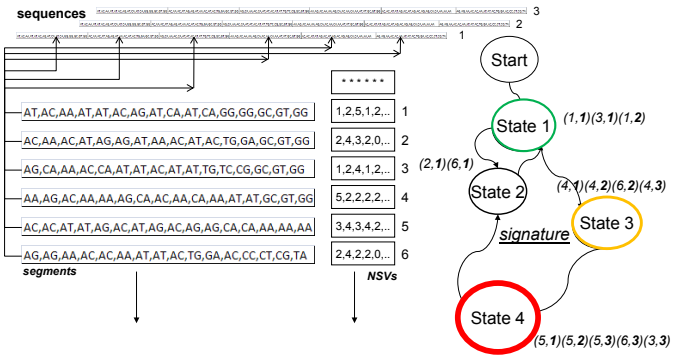


Figure 2: Example of a Profile EMM of three sequences. **The NSVs from three sequences are consolidated into a single EMM. The notation (x,y) represents NSV x from sequence y. The state with the most NSVs clustered is shown to be relatively larger for clarity. However, only the mean representation, i.e., the centroid vector is maintained per state. Thus the EMM is compact for modeling multiple sequence profiles.**

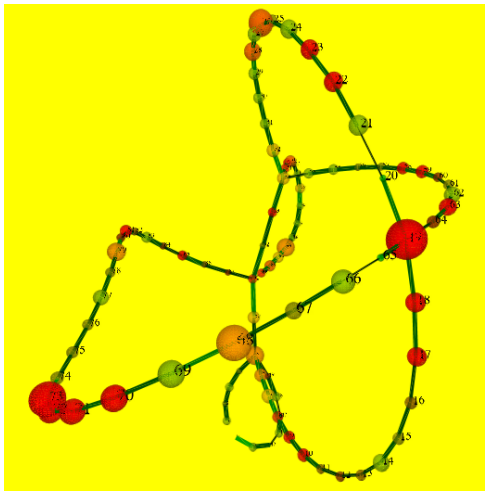


Figure 4: ProfileEMM visualization. **This profile EMM visualization is an example for a two member protein family called *Globins* which have similar primary and teritary structures.**

Profile HMM	Profile EMM
• Alignment based • Requires multiple sequence alignment as input. • Larger space needed to maintain M/I/D states per position. • Used for modeling DNA & Amino Acids. • Annotates conserved positions, insertions and deletions.	• Alignment-free • Requires p-mer based statistical signatures. • Compact due to clustering. Number of states << size of sequence. • Used for modeling DNA & RNA. • Most frequently occurring sequence segments and transitions are readily shown in the model built.

Table 1: Modeling styles in profile models. **The table highlights the differences in model creation and model presentation between the profile HMM and EMM. The compact representation and not requiring multiple sequence alignment places profileEMM ahead of profileHMM.**

2.3 Sequence Scoring

ProfileHMM does not use ad hoc scores, but uses probabilities instead. The consensus columns imported from multiple sequence alignment are used to create emission probabilities. Similarly, the transitions among the three states at every position along the sequence are counted and normalized to use as adjacent probabilities. In the basic profileHMM architecture, this is all that is required. However, the subsequent and recent profileHMM implementations include structural information and Dirichlet mixtures [7] to fine tune the emission probabilities thus improving the underlying model. No ad hoc scores are applied for gap cost in profileHMMs. They are derived simply from the probabilistic framework itself using the known transitions from insert and match states. The scoring for a sequence against a profileHMM is done as follows:

1. Convert sequence to a profileHMM.
2. Determine the most likely path.
3. Compute the Viterbi score along the path, or.
4. Compute the forward score along the path.

The difference between Viterbi and Forward scoring is that the former considers whichever of the three states - match, insert or delete generates the highest probability where as the latter sums all of them. Though Viterbi generates the optimal probability, Durbin etal comments that in practice the Forward scoring is more accurate [7]. The time complexity for a single model evaluation is given as $O(NM_s^2)$ where N is length of a sequence, but M_s is number of states which can be three times the number of consensus positions. Since an evaluation of a sequence is typically done against a database of existing models, we will describe this next before looking at EMM scoring details.

To the best of our knowledge, the existing profileHMM literature or the current implementations are not published to the detail needed to perform a careful analysis. But, the basic algorithms used for evaluating a single model against a query are well understood to be based on Viterbi and the forward algorithm based on dynamic programming [7]. The book by Sean Eddy et al [7] describes these algorithms and various facets of them in great detail. The time complexity for a single model evaluation is given as $O(NM_s^2)$ where N is length of a sequence, but M_s is number of states which

can be three times the number of consensus positions. In case of a search of a single query against a large database of models, this can be extrapolated with a multiplication factor equivalent to the size of the database in terms of models.

Since the HMM based algorithms are well known, we will present only the details of the EMM method of model evaluation for a query as follows: For each EMM in the model database,

1. For each NSV s_i (which is a count vector for a segment) of the query sequence,
   (a) Find the nearest match state s'_i of the model referred to as *quasi alignment*.
   (b) Score the quasi alignment as Q_i.
   (c) Apply a weight w_i of 1 or a penalty ϵ if the transition between the previously matched state and the current matched state is not present in the model.
2. Once all the NSVs of the query sequence are thus processed, generate the overall score M_j where j is the model number which ranges over all the available models in the database by using the equation $M_j = \frac{1}{1+\sum w_i Q_i}$.

Profile HMM	Profile EMM
• Based on the probabilities of HMM framework (e.g. emission and adjacent probabilities). • Standard dynamic programming algorithms are used (e.g. Viterbi and HMM Forward algorithms). • No inherent support for reporting statistical significance. • Adds a probabilistic basis for generating phylogeny .	• Based on LoD scores of Karlin-Altschul statistics framework (e.g. BLAST) • Sum-scores of quasi-alignments used as sequence score (e.g. scoring a sequence with multiple local alignments in BLAST). • Uses adapted Karlin-Altschul statistics to report statistical significance. • Inherent support for Phylogeny (e.g. Reciprocal of Sum-score is a valid distance metric)

Table 2: Scoring in profile models. The table lists the highlights of scoring in both profile HMM and EMM. Both use strong roots and are well supported, but profileEMM outperforms profileHMM by not using the dynamic programming algorithms and by having inherent support for statistical significance reporting.

2.4 Score Matrix for EMM

The key differentiating aspect of BLAST is its successful use of a heuristic justified with sound theory of statistical significance which has come to be known as Karlin-Altschul statistics [14]. Initial implementations of profileHMM did not include any form of significance reporting though more recent versions do report a form of E-score. We describe next how profileEMM combines scoring and significance reporting.

Based on principles of Karlin-Altschul statistics as used in BLAST, profileEMM sets up a Gumbel extreme value distribution for scores, but uses a slightly different method for determining statistical significance during the scoring process. Unlike the well known score matrices like BLOSUM and PAM used by BLAST, profile EMM requires dynamic generation of score matrices for each EMM [16]. High level summary is presented as follows:

1. Create LOD score matrix for each model.
2. Adjust the scores when assessing membership using the individual probabilities of the bases in a member's sequence(s).
3. Fit a Gumbel distribution by computing the parameters λ and K using numerical methods at `http://www.ncbi.nlm.nih.gov/IEB/ToolBox/C_DOC/lxr/source/tools/blast.c`.

With an adjusted score matrix for each query-model pair, quasi alignments are scored as weighted Manhattan distance between the query NSV and the matched model state's centroid details of which are described in [16]. Each NSV of the query is scored and aggregated to generate a sequence wide score for a query-model pair. Since there are many models (profileEMMs), the one with the highest score is chosen to be the most homologous. Table 2 summarizes the scoring schemes for both profile modeling methods.

3. EXPERIMENTS

Designing experiments to compare profile HMM and EMM is quite difficult. Ideally, to compare profileEMMs and profileHMMs, one would have to build model libraries for the same data and test the same query sequences against both. However, this is also impractical since profileHMM libraries are large and have been built over a long period of time. The approach we take instead is to use some profileHMM datasets and use them in a profileEMM environment and see how they perform with respect to accuracy and computational resources.

Two datasets are used in our experiments. Five experiments are performed as follows: 1) Build models and compare methods; 2) Build phylogeny for DNA; 3) Assess homology using profileEMM for DNA; 4)Perform RNA based phylogeny and classification; and 5) Repeat experiment (3) using Viterbi and Forward algorithms.

RNA dataset.

For our RNA analysis,we will consider using the 16S rRNA sequences of microbial organisms. 16S rRNA, a part of ribosomal RNA, is an essential and ubiquitous gene sequence and it is commonly collected and used for microbial identification [4] and classification. The 16S rRNA Database utilized in this analysis is derived from the NCBI Microbial Complete Genome Database at `http://www.ncbi.nlm.nih.gov/genomes/lproks.cgi`.

The specific dataset that is used is from the phylum *Euryarchaeota* and there are 33 strains included in the test.

DNA Dataset.

For our DNA data we decided to use a small sample from the book by Durbin et al [7] which is used in describing profileHMMs. This dataset is presented in table 3 and lists 7 globins, a representative protein family, in the form of their identifiers in the SCOP database [7]. We chose another dataset that is unlabeled from the same source [7] to allow us to analyze two datasets together in phylogeny and homology experiments. This dataset is described in the table 4.

Experiment 1: Build models for DNA and RNA.

The Figures 3 and 4 show the profileEMMs for the 16S rRNA for *E.Coli* and a 2-member globin family consisting of *HBB and HBA HUMAN*. These were generated to highlight

	amino acid sequence	codons
amino1	FPHFDLSHGSAQ	UUUCCUCAUUUUGAUUUAUCUCAUGGUUCUGCUCAA
amino2	FESFGDLSTPDFAVMGNPK	UUUGAAUCUUUUGGUGAUUUAUCUACUCCUGAUUUUGCUGUUAUGGGUAAUCCUAAA
amino3	FDRFKHLKTEAEMKASED	UUUGAUCGUUUUAAACAUUUAAAAACUGAAGCUGAAAUGAAAGCUUCUGAAGAU
amino4	FTQFAGKDLESIKGTAP	UUUACUCAAUUUGCUGGUAAAGAUUUAGAAUCUAUUAAAGGUACUGCUCCU
amino5	FPKFKGLTTADQLKKSAD	UUUCCUAAAUUUAAAGGUUUAACUACUGCUGAUCAAUUAAAAAAAUCUGCUGAU
amino6	FSFLKGTSEVPQNNPE	UUUUCUUUUUUAAAAGGUACUUCUGAAGUUCCUCAAAAUAAUCCUGAA
amino7	FGFSGASDPG	UUUGGUUUUUCUGGUGCUUCUGAUCCUGGU

Table 4: Unlabeled PFAM data. **The table lists protein sequences of 7 unlabeled species from the book by Durbin et al [7]. Conversion from proteins to codons is done using ExPASy tools [12].**

SCOP Identifier	Sequence Identifier
HBA_HUMAN	Homo-sapiens-GN
HBB_HUMAN	Homo-sapiens-GNHBB
MYG_PHYCA	Physeter-catodon-GN
GLB3_CHITP	Chironomus-Thummi-piGer
GLB5_PETMA	Petromyzon-marinus
LGB2_LUPLU	Lupinus-luteus-PE
GLB1_GLYDI	Glycera-dibranchiata

Table 3: Globin family of proteins. **The table lists the SCOP [19] identifiers for 7 globins which from a protein family. The proteins are extracted from UNIPROT database [2] and then converted to codons using ExPASy tools [12].**

the conserved regions as readily shown by the profileEMM. In this experiment, which also serves as a preparation step for the other experiments, profileEMMs are generated for all DNA and RNA sequences used and organized into model libraries. As described in the model building section 2, the sequences are segmented into equal size segments, converted to NSVs using a 3-mer counting. In case of DNA models, maximum overlap is utilized to improve classification performance in the upcoming analyses. For a profile of 282 sequences, building profileEMM on a dual-pentium laptop took less than 30 minutes where as the same took well over 30 minutes on the online server based profileEMM; these times also include 3-mer counting and alignment respectively.

Experiment 2: DNA phylogeny using profileEMM.

The two protein families from Tables 3 and 4 are used for this experiment. ProfileEMM distance metric which conforms to the formal constraints for a valid metric [16] is used to compute the distance matrix for five members from each family as shown in table 5. A phylogenetic tree using BIONJ neighbor joining algorithm [11] is generated (not shown) using the PHYlogenetic Inferenece Pacakage (PHYLIP) [10] on the web. The corresponding phylogenetic tree analysis interestingly indicates that some members of the unlabeled proteins may in fact be closer to the globin family members. Though this cannot be confirmed, the source for the data from the book by Durbin et al [7] seems to have come from the same 7 genomes though from different protein sequences.

Experiment 3: DNA homology assessment with profileEMM.

ProfileEMM classification algorithm described by Kotamarti et al in [16] is based on a Log-odds scoring scheme using Karlin-Altschul statistics [14]. This classifier is used to search for the most homologous model given an individual protein from the two sample protein datasets, i.e, 3 and 4. For this experiment, the models built in the first experiment are utilized as the model library against which a given protein sequence is evaluated. In other words, classifying a protein sequence basically involves first generating a model, then exhaustively evaluating all models in the library to determine the best matching model. It may be noted, unless the models have high specificity in the library and the evaluator is statistically rigorous, such search may yield poor results. This is not an issue for prfoileEMM and in fact, the profileEMM classifier performed perfectly associating each protein sequence with the correct model with a significance assessment.

Experiment 4: RNA based phylogeny.

In this experiment we use the 16S rRNA data to demonstrate profileEMM in phylogeny. Using the models generated and organized in experiment 1, the phylogeny is generated using the same algorithm that was for DNA phylogeny except that the data used comes from the most diverse genus *Burkholderia* and the results are shown in figure 5[16]. The figure clearly shown the rRNA sequences that are indeed related except for questionable placement of two sequences - one from each *pseudomallei* and *mallei*. Kotamarti et al describe this to be a possible taxonomy issue as the distances were confirmed using a multiple sequence alignment [16].

Experiment 5: Comparing Viterbi/Forward algorithms with native EMM algorithms.

Finally, we would like to describe the effectiveness of native EMM based evaluation algorithms against the more standard Viterbi/Forward algorithms. For this, we added position information to EMM states to derive the emission probabilities that are typically used in standard viterbi like algorithms. Unlike the alignment paradigm where there are three possible states (Match, Insert and Delete) through which a transition could occur, in EMM there is only one type of state, i.e, a *fuzzy* match state only. It is fuzzy because the matching is not precise when clustering at model building time. This is a major advantage for two reasons: compression but also because of the fuzzy matching (clustering), minor changes such as single-nucleotide polymorphism (SNPS) or other short differences that occur will be automatically handled. It implies that a Viterbi style scoring would simply aggregate logarithmic values of emission and transition probabilities along the way as a query sequence is assessed for homology. Furthermore, the Forward and the Viterbi algorithms collapse into a single algorithm since there no multiple states that need to be checked for maximum (in case of the Viterbi) or combined (in case of the Forward algorithm). To verify how well this works, we first tested on the protein datasets introduced earlier. The algorithm predicted with 100% accuracy just as the EMM native LoD scoring algorithm.

	amino1_3_1	amino2_3_1	amino3_3_1	amino4_3_1	amino5_3_1	Glycera-di	Lupinus-lu	Chironomus	Petromyzon	Homo-sapie
amino1_3_1	0	0.0148	0.0024	0.0089	0.0043	0.0028	0.0046	0.0037	0.0032	0.0037
amino2_3_1	0.0148	0	0.0016	0.0151	0.0022	0.0044	0.0032	0.0012	0.0029	0.004
amino3_3_1	0.0024	0.0016	0	0.0021	0.0095	0.004	0.0031	0.0036	0.0037	0.0041
amino4_3_1	0.0089	0.0151	0.0021	0	0.0034	0.0055	0.009	0.0014	0.0039	0.0115
amino5_3_1	0.0043	0.0022	0.0095	0.0034	0	0.0029	0.0036	0.0015	0.0045	0.0029
Glycera-di	0.0028	0.0044	0.004	0.0055	0.0029	0	0.0096	0.0049	0.0048	0.0071
Lupinus-lu	0.0046	0.0032	0.0031	0.009	0.0036	0.0096	0	0.0034	0.0074	0.0043
Chironomus	0.0037	0.0012	0.0036	0.0014	0.0015	0.0049	0.0034	0	0.0026	0.0033
Petromyzon	0.0032	0.0029	0.0037	0.0039	0.0045	0.0048	0.0074	0.0026	0	0.0024
Homo-sapie	0.0037	0.004	0.0041	0.0115	0.0029	0.0071	0.0043	0.0033	0.0024	0

Table 5: A protein families distance matrix. The table reflects the pairwise distances among 10 proteins. The five are extracted from the globin dataset 3 and the other five from the unlabeled dataset 4. The matrix is generated using profileEMM distance metric which satisfies the formal requirements of a metric and the algorithm is described in [15].

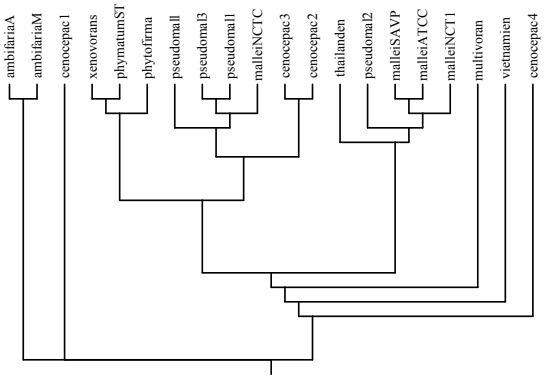


Figure 5: Phylogeny of Burkholdier: The phylogeny of Burkholdier, generated using our distance metric, is shown. The topological accuracy can be analyzed by verifying the placement of similar organisms. With the exception of a few organisms, the topology is generally correct.

Next,we tested it on slightly larger dataset of 33 members from the phylum *Euryarchaeota*. The results are summarized in the table 6. The success of only 70% for the Viterbi/Forward algorithms compared with the 100% success for the native EMM algorithm implies some missing information for the former. Perhaps, in an alignment-free context, much larger data basis is used in a single node where as in alignment, a codon is used as a single node and as such some information loss occurs if only the emission probability is considered. The additional information comes from the Log-Odds treatment of the contents of an NSV in case of EMM and hence the success in finer classification tests.

Additionally, sub-genus classification of microbial sequences is unavailable today to the best of our knowledge. Using EMM addresses this issue quite efficiently. First, a species library is created in the form EMMs. Next, classification for all strains is generated using all the available 16S rRNA copies. Ambiguous classifications where more than one species matched a strain are reported for a more involved analysis using markers other than the 16S rRNA. But the percentage of these was found to be less than 2% by the authors and the problem is well known issue due to the heterogeneity of the 16S rRNA.

4. DISCUSSION OF RESULTS

Much of the discussion is captured in the previous section itself in the form of annotated figures and tables. The first experiment established that profileEMMs are more compact than the profileHMMs and that much conservation information is readily observable in the EMM visualizations themselves. In our observation, the model building time for profile HMM is much less than that for the profile EMM, but the model preparation time, i.e., multiple sequence alignment is much larger for the former. We found the EMM to outperform HMM for the overall model build times. The experiment 2 proved that phylogenetic analysis is possible using DNA sequences with EMM. The experiment 3 showed a simple homology assessment for protein families again using the EMM. Experiment 4 showed that RNA sequences can also be analyzed for phylogeny unlike profileHMMs which require a more complex covariance model structure. In the final experiment, we used the same algorithmic constructs such as Viterbi/Forward from the alignment based profileHMM and showed that they are not adequate for alignment-free and that score based modeling is more applicable for alignment-free studies.

5. CONCLUSION

Sequence analysis has extended from statistical BLAST based query of sequence database to probabilistic profileHMM based familial modeling (Figure 1) and analysis. With models as proven and useful representations, as their library sizes too start increasing, faster and better assessment methods become necessary. In anticipation toward a much larger model database with performance issues due to increased data from the next generation sequencing, we established that profileHMM methods could be improved using profileEMM methods. We clearly showed that model building, model evaluation and phylogenetic analysis for both DNA and RNA sequences is efficiently possible more effectively with profileEMM.

A much larger exercise of recreating profileEMMs from the existing PFAM database should be possible since much of the hard work of gathering required input sequences is already well established and in place. It would be interesting to see how motif finding would work with profileEMM.

It is necessary that the profile concept is well maintained and required to support statistical significance which lacks in the profileHMM arena to the extent it is available with BLAST. By addressing the performance issues with profileHMM and alignment in general by using alignment-free methods, we provide an alternate equally faster and statistically sound alignment-free complement to the prevalence of and precedence set by BLAST.

5.1 Acknowledgments

This work was supported by the NSF Net-Centric IUCRC/ T-SYSTEM Inc. industrial memberships in the form of graduate studies sponsorship for the first author.

	VITERBI	EMM		VITERBI/EMM
Haloarcula-marismortui-ATCC-43049	$	*	Methanococcoides-burtonii-DSM-6242	*
Methanococcus-aeolicus-Nankai-3	$	*	Methanococcus-maripaludis-C5	*
Methanococcus-maripaludis-C6	$	*	Methanococcus-maripaludis-C7	*
Methanosarcina-acetivorans-C2A	$	*	Methanococcus-maripaludis-S2	*
Methanosarcina-barkeri-str--Fusaro	$	*	Methanococcus-vannielii-SB	*
Methanosarcina-mazei-Go1	$	*	Methanocorpusculum-labreanum-Z	*
Pyrococcus-furiosus-DSM-3638	$	*	Methanoculleus-marisnigri-JR1	*
Pyrococcus-horikoshii-OT3	$	*	Methanopyrus-kandleri-AV19	*
Thermoplasma-acidophilum-DSM-1728	$	*	Methanosaeta-thermophila-PT	*
Thermoplasma-volcanium-GSS1	$	*	Methanosphaera-stadtmanae-DSM-3091	*
Archaeoglobus-fulgidus-DSM-4304	*	*	Methanospirillum-hungatei-JF-1	*
Halobacterium-salinarum-R1	*	*	Methanothermobacter-thermautotrophicus-str--Delta-H	*
Halobacterium-sp--NRC-1	*	*	Natronomonas-pharaonis-DSM-2160	*
Haloquadratum-walsbyi-DSM-16790	*	*	Picrophilus-torridus-DSM-9790	*
Methanobrevibacter-smithii-ATCC-35061	*	*	Pyrococcus-abyssi-GE5	*
Methanocaldococcus-jannaschii-DSM-2661	*	*	Thermococcus-kodakarensis-KOD1	*
			Thermococcus-onnurineus-NA1	*

Table 6: Comparing algorithms (Viterbi/Forward and native EMM). This table summarizes the results of a homology assessment of the standard Viterbi/Forward algorithm against EMM's native Log Odds Scoring algorithm. EMM algorithm outperforms with 100% accuracy in prediction against the 70% success in case of the probabilistic Viterbi/Forward algorithm. The $ represents an error where as a * represents success. The high miss rate with probabilistic algorithm is perhaps due to information loss in considering wider regions as opposed to a single amino acid (codon) position in alignment paradigm. The information loss does not exist with EMM's method as the contents of a state are taken into consideration.

6. REFERENCES

[1] S. F. Altschul, T. L. Madden, A. A. Schäffer, J. Zhang, Z. Zhang, W. Miller, and D. J. Lipman. Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*, 25(17):3389–3402, Sep 1997.

[2] R. Apweiler, A. Bairoch, C. H. Wu, W. C. Barker, B. Boeckmann, S. Ferro, E. Gasteiger, H. Huang, R. Lopez, M. Magrane, M. J. Martin, D. A. Natale, C. O'Donovan, N. Redaschi, and L.-S. L. Yeh. UniProt: the Universal Protein knowledgebase. *Nucleic Acids Res*, 32(Database issue):D115–D119, Jan 2004.

[3] B. E. Blaisdell. A measure of the similarity of sets of sequences not requiring sequence alignment. *Proc Natl Acad Sci U S A*, 83(14):5155–5159, Jul 1986.

[4] J. E. Clarridge. Impact of 16S rRNA gene sequence analysis for identification of bacteria on clinical microbiology and infectious diseases. *Clin Microbiol Rev*, 17(4):840–62, table of contents, Oct 2004.

[5] M. H. Dunham, N. Ayewah, Z. Li, K. Bean, and J. Huang. *Spatiotemporal Prediction using Data Mining Tools.* Idea Group, 2005.

[6] M. H. Dunham, Y. Meng, and J. Huang. Extensible Markov model. In *Proc. Fourth IEEE International Conference on Data Mining ICDM '04*, pages 371–374, 1–4 Nov. 2004.

[7] R. Durbin. *Biological Sequence Analysis: probabilistic models of proteins and nucleic acids*. Cambridge University Press, 1998.

[8] S. R. Eddy. Profile hidden Markov models. *Bioinformatics*, 14(9):755–763, 1998.

[9] S. R. Eddy and R. Durbin. RNA sequence analysis using covariance models. *Nucleic Acids Res*, 22(11):2079–2088, Jun 1994.

[10] J. Felsenstein. Phylip (phylogeny inference package), version 3.57 c. *Seattle: University of Washington*, 1995.

[11] O. Gascuel. BIONJ: an improved version of the NJ algorithm based on a simple model of sequence data. *Mol Biol Evol*, 14(7):685–695, Jul 1997.

[12] E. Gasteiger, A. Gattiker, C. Hoogland, I. Ivanyi, R. D. Appel, and A. Bairoch. ExPASy: The proteomics server for in-depth protein knowledge and analysis. *Nucleic Acids Res*, 31(13):3784–3788, Jul 2003.

[13] C. Isaksson and M. H. Dunham. A comparative study of outlier detection. *International Conference on Machine Learning and Data Mining MLDM*, 2009.

[14] S. Karlin and S. F. Altschul. Methods for assessing the statistical significance of molecular sequence features by using general scoring schemes. *Proc Natl Acad Sci U S A*, 87(6):2264–2268, Mar 1990.

[15] R. Kotamarti, M. Hahsler, D. Raiford, and M. Dunham. Sequence Transformation to a Complex Signature Form for Consistent Phylogenetic Tree Using Extensible Markov Model. In *Proceedings of IEEE CIBCB 2010*, 2010.

[16] R. M. Kotamarti, D. W. Raiford, M. Hahsler, Y. Wang, M. McGee, and M. H. Dunham. Targeted Genomic signature profiling with Quasi-alignment statistics. *COBRA Preprint Series*, November 2009.

[17] A. Krogh, M. Brown, I. S. Mian, K. Sjölander, and D. Haussler. Hidden Markov models in computational biology. Applications to protein modeling. *J Mol Biol*, 235(5):1501–1531, Feb 1994.

[18] M. Madera and J. Gough. A comparison of profile hidden markov model procedures for remote homology detection. *Nucleic Acids Res*, 30(19):4321–4328, Oct 2002.

[19] A. G. Murzin, S. E. Brenner, T. Hubbard, and C. Chothia. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J Mol Biol*, 247(4):536–540, Apr 1995.

[20] J. D. Thompson, D. G. Higgins, and T. J. Gibson. CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucleic Acids Res*, 22(22):4673–4680, Nov 1994.

[21] S. Vinga and J. Almeida. Alignment-free sequence comparison - a review. *Bioinformatics*, 19(4):513–523, Mar 2003.

A density-based algorithm for evaluating the statistical significance of individual classification results

Loai M. Alnemer, Jianfei Wu, Omar Al-Azzam, and Anne M. Denton
Department of Computer Science and Operations Research
North Dakota State University
Fargo, ND
{loai.al-nimer,jianfei.wu,omar.al-azzam,anne.denton}@ndsu.edu

ABSTRACT

The significance of a prediction, at the level of each record, is of great interest in many application areas including bioinformatics. Traditional statistics techniques, such as discriminant analysis, only provide answers for significance when the input data follow narrow assumptions, such as being normally distributed. We present a density-based algorithm to determine prediction significance at the level of each record in data that consist of many binary attributes as well as class labels. The algorithm is based on applying a Poisson test to actual and expected neighbors of the given class label. We evaluate the approach on yeast protein domains and functions. The experimental results show that the density-based algorithm is far more representative of the baseline ensemble model than two comparison methods, including the use of the distance to the decision boundary in support-vector machines, and a conformal technique. The density-based technique also shows better scaling with respect to the number of class labels than the comparison techniques.

Keywords

Significance of classification, Density-based algorithm, Ensemble-based algorithm, Conformal prediction, Parzen-window classification

Categories and Subject Descriptors

I.5 [**Pattern Recognition**]: Design Methodology—*Classifier design and evaluation*; G.3 [**Probability and Statistics**]: Distribution functions; H.2.8 [**Database Applications**]: Data mining

1. INTRODUCTION

In classification problems, the individual predictions are only a part of the information that is possibly relevant to a user. Getting a measure of how reliable a classification result is, may be equally important. While this need was seen as a high priority in developing traditional statistical analysis techniques, such as discriminant analysis [8], modern classification algorithms typically ignore the importance of a reliability estimate. Some classification techniques provide measures that may be used as estimates of the reliability of a prediction, such as the distance to the decision boundary in support vector machines. We will show in the evaluation section that the distance to the decision boundary does not correlate well with the baseline model, which we constructed from an ensemble of classifiers.

Information on the reliability of an individual classification result can be of interest as part of a pattern mining tool. In some applications only the very reliable results are needed. One such example is functional annotation of proteins. It may be desirable to only annotate those proteins, for which the prediction is significant, since missing annotations are less problematic than incorrect ones. In other applications the classification result is used to identify an interesting set of candidates for further study since the number of objects that can be handled by an experimental process may be limited. One such application is the design of markers genes for gene mapping.

A further problem domain, in which it is important to get quantitative reliability information is classification from multiple sources. Especially in the biological sciences, it is very common that many data sources are available that may contribute to a classification problem. The same genes and proteins from the same organism are often characterized by many different types of information including, for example, protein domain information and gene expression data. Even for a single type of data such as for gene expression results, it is often advisable to keep data separate if they were collected using different platforms or experimental conditions. A classification result that can be explained reliably on the basis of just one set of experiments may be more useful than a result of some collective classification based on data from many research groups.

Much work has been done on calculating the overall significance of a classifier [17, 16, 9]. Such approaches ignore the possibility that some input data may result in a high classification accuracy for some objects but not for others. In this paper, we consider the problem of evaluating the significance of prediction for each record in the data set. Table 1 illustrates the problem of interest. Each record represents a gene, with each gene having 8 attributes and 3 class labels. Note that although the examples are designed around genes, the algorithms could be applied to any kind of record that has two sets of associated binary data: one set acting as attributes and one set contribut-

BIOKDD'10, July 25th, 2010, Washington, DC, USA.

Table 1: Toy example showing 8 genes, each of which has 8 attributes (A1-A8). Three class labels are shown (C1-C3) and significant class labels are shown in bold.

Gene	A1	A2	A3	A4	A5	A6	A7	A8	C1	C2	C3
G1	0	0	1	0	0	1	0	1	**1**	**1**	1
G2	0	0	1	0	0	1	0	1	**1**	**1**	0
G3	0	0	1	0	0	1	0	1	**1**	**1**	0
G4	0	0	1	0	0	1	0	1	**1**	0	0
G5	0	0	0	1	1	0	1	0	0	0	1
G6	0	1	0	0	1	0	0	1	0	1	1
G7	1	0	0	0	1	0	1	0	0	0	1
G8	1	0	0	0	0	0	1	0	0	0	0
.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.	.

ing the class labels. Significant predictions are shown in bold. The table illustrates that there may be class labels, such as C1, for which all results are significant, and class labels, such as C3, for which no significant predictions can be made. However, there can also be class labels for which the predictions for some genes are significant to be 1 but not for all, such as C2.

A density-based algorithm is used and the significance of the prediction result is evaluated assuming a Poisson distribution and calculating the p-value for each record. It is assumed that if the class label were independent of the distribution in attribute space, then the number of records with class label 1 within a predefined neighborhood of each point should not differ significantly from what would be expected by random chance. The value that is expected by random chance is calculated based on the total number of points in the neighborhood and the overall ratio of records with class label 1. Density-based methods are widely used in both clustering [1] and classification [4, 22] applications. Since a wide range of similarity functions is available, density-based algorithms can be flexibly adjusted to many problems of interest.

We compare the significance results with a baseline that is constructed by considering an ensemble of classifiers derived from subsampling. If the results over a classifier ensemble are available, the significance can be calculated without further assumptions. Such an approach is inefficient and its applicability to large-scale problems may be limited. However, it provides a useful baseline against which faster algorithms can be tested.

As a comparison technique, we consider the distance from the decision boundary using a support vector machine implementation, SVM^{perf}[7]. We consider the distance from the hyperplane as a confidence value and compare it with the significance value for each record. We use the correlation coefficient value to compare the degree of the relationship between the baseline algorithm and the density-based algorithm and compare it with the degree of the relationship between the baseline algorithm and the distance to the decision boundary results. The correlation coefficient value reflects the degree of the relationship.

We also compare our results with conformal prediction using a nearest neighbor algorithm as proposed in [21]. As baseline, we again use ensemble-based results. The comparison with ensemble-based results, in this case, has to be done on the basis of confidence rather than significance, since conformal prediction uses confidence. The comparisons, density-based vs. baseline and conformal prediction vs. baseline can be quantified through classification-style results, where a true positive is a prediction that is being considered as significant (or confident) by both the density-based (or conformal) algorithm and the baseline.

Probabilistic classification techniques that give a measure of the probability of the records to be 1 or 0, such as Bayesian networks, were not included in the study because they are usually considered to be too slow for data mining applications.

The algorithm is evaluated on a data set of yeast protein domains and functions. The significance of a prediction of protein function based on the existence of protein domains is calculated. The presence of a protein domain is recorded in a binary fashion, i.e. multiple occurrences are not considered separately. We show that our density-based algorithm not only shows the best agreement with the baseline method, but also the best scaling with respect to the number of class labels of the three algorithms under consideration.

The rest of this paper is organized as follows: Section 2 discusses related research in this area; Section 3 introduces the basic idea of the proposed algorithm; the comparison algorithm and the experimental results are described and discussed in Section 4 and Section 5 concludes the paper.

2. RELATED WORK

The reliability of classification results is studied comprehensively for traditional statistics techniques such as discriminant analysis. Discriminant analysis is a technique to differentiate between groups based on several variables [8], and to classify samples into those groups.

The input data for discriminant analysis are assumed to be normally distributed. This assumption is not satisfied for our data set of binary input data. Linear discriminant analysis, LDA, also imposes constraints on the covariance matrix. LDA has been shown to be inferior to principal component analysis in face recognition [11] and to perform poorly in multiclass classification on low-dimensional data sets [10]. While in quadratic discriminant analysis group-specific covariance matrices are assumed [12], the input data are also assumed to follow a normal distribution.

The distance to the hyperplane in support vector machines is often used directly or indirectly for further processing [19, 15]. Some of the works give a relationship between the distance from the hyperplane and the reliability of the label [5].

Significance of entire classifiers is an important problem in data mining research and has been studied extensively [17, 16, 9]. In this work, we focus on the significance of classification for each record in the database, while most other statistical measures are limited to determine the significance of the complete classification result. Results on the significance of classification can also be used to evaluated feature selection, as shown in work by Lee and Bottema [9]. This work also demonstrates that a high performance score, such as a high area under the ROC curve, may not guarantee that classification is significant.

Conformal prediction was introduced by Glenn Shafer and Vladimir Vovk [21] in the context of online learning. It uses the information from the past prediction to enhance the next prediction. The algorithm first computes the non-conformality

score for the record in question, and then computes a confidence. This process can also be used to test the confidence of classification results. Shafer and Vovk proposed conformal prediction using a nearest neighbor algorithm, a least-square algorithm and a support vector machine algorithm. Conformal prediction can be applied to many data mining fields [13, 20].

3. DENSITY-BASED ALGORITHM

3.1 Notation

We assume that each object is represented by a vector of attributes and a vector of class labels $V_{tot} = \{V, V_c\}$. Both attributes and class labels may either be present or absent, but are more commonly absent than present, as is the case for items in market basket research. Conceptually such attributes can be represented as vectors of binary values V = $\{a_1, a_2 \dots a_m\}$ and $V_c = \{c_1, c_2 \dots c_k\}$, where m and k are constant throughout the data set. In contrast to typical classification problems, we also assume that there can be many class labels, and there are no constraints as to how many class labels can be 1 for any one object. That means that the classification problems can, in principle, be addressed independently. For performance reasons it may, however, be advisable to store intermediate results. Data sets with a large number of potential class labels are common in many domains, and especially in bioinformatics. Section 4 provides details on an example data set with this property.

The number of 1-values for a particular object may vary substantially. Therefore, we choose cosine similarity, which includes normalization of the respective vectors, also cosine similarity is well suitable for high dimension data set. For any two vectors V_i, V_j the similarity is:

$$sim(V_i, V_j) = \cos(\bar{V_i}, \bar{V_j}) = \frac{\bar{V_i}.\bar{V_j}}{|\bar{V_i}|.|\bar{V_j}|} \quad (1)$$

There is no general agreement on what is the best threshold for the cosine similarity. Some researchers argue that it should be less than 0.2 [6], while others show that a large similarity threshold may give a better recall rate and precision [18]. For our experiment we ran the density algorithm for threshold values in the interval $(0.1 - 0.9)$ in increments of 0.1. The results show that large threshold values (above 0.6) include only few records that have almost identical attributes, while a small threshold value (under 0.4) groups proteins that don't appear to be strongly related. Overall, a threshold of 0.5 appeared to be most appropriate and was used throughout the evaluation.

3.2 Outline of the algorithm

To describe the density-based algorithm, assume that the data set contains n vectors and each vector has m attributes and k class labels. The algorithm can be summarized as follows:

Find total neighbors: For each record, count the number of neighbors using eq. (1).

Find actual neighbors: For each record and for each class label, count the number of actual neighbors of the selected class label.

Find expected neighbors: For each record and for each class label, calculate the expected number of neighbors, based on the total number of neighbors and the total fraction of records that have the class label.

Determine Significance: Calculate the p-value for observing the given number of actual and expected neighbors based on a Poisson distribution. Records with a p-value $\leq$0.05 are considered significant.

3.3 Determining neighbors

The total number of neighbors N_i with respect to record R_i are those records, for which the cosine similarity, eq. (1), is larger than a threshold value. This step only has to be done once for all class labels.

The actual number of neighbors with respect to record R_i, for which the class label of interest is 1, has to be calculated separately for each class label. The same threshold is used as for the total number of neighbors.

The expected number of neighbors is determined as follows

$$ExpectedNN(R_i) = \frac{N_i * CL}{n} \quad (2)$$

where N_i is the total number of neighbors to record R_i, CL is the number of records, for which class label equals to 1, and n is the number of all records in the data set.

3.4 Calculating significance

The null hypothesis H0 for the significance calculation is that the number of actual neighbors does not differ from the expected number of neighbors by more than what would be expected by random chance alone. The threshold p-value used in this paper is $\alpha = 0.05$. If the probability of seeing as many or more neighbors randomly is smaller than α we reject the null hypothesis. We use a Poisson test since the actual number of neighbors that are 1 is expected to be small in comparison with the total number of neighbors. The Poisson test was chosen, since it is the appropriate statistical test when the number of possible events is large, and the probability of each is small. The probability mass function for the Poisson distribution is:

$$CPoisson(k, \lambda) = \sum_{i \geq k} \frac{e^{-\lambda}\lambda^i}{i!} = 1 - \sum_{i<k} \frac{e^{-\lambda}\lambda^i}{i!} \quad (3)$$

where e is Euler's number, k is the number of actual neighbors and λ is the number of expected neighbors.

R_i is considered significant if $CPoisson(k, \lambda) \leq \alpha$.

4. ALGORITHMS

4.1 Density-based algorithm

Pseudocode for the density-based algorithm is presented in Algorithm 1. The functions $TotalNeighbors$ and $ActualNeighbors$ are implemented using cosine similarity, see eq. (1). $ExpectedNeighbors$ is defined in eq. (2), and $CPoisson$ in eq. (3).

4.2 Ensemble-based algorithm

In order to be able to compare the performance and the effectiveness of the density-based algorithm with a comparison method, we need to access to baseline significance values. In contrast to the standard classification problem, for which the labeled data set provides the correct answers, there is no sim-

Algorithm 1: Density-based algorithm

Data: $dataPoints$; /* n data points */
foreach $pt \in dataPoints$ **do**
  $TotalN = TotalNeighbors(pt, dataPoints)$;
  $ActualN = ActualNeighbors(pt, dataPoints)$;
  $ExpectedN = ExpectedNeighbors(pt, dataPoints)$;
  $pValue(pt) = CPoisson(ActualN, ExpectedN)$;
return $pValue$

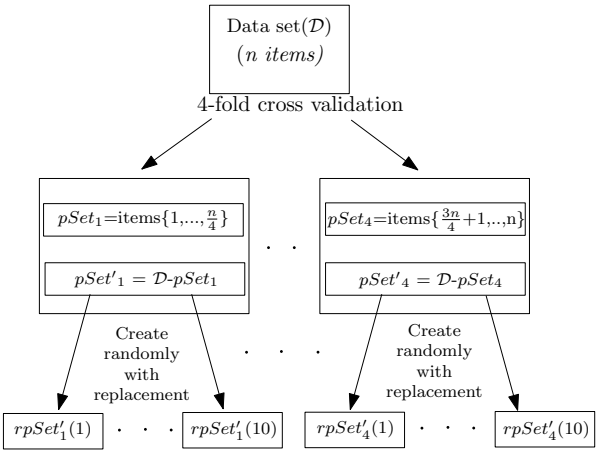


Figure 1: A schematic of the data sampling used for the ensemble-based algorithms.

ple "correct" answer available for the significance of a prediction. Instead we construct a baseline model from an ensemble of classifiers that is based on sampling the data. The distribution of results is then used to calculate significance based on a binomial distribution. The next two subsections describe the ensemble-based algorithm.

4.2.1 Data sampling

In the ensemble-based algorithm we partition the data set $\mathcal{D}$ into n_p partitions $\{pSet_1, .., pSet_{n_p}\}$. For each partition $pSet_i$ we create the complementary set $pSet'_i$ where $pSet_i \cap pSet'_i = \phi$ and $pSet_i \cup pSet'_i = \mathcal{D}$. The sets $\{pSet'_1, .., pSet'_{n_p}\}$ are used to create n_s random samples ($rpSets$) by sampling with replacement. Then for each record in the test set, we calculate its neighbors in each corresponding $rpSet$ in $rpSets$ and make a prediction using its actual and expected neighbors. This step returns 10 predictions for each record. Then we calculate the p-value for each record using the binomial distribution. In the evaluation we chose the number of partitions to be 4 ($n_p = 4$), corresponding to 4-fold cross-validation, and the number of samples for each partition to be 10 ($n_s = 10$). The sampling is illustrated in Fig. 1.

4.2.2 Ensemble-based algorithm outline

In this algorithm, density-based classification, which is sometimes also called Parzen-window classification, is used over the data sets that are created by sampling with replacement. This process is repeated over four training sets that correspond to 4-fold cross validation.

For each $pSet_k$:
Find total neighbors: For each R_i in the $pSet_k$ calculate the total number of neighbors in each of the $rpSets$ by computing the cosine similarity. This will result in n_s different values $totalN_j$ for each R_i.
Find actual neighbors: Calculate the actual number of neighbors for each record R_i based on each of the $rpSets$.
Find expected neighbors: Calculate the expected number of neighbors for each record R_i based on each of the $rpSets$.
Make predictions: For each R_i and each $rpSets$ predict the class label as follow: class label = 1 if the number of actual number of neighbors is greater than the expected number of neighbors ($actualN_j > expN_j$) and 0 otherwise, resulting in n_s predictions.
Determine significance of predictions: To determine the significance of the result, we calculate the p-value of number of predictions of 1 vs. 0 using a binomial distribution. The null hypothesis for the binomial test is that the predictions are randomly distributed with the probability of being 1 calculated as the ratio of total predictions of 1 over all predictions. To calculate the p-value use eq. (4).

$$BPMF = \sum_{i=n}^{N} \binom{N}{n} p^n (1-p)^{(N-n)} \quad (4)$$

where N is the number of samples, n the number of 1 predictions, and p is the number of genes that have class label 1 divided by the number of all genes.

Table 3 shows the p-value for 5 example genes from the data set using the density-based algorithm. Significant predictions are rendered in bold face. Table 4 shows the predicted data for five genes from applying the ensemble-based algorithm using ten random samples.

4.3 Comparison algorithms

4.3.1 Support Vector Machines

Support vector machines are widely used for classification. In this paper, we used a tool called SVM^{perf} [7], which also returns the distance from the decision boundary for each record in the dataset.

4.3.2 Conformal prediction

The conformal prediction algorithm calculates the level of confidence of each prediction using the previous predictions [21]. Conformal prediction was originally designed for an online setting where the prediction of the record depends on the records that already predicted. That means that the comparison step will be done between the current record and the records that was already visited. For comparison purposes, in this paper, we predict the new record assuming that all other records are already predicted.

The conformal prediction algorithm uses a nonconformality score that measures the degree to which the relation between the record and all previous records is unusual, and constructs a prediction region from the result. In this paper we calculate the nonconformality score between each record and all other records.

While the conformal prediction can be used with any classification techniques, [21] considers conformal prediction based on three distance measures to calculate the nonconformality

Table 2: The protein functions that are used in this paper and their abbreviations.

Function Name	Abbreviation
Hydrolase Activity	Function1
Transferase Activity	Function2
Protein Binding	Function3
Transporter Activity	Function4
Structural Molecule Activity	Function5

Table 3: Example of 5 genes and their p-values after applying the density-based algorithm. CL1 to CL5 are the 5 class labels corresponding to Function1 – Function5.

Gene	CL1	CL2	CL3	CL4	CL5
G4	**2.0E-09**	0.43	0.86	0.16	0.71
G33	0.94	**6.0E-12**	0.85	0.95	0.70
G70	0.68	0.61	**1.3E-07**	0.79	0.39
G93	0.10	0.43	0.33	0.99	0.07
G100	0.53	0.46	0.40	**5.0E-05**	0.28

score. We choose to use the cosine similarity measure to calculate this score with the goal of eliminating the effect of distance measures in our comparison. To gain a baseline model we use the same classification algorithm as is used in the conformal prediction to construct an ensemble-based algorithm.

Determining the nonconformality score: As described in [21]: Consider $\mathcal{D}$ ={r_1,...,r_{n-1}} to be the dataset that contains all records without r_n, where each r_i contains the attribute set $v_i = \{a_1, ...a_m\}$ and a class label c_i, and consider that $r_n = (v_n, c_n)$ is a record where we know the set of v_n but not the class label c_n. The nearest neighbor method will find the closest v_i to v_n and uses its c_i as a prediction to c_n. But while we have only two labels, it is difficult to tell how wrong our prediction is. For that reason we measure the nonconformality of r_n to the others by comparing v's distance to other records with the same label to its distance to old records with a different label. For example, we can set

$$A(\mathcal{D}, r_n) = \frac{\min\{1NN(r_n, r_j) : 1 \leq j \leq n \& c_n = c_j\}}{\min\{1NN(r_n, r_j) : 1 \leq j \leq n \& c_n \neq c_j\}} \quad (5)$$

where $A(\mathcal{D}, r_n)$ is the nonconformality score between r_n and all all $n-1$ records in $\mathcal{D}$.

Calculating confidence: Consider $\alpha = A(\mathcal{D}, r_n)$ then:

$$Confidence(c) = \frac{\#\{i = 1, ..., n | \alpha_i \geq \alpha_n\}}{n} \quad (6)$$

where n is the number of records in the database. Then $Confidence(c)$ is the confidence value for the current labeling c. Finally we include c in the confidence region if and only if $Confidence(c) \geq 0.5$, where 0.5 is our threshold.

5. EXPERIMENTAL EVALUATION

Our experiments were conducted on a personal computer with a 2.0 GHz CPU and 3 GB RAM under Windows Vista home edition, and the algorithm is implemented in VB.NET language.

Table 4: Resampling result for the 5 example genes from Table 3.

Gene	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10	p-value
G4	1	1	1	1	1	1	1	1	1	1	9.8E-10
G33	0	0	0	0	0	0	0	0	0	0	1
G70	0	1	0	0	0	0	0	0	0	0	0.74
G93	0	0	0	0	0	0	0	0	0	0	1
G100	0	0	1	0	0	0	0	0	1	0	0.36

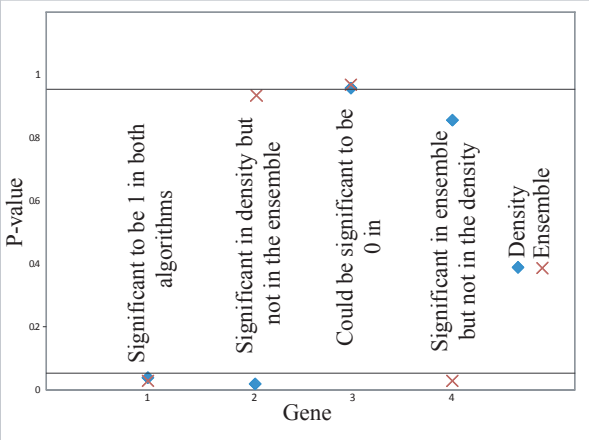


Figure 2: Example p-values for the density-based algorithm in comparison with the corresponding p-values for ensemble-based algorithm. Four example genes from our experiment are shown.

The evaluation is done on yeast data. Since yeast is a model organism, extensive functional information has been collected as well as protein sequence domain information. The domain information is collected from the Interpro database which maintains protein domain models from several sources [14]. Yeast domains can be downloaded from [3]. We consider only those protein domains that appear in at least 5 genes, leaving 791 protein domains. Only those genes are considered, which have at least one of the domains, leaving 2006 genes. For simplicity, we use a binary representation in which protein presence corresponds to a binary value of one, and absence to a value of 0. We use gene ontology terms from [2] as functional information. The go slim rolled up version is used to ensure that functional annotations have appropriate support. Experiments are reported for the most 5 common protein functions in the yeast database. The respective functions are shown in Table 2.

5.1 Quantitative evaluation on yeast data set

As a quantitative evaluation we apply the density-based algorithm and the baseline ensemble-based algorithm to each protein function in Table 2 and compare the results. For each function we calculate the contingency table of class labels of 1.

Fig. 2 shows the four possibilities of the results that can occur. Gene 1 has a p-value of less than 0.05 for both the density-based algorithm and the ensemble-based algorithm. We consider this a true positive value. Gene 2 is considered significant according to the density-based algorithm but not

Table 5: Contingency table for density-based algorithm versus ensemble-based algorithm for Function1. All genes are represented regardless of actual class labels and the confidence of the class label being 1 is compared with the confidence based on an ensemble.

		Ensemble-based	
		<0.05	>0.05
Density-based	<0.05	244	60
	>0.05	8	1694

Table 6: Contingency table for conformal prediction algorithm vs. ensemble for Function1. All genes are represented regardless of actual class labels and the confidence of the class label being 1 is compared with the confidence based on an ensemble.

		Conformal-ensemble	
		>0.5	<0.5
Conformal-1NN	>0.5	219	632
	<0.5	32	1123

according to the ensemble-based algorithm. Since we predict it to be significant, when the baseline method considers it to be insignificant, we consider it a false positive case. Gene 3 has p-values greater than 0.05 for both algorithms and is, therefore, considered to be a true negative. Finally, gene 4 has a p-value of greater than 0.05 for the density-based algorithm and a p-value of less than 0.05 for the ensemble-based algorithm, which makes it a false negative case. The actual values for this example were taken from the experiment on the yeast data set.

Fig. 3 shows the complete results for Function1 in graphical form. Only those records are shown for which the function is present (i.e., 1) for the gene to avoid overloading the figure. The top right area corresponds to true negative values and the small bottom left corner to true positive results. Results in the top left, vertically oriented area, which are only significant for the density-based algorithm are considered false positives, and those in the bottom right, horizontally oriented area are false negatives. The plot shows a large number of true positive points, indicating that the prediction significance is high overall. Although the visual comparison of true positive and negative results vs. false positive and negative results is made difficult by the differences in sizes of the areas, the results also do show that the number of true positives is higher than that of false positives and negatives combined, which supports the quality of the density-based significance calculation.

The numerical results for the total data set (Function1 being 0 or 1) are shown in Table 5 in the form of a contingency table. It shows that for almost 97% of the data the predictions of the density-based and the ensemble-based algorithm match.

The corresponding results for the confidence of the conformal-1NN algorithm versus the confidence value of conformal-ensemble algorithm are shown in Fig. 4. Again, the displayed results are limited to those records that have Function1 being 1. The full contingency table is shown in Table 6.

5.2 Effectiveness

To determine the effectiveness of the algorithm, we calcu-

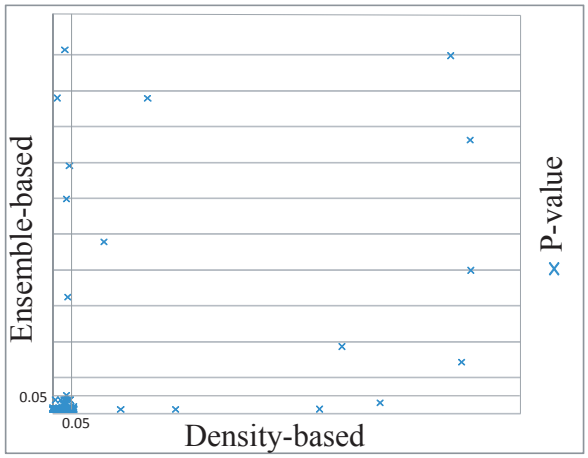


Figure 3: A plot of p-values for the density-based algorithm vs. the ensemble-based algorithm. The plot shows the result for those proteins that have Function1.

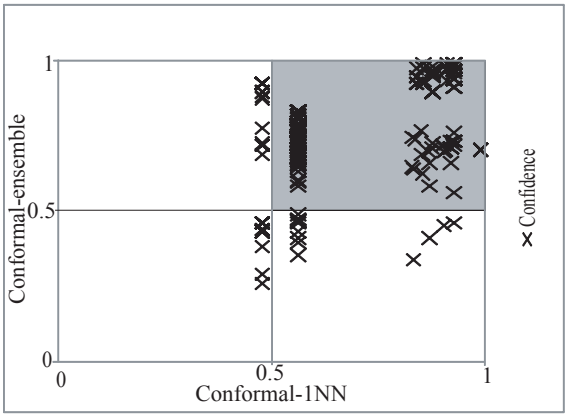


Figure 4: A plot of confidence of the conformal-1NN algorithm vs. the conformal-ensemble algorithm. The plot shows the result for those proteins that have Function1.

late the correlation between the density-based and the ensemble-based algorithms and compare it with the correlation between the SVM and the ensemble-based algorithm. We use the Spearman's rank correlation coefficient, or Spearman's rho, to calculate the correlation between the two ranked variables. The correlation coefficient is a number between -1 and 1 that tell us the direction of the association between the two variables and the magnitude of this association.

Fig. 5 shows that the correlation between the density-based algorithm and the ensemble-based algorithm is higher than that between SVM and the ensemble-based algorithm. All values are positive which means that all measures are correlated, but the correlation between the density-based algorithm and the ensemble-based algorithm is larger.

Fig. 6 shows the sensitivity of the density-based algorithm versus the ensemble-based algorithm, and the sensitivity of the conformal prediction algorithm as applied to the same data. It shows that the sensitivity with respect to the baseline algo-

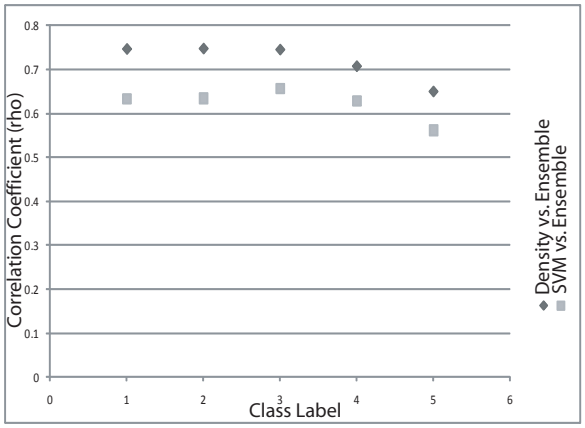


Figure 5: Comparison of the Correlation between the density-based algorithm and the ensemble-based algorithm and that between SVM and the ensemble-based algorithm.

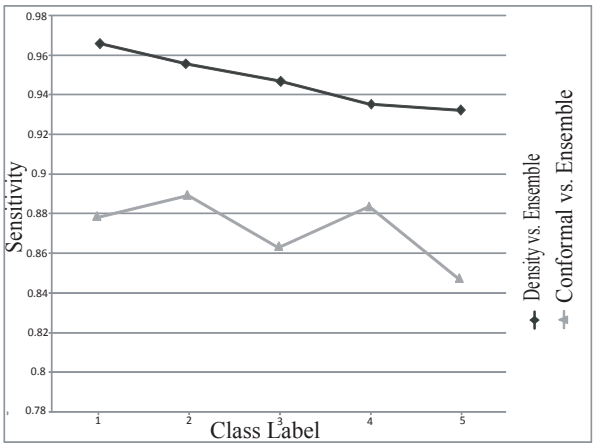


Figure 6: Sensitivity of the density-based and conformal algorithms in relationship to the corresponding ensemble-based algorithms. All five functions are considered in for both algorithms.

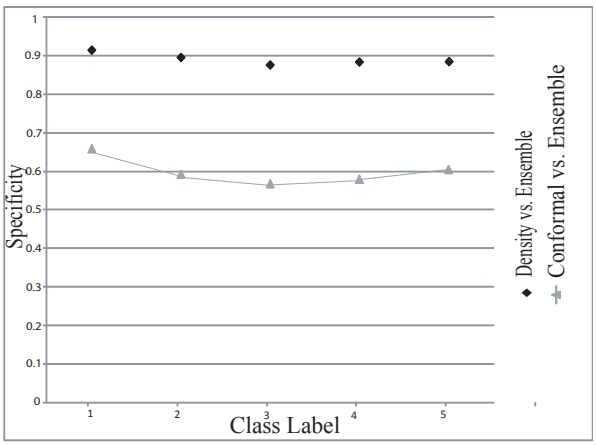


Figure 7: Specificity of the density-based and conformal algorithms in relationship to the corresponding ensemble-based algorithms. All five functions are considered in for both algorithms.

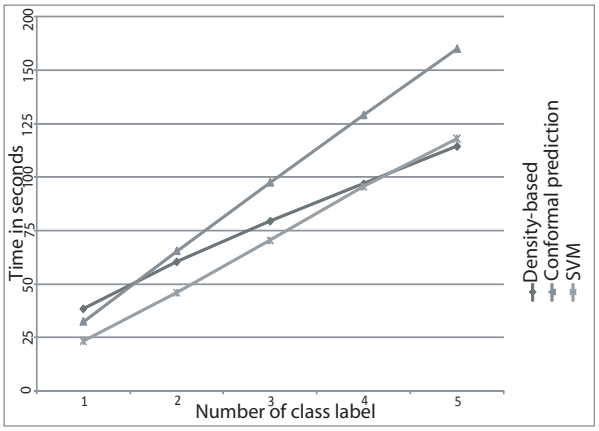


Figure 8: Execution time depending on the number of class labels under consideration.

rithm is over 90% in all function for the density-based algorithm, while it is less than 90% in all functions for the conformal prediction.

The corresponding specificity comparison is shown in Fig. 7. Again the density-based algorithm shows a much higher specificity with respect to its corresponding ensemble-based algorithm than the conformal prediction does.

5.3 Performance

Fig. 8 shows the running time of the density-based algorithm, conformal algorithm, and SVM algorithm, versus the number of class labels. It can be seen that the running time of the conformal prediction algorithm is better than that of the density algorithm when it is applies to only one function. For an increasing number of functions the density-based algorithm becomes more favorable, since the total number of neighbors does not have to be recomputed. The SVM algorithm is faster than the density algorithm when the number of class label is less than five, but for this comparison, too, the density-based algorithm becomes the more favorable choice for a larger number of class labels. The counting of the actual number of neighbors, which does have to be done for each class label, is less time consuming than the counting of the total number of neighbors since the class labels are sparse (far fewer 1 than 0 values). The conformal prediction algorithm calculates the value of the nearest neighbors between selected records and all other records, which means there is only one calculation for each record. It does, however, scan the full data set two additional times in comparison with the density-based algorithm. The ensemble-based algorithm is not included in the comparison, since it is only intended as a baseline, and cannot be expected to be competitive with any of the other algorithms, due to the large number of classifications that have to be performed.

6. CONCLUSIONS

In this paper we have presented a density-based algorithm to calculate the significance of class label predictions for each record of a database. The algorithm was designed to be applicable to current data sets in bioinformatics for which attributes are often binary and sparse. The algorithm is evaluated with respect to a baseline algorithm that is constructed from an ensemble of predictions. We compare the performance with a support vector machine algorithm, for which the distance to the decision boundary is considered as confidence measure, and with a conformal prediction algorithm. In both cases the effectiveness is substantially improved, while the efficiency is comparable, and may even be better for a large number of class labels.

The algorithm is evaluated on real protein domain (from Interpro) and function data (Gene Ontology terms). The results on the five most prevalent functional annotations are presented and it is shown that the effectiveness of our proposed algorithm is consistent across all five. The evaluation of the algorithm shows that the average accuracy of the density-based algorithm is 96%, while the average sensitivity is 93% and the average specificity is 91%. In contrast, the average accuracy of the conformal algorithm is only 88%, the average sensitivity is 87% and the average specificity is 61%. For the comparison with support vector machines, we show that the results of the density-based algorithm show a much higher correlation with the baseline algorithm than the comparison results. We also show that because of the potential for reuse of intermediate results, the density-based algorithm scales more favorably with the number of class labels than either of the comparison algorithms.

7. ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation under Grant No. IDM-0415190.

8. REFERENCES

[1] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander. Lof: Identifying density-based local outliers. In *SIGMOD Conference*, pages 93–104, 2000.

[2] G. O. Consortium. The Gene Ontology (GO) database and informatics resource. *Nucleic Acids Research*, 32(suppl_1):D258–261, 2004.

[3] S. G. Database. Interproscan results using s. cerevisiae protein sequences ftp://genome-ftp.stanford.edu/pub/yeast/sequence_similarity/domains/domains.tab.

[4] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. Wiley-Interscience Publication, 2000.

[5] M. Fauvel, J. Chanussot, and J. Benediktsson. A combined support vector machines classification based on decision fusion. In *Proc. IEEE Intl. Geoscience and Remote Sensing Symposium*, pages pp. 2494–2497, 2006.

[6] T. Hasegawa, S. Sekine, and R. Grishman. Discovering relations among named entities from large corpora. In *ACL '04: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, page 415, 2004.

[7] T. Joachims. A support vector method for multivariate performance measures. In *ICML*, pages 377–384, 2005.

[8] W. R. Klecka. *Discriminant analysis*. Wiley-Interscience Publication, 1980.

[9] G. N. Lee and M. J. Bottema. Significance of classification scores subsequent to feature selection. *Pattern Recogn. Lett.*, 27(14):1702–1709, 2006.

[10] T. Li, S. Zhu, and M. Ogihara. Using discriminant analysis for multi-class classification: an experimental investigation. *Knowl. Inf. Syst.*, 10(4):453–472, 2006.

[11] A. M. Marti'nez and A. C. Kak. Pca versus lda. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):228–233, 2001.

[12] G. J. Mclachlan. *Discriminant Analysis and Statistical Pattern Recognition (Wiley Series in Probability and Statistics)*. Wiley-Interscience, August 2004.

[13] M. Moed and E. N. Smirnov. Efficient adaboost region classification. In *MLDM '09: Proceedings of the 6th International Conference on Machine Learning and Data Mining in Pattern Recognition*, pages 123–136, Berlin, Heidelberg, 2009. Springer-Verlag.

[14] N. Mulder, R. Apweiler, and T. Attwood. New developments in the interpro database. *Nucleic Acids Research*, 35:D224–228, 2007.

[15] G. Nalbantov, J. Bioch, and P. Groenen. Classification with support hyperplanes. Econometric Institute Report EI 2006-42, Erasmus University Rotterdam, Econometric Institute, 2010.

[16] E. Nyssen. Evaluation of pattern classifiers: testing the significance of classification efficiency using an exact probability technique. *Pattern Recogn. Lett.*, 17(11):1125–1129, 1996.

[17] E. Nyssen. Evaluation of pattern classifiers — applying a monte carlo significance test to the classification efficiency. *Pattern Recogn. Lett.*, 19(1):1–6, 1998.

[18] H. Qi, J. Otterbacher, A. Winkel, and D. R. Radev. The university of michigan at trec2002: Question answering and novelty tracks. In *Current Issues in Parsing Technology*. Kluwer Academic Publishers, 2002.

[19] S. Ramaswamy, P. Tamayo, R. Rifkin, S. Mukherjee, C. H. Yeang, M. Angelo, C. Ladd, M. Reich, E. Latulippe, J. P. Mesirov, T. Poggio, W. Gerald, M. Loda, E. S. Lander, and T. R. Golub. Multiclass cancer diagnosis using tumor gene expression signatures. *Proceedings of the National Academy of Sciences of the United States of America*, 98:15149 – 54, 2001/12/18/ 2001.

[20] F.-M. Schleif, T. Villmann, M. Kostrzewa, B. Hammer, and A. Gammerman. Cancer informatics by prototype networks in mass spectrometry. *Artif. Intell. Med.*, 45(2-3):215–228, 2009.

[21] G. Shafer and V. Vovk. A tutorial on conformal prediction. *J. Mach. Learn. Res.*, 9:371–421, 2008.

[22] H. Wang, D. Bell, and I. Düntsch. A density based approach to classification. In *SAC '03: Proceedings of the 2003 ACM symposium on Applied computing*, pages 470–474, New York, NY, USA, 2003. ACM.

Opinion Mining for Biomedical Text Data: Feature Space Design and Feature Selection

Rajesh Swaminathan
Dept. of Computer Science
San Francisco State University
San Francisco, CA, USA, 94132
rajeshs@sfsu.edu

Abhishek Sharma
Dept. of Computer Science
San Francisco State University
San Francisco, CA, USA, 94132
asharma1@sfsu.edu

Hui Yang*
Dept. of Computer Science
San Francisco State University
San Francisco, CA, USA, 94132
huiyang@cs.sfsu.edu

ABSTRACT
Unstructured text (e.g., journal articles) remains as the primary means for publishing biomedical research results. To extract and integrate knowledge from such data, text mining has been routinely applied. One important task is extracting relationships between bio-entities such as foods and diseases. Most existing studies however stop short of further analyzing the extracted relationships such as the polarity and the level of certainty at which the authors reported on a given relationship. The latter is termed as the relationship strength and marked at three levels—weak, medium and strong. We have previously reported a preliminary study on this issue [22], and here we detail our studies on constructing a novel feature space towards effectively predicting the polarity and strength of a relationship. Unlike previous work, four types of polarity instead of three are considered, namely, positive, negative, neutral and no-relationship. Another contribution is that in addition to the commonly accepted lexicon-based features, we have identified a set of novel features that capture both the semantic and structural aspects of a relationship. Our intensive evaluations demonstrate that combining these new features with the lexicon-based ones can achieve the best accuracy for polarity prediction (~0.91). This however is not the case for strength prediction, where lexicon-based features alone are sufficient (~0.96).

Categories and Subject Descriptors
H.2.8 [**Database Applications**]: Data Mining; J.3 [**Life and Medical Sciences**]– *biology and genetics, health;* I.7 [**Document and Text Processing**] Document analysis

General Terms
Algorithms, Design, Verification, and Experimentation.

Keywords
Biomedical text mining, relationship polarity and strength analysis, feature space design, feature selection.

* Corresponding author.

1. INTRODUCTION
Unstructured text (e.g., journal articles) remains as the primary means for publishing biomedical research results. For instance, the biomedical database MEDLINE in the year of 2004 contained over 12.5 million articles and it is currently growing at a rate of ~500,000 new articles each year [16]. To uncover the knowledge contained in such unstructured data, a host of automated or semi-automated text mining tasks have been developed and successfully applied [21] [6]. One of the tasks gaining high importance is extracting the relationships between the biological entities within the biomedical articles [1][2][7]. For instance in [17], the sentence "Soy consumption significantly decreased breast cancer risk" (S#1) describes a relationship between soy and cancer. Many studies have been conducted to extract such relationships with an ultimate goal of building a comprehensive bio-network [1][5]. These studies however stop short of delving into a relationship. Let us take the above sentence as an example. It expresses a positive correlation between soy and cancer. Furthermore, the word "significantly" indicates that the authors are highly confident of this relationship. We term the above two semantic features of a relationship as the *relationship polarity* and *relationship strength,* respectively. In this article, we detail our studies on constructing a novel feature space towards effectively predicting the polarity and strength of a relationship at the sentence level.

Specifically we have classified the relationship polarity into four types: positive, negative, neutral and no-relationship (Table 1). The strength feature on the other hand has three values: weak, medium and strong (Table 2). Note that the "no-relationship" polarity is highly evident in biomedical articles and has rarely been explored except the work in [25], where it is termed as "no outcome". No-relationship is different from the neutral polarity: a "no-relationship" indicates that no association is found between the biological entities in consideration, whereas for a neutral polarity, the entities are associated but has with no-orientation. Please refer to Table 1 for examples.

Table 1 Relationship polarity classification.

Example Sentence	Polarity
Soy consumption decreases cancer risk	Positive
Vitamin E increased the risk of pneumonia	Negative
Tofu intake showed a significant association with the serum concentrations of genistein.	Neutral
No significant associations were found between intake of individual and carotenoids and colorectal cancer risk.	No-Relationship

Table 2 Relationship strength levels.

Example Sentence	Strength
Vitamin E may increase the risk pneumonia	Weak(1)
Vitamin E increased the risk pneumonia	Medium(2)
Vitamin E significantly increased the risk pneumonia	Strong(3)

Although there are few studies reported to analyze the polarity and strength of relationships extracted from biomedical text, the concept is not new. Such tasks are commonly referred to as opinion mining or sentiment analysis and have gained increasing attention in the past few years [3][4][18][19][20][24]. Existing work however primarily focuses on text created by day-to-day web users such as product reviews and blogs. Both supervised and unsupervised approaches have been proposed in the past [1][20]. These approaches are mostly lexicon-based. For instance, adjectives and adverbs are routinely employed to detect the polarity of a review [19] [24]. This however does not apply to formally written biomedical articles, as adjectives and adverbs are used sparingly. In addition, semantics-based structure information can play an important role in the latter case. Let us use the above sentence S#1 again to illustrate this. The reason it has a positive polarity is because "soy" reduces the risk of a disease, i.e., cancer. Should another type of entity (e.g. chemical) be in the same place of "breast cancer risk", the polarity of this relationship would be changed. It is therefore important to construct novel features to capture such information.

In this article, we propose to construct a feature space consisting of both lexicon-based and semantics-based sequential features to address the polarity and strength mining problem. We specifically consider part of speech (POS) based unigrams, bigrams, co-references and semantics-based sequential features (e.g., food → decrease → disease in S#1). These features are designed to capture the syntax and semantics of a relationship at a variety of levels: word, phrase and sentence. We then use a wrapper-based approach [15] to select the subset of features that works the best for polarity and strength prediction, respectively.

To select the set of features we adopt the Support Vector Machine (SVM) for polarity mining and Support Vector Regression (SVR) for strength analysis. Since SVM is a binary classifier two methods are considered to classify the four types of polarity, one vs. all (positive against the rest) and two vs. two ((neutral and no-relationship) against (positive and negative)). Four different kernel functions including linear, polynomial, RBF and sigmoid are considered to build the SVM and SVR. The results show that for polarity prediction, different combinations of features produce different accuracies ranging between 0.8~0.91. The best accuracy of 0.91 is produced by the two vs. two SVM when used with a combination of POS-based unigrams and semantics- based sequential features. But for strength mining any combination of features results in almost same accuracy (~0.96).

Please note that although Niu *et al.* [25] also targets to classify the polarity of medical evidences into four classes similar to this work, our work differs from their work in several important aspects: (1) we focus on binary relationships between bio-entities, whereas they focus on descriptive medical evidences; (2) we focus on designing and selecting an effective feature space for both polarity and strength analysis while they mainly rely on traditional syntactic and semantic features such as unigrams and entity type; and (3) together with strength, our scheme is better at capturing the semantic nature of a relationship. For instance, they consider the following sentence as neutral: "The first RCT found that diclofenac plusmisoprostol versus placebo for 25 weeks produced no significant difference in cognitive function or global status." Under our scheme, it will be labeled as "no-relationship" with a "weak" strength level, which is more appropriate.

Note that prior to this study, we have conducted some preliminary work and implemented a system prototype as shown in Figure 1 [22]. The proposed system includes four modules: (i) entity recognition, which extracts the following types of biological entities from the abstracts: foods, diseases, chemicals, genes, and proteins; (ii) relationship extraction, which determines and extracts the binary relationship between two entities; (iii) polarity analysis, and (iv) strength analysis, which predicts the polarity and strength respectively of an extracted relationship. A detailed description can be found in [22].

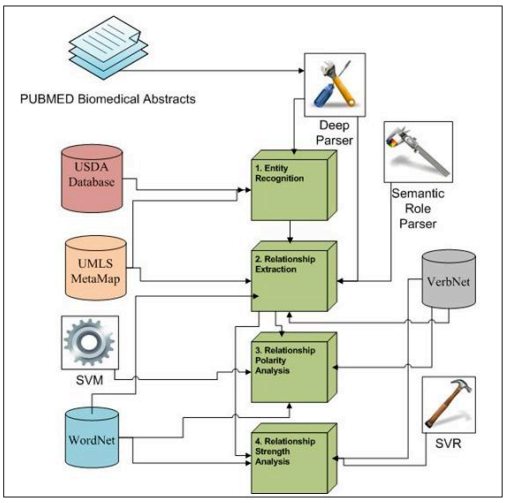


Figure 1. The system architecture proposed in [22].

Compared to the preliminary study, this work has significantly advanced the last two modules for polarity and strength analysis in the following ways: (i) we have explored a new polarity type "no-relationship" common in biomedical articles but not yet explored by the existing systems; (ii) we construct a new feature space consisting of both lexicon-based and semantics-based sequential features and provide an intensive analysis of their impact on the polarity and strength mining problem; whereas the previous work only considers unigrams based features; and (iii) we have conducted strength analysis for both neutral and no-relationship.

2. CORPUS CREATION

Due to the novel nature of the problem at hand, no corpus is currently available. We therefore have manually created a corpus, which will be used later to design a feature space for both polarity and strength analysis. We next describe the creation process.

We first downloaded 1000 abstracts from the PUBMED database [13] using a program developed by us. Each abstract contains one of the following 3 sets of keywords: (i) soy, cancer; (ii) beta-carotene, cancer; and (iii) benzyl, cancer. Using these 1000 abstracts, we aim to create a corpus that consists of a relatively large number of relationship-bearing sentences. Each sentence will be annotated to identify the following information: (i) bio-

entities with their semantic type e.g., foods and diseases; (ii) the relationship depicting phrases; and (iii) the polarity and strength of each relationship. For instance, the sentence in section I will be annotated as follows: {\food soy consumption} {\relationship 3+ significantly decreased} {\disease cancer risk}, where "3+" indicates that it is a positive (+) relationship with a strong (3) strength. Please see Tables 1-2 for the symbols we have introduced to represent each polarity and strength category.

For this corpus, we are specifically interested in the relationships between diseases and four types of bio-entities including chemicals, genes, foods, and proteins. Hence not every abstract downloaded earlier consists of such relationships. We employ a heuristic criterion to screen out the irrelevant abstracts. Specifically we retain an abstract if it contains >= 3 disease entities and >=3 occurrences of the other types of entities. We empirically evaluated this criterion on a total of 130 randomly selected abstracts and observed that it retains all the relevant abstracts (i.e., recall=100%) with a reasonable precision (86%).

In order to perform this screening task, we utilize the entity extraction module we have developed previously [22] to automatically label the following types of entities in all the 1000 abstracts: food, disease, chemical, gene and protein. A total of 200 abstracts are selected for further annotation using the aforementioned heuristic. For each of the remaining 200 abstracts, we next recruited a team of five annotators. Each individual manually identify all the relationships of interest and also annotate the polarity and strength of each relationship. Three of the five members have worked in biomedical text mining for more than a year and the other two are computer science graduates. A total of 800 relationships are identified and annotated.

Table 3 IAA according to polarity (P) and strength (S).

Parameter	IAA
Total Acceptance rate in S and P by all five members	57%
Total Acceptance rate in only S by all five members	85%
Total Acceptance rate in only P by all five members	68%
Highest Acceptance rate by any three members in both S and P	75%
Highest Acceptance rate by any three members in only S	90%
Highest Acceptance rate by any 3 members in only P	83%

Table 4 Polarity-based distribution of the corpus.

Actual Polarity	Positive (+)	Negative (-)	Neutral (=)	No-Rel. (!)
Total	447	143	117	93

Table 5 Strength-based distribution of the corpus.

Actual Strength	Weak(1)	Medium (2)	Strong(3)
Total	135	558	107

To determine the reliability of our scheme we did an inter annotator agreement (IAA) study as shown in Table 3. We observe that the strength annotations have much higher IAA rates. The reason is that the strength is largely determined by the presence or absence of certain words within a sentence (e.g., *maybe* and *significant*); annotators therefore often agree with each other. On the other hand, the IAA on polarity is much lower than that on strength. This is mainly because different annotators tend to employ their personal understanding of the contextual information in the abstract to rate the polarity of a relationship. The current scope of polarity/strength analysis is however primarily at the sentence level, i.e., without considering contextual information. This will be addressed in our future work.

To finalize the annotated corpus, we use the "majority rule" policy. For relationships that cannot be agreed upon by the majority, we held a group discussion to collaboratively decide the polarity or strength. Tables 4 and 5 present the polarity and strength distribution in our corpus of 800 relationships. Note how positive polarity and medium strength dominate this corpus.

3. FEATURE SPACE DESIGN

In this section, we describe in detail the list of syntactic and semantic features we have constructed to facilitate effective relationship polarity and strength analysis. The annotated corpus is used to evaluate these features and then select an optimal subset of features for both tasks, which will be discussed in Section 4.

Given an annotated sentence of the form "We conclude that {\chemical BITC treatment} {\relationship 2+ reduced cell survival and induced apoptosis} in {\protein caspase-3}."(S#2), we consider and construct three types of features including unigrams, bigrams, and semantic based sequential features.

3.1 POS-based Unigrams

Previous studies show that unigram-based models can achieve over 80% accuracy for analyzing movie reviews [1]. We have also demonstrated in our previous work that unigrams are helpful in polarity and strength prediction [22]. Hence we build part of speech-based unigrams into the feature space. Two different methods are used to construct the unigrams:

a) We construct the unigrams using only the annotated relationship-depicting phrase, i.e., only the phrase "*reduced cell survival and induced apoptosis"* in the above sentence.

b) In this method we expand our unigram construction to include the neighborhood of the relationship-depicting phrase (RDP). Specifically this neighborhood centers at the RDP and extends on both sides, spanning from the left boundary of the entity immediately preceding the RDP to the right boundary of the entity immediately succeeding the RDP. So for the above sentence, the entire phrase "BITC treatment reduced cell survival and induced apoptosis in caspase-3" is used in constructing the unigrams.

To obtain unigram features, for both of the above methods we first remove all the non-content words (e.g. the, and). We then do unigram normalization such that different grammatical variations of the same root word (e.g., increase, increasing, and increases) will be treated as a single unigram. We next augment our list by obtaining the synonyms of each unigram from WordNet [10]. We then identify all the verbs in our unigram list, for each verb we use VerbNet [9] to identify its semantic class and add all the verbs in the class to the unigram list as well. Note that the unigrams are organized into equivalence classes, each corresponding to a set of synonyms of a given word.

We use two sources to determine the POS tag of a given unigram:

1) Penn Tree Bank Parser [11]: We use it to first generate a parse tree for a relationship-bearing sentence and then extract the POS tag for each individual unigram using the parse tree.

2) WordNet [10]: The parse tree rendered by the above parser is probabilistic by nature. As a result, sometimes, the POS tag of a unigram can be either missing or incorrect. In such cases we use the POS from WordNet as the default POS value.

3.2 Bigrams

Bigrams and trigrams have been shown capable of yielding better polarity classification accuracy for product reviews under certain settings [18]. Moreover we have seen that bigram features are common in our dataset and thus hypothesize they might have positive effect on the final strength prediction. For instance, the bigram "significantly decreased" in the sentence "{\protein Soy consumption} {\relationship 3+ significantly decreased} {\disease cancer risk}" (S#3) can be used to determine the correct strength as 3 (Strong). For the construction of bigram features, we use a public dataset consisting of bigrams that commonly appear in PUBMED [8]. We use all the bigrams that have a term frequency and document frequency of at least 10000 in the dataset.

3.3 Semantics-based sequential features

We have observed that semantic types can influence the polarity of a relationship. For instance, let us compare the following sentence with S#3 in the above paragraph, "{/chemical Phytoestrogens} are {\relationship 2= responsible for decreased} {\protein CYP24 expression}" (S#4). Both sentences consist of the word "decreased". But S#3 has a positive polarity because the relationship is between a protein and a disease, whereas S#4 is neutral because the relationship is between a chemical and a protein and there is no information available at the sentence level to indicate otherwise. To determine at what level the semantics based sequential features can affect the overall performance of our system we construct these features at three different levels: entity, phrase, and sentence.

3.3.1 Unary Sequential Structures (USS)

Given a relationship-depicting phrase (RDP), the unary sequential features capture the semantics of the entities immediately proceeding and succeeding RDP. Let STl and STr be the semantic type of the left and right entities respectively. The possible values of STl and STr in this work include food, protein, disease, gene, chemical and relationship. "Relationship" is considered as one semantic type because two relationship-depicting phrases can be juxtaposed in the same sentence. In our implementation, we represent this feature by 12 binary dimensions in the form is_left (st, RDP) and is_right(st, RDP), where RDP is the relationship-depicting phrase under study , st represents one of the six semantic types listed above, is_left( ) and is_right( ) are two predicates that return 1 if a given semantic type precedes or succeeds the RDP. For instance the sentence (S#4) in the previous paragraph has a chemical entity to the left of the RDP, and a protein to the right. According to the above representation, the is_left (chemical, RDP) and is_right (protein, RDP) will be set to 1 and the remaining 10 dimensions will be set to 0. We term such sentences as "unary sequential" because they are constructed on the basis of individual entities around a RDP.

3.3.2 Binary/Ternary Sequential Structures (BTSS)

For a given RDP, the previous feature requires 2 separate dimensions to capture the semantic neighborhood. This might pose an issue later when being used to build a SVM classifier because there is no guarantee that these two dimensions will be considered by the SVM as two co-related dimensions. To overcome this limitation, we propose to construct binary and ternary sequential structure based features. Let E represent a bio-entity including relationship and R represent an RDP. These features capture the semantic neighborhood of R in the form of E-R-E (ternary), E-R or R-E (binary). In other words, only the immediately preceding or succeeding entity is included in the neighborhood. The E-R or R-E case exists due to either flaws in the entity recognition algorithm or complex sentence structure. We then examine the semantic types of the involved entities, which can be one of the following: foods, diseases, chemicals, genes and proteins. A total of 35 binary dimensions are created to capture the 35 possible sequential features, for instance, protein-R-disease, R-disease, and food-R. Again let us use the sentence S#4 as an example. The sentence exhibits a chemical-R-protein structure and only one of the 35 dimensions will be set to 1. Hence, the correlation structure between the chemical and protein entities are preserved as compared to the unary sequential structure based features.

3.3.3 K-array Sequential Structure (KSS)

K-array Sequential structures expand the semantic sequential structures to the sentence level. Rather than just focusing on the RDP-based neighborhood within the sentence, we take all the biological entities including the relationship occurring within an entire sentence into consideration. For example the sentence (S#5) "{\chemical Genistein}, a natural {\chemical isoflavonoid} found in {\food soy products}, {\relationship 3+ has been shown to inhibit cell growth and induce apoptosis} in a wide variety of cell lines" exhibits the following K-array sequential structure "chemical-chemical-food-relationship".

3.3.4 C.4 Co-references

We also take co-references into account when constructing the semantics-based sequential features. The co-reference module has been explained in detail in [22]. Consider the sentence (S#6) "We also found that it {\relationship 1+ may decrease} {\disease cancer} occurrences". Before constructing the semantics based sequential features of this sentence, we locate the entity to which "it" refers and retrieve its semantic type.

4. Feature Selection

As described in the previous section, we have constructed a feature space consisting of both lexicon-based and semantics-based sequential features. It is important to learn to what extent such features can be used to accurately predict the polarity and strength of relationships. We treat this as a feature selection problem and adopt the wrapper-based method for this purpose [15]. Specifically, we build multi-stage Support Vector Machine (SVM) models to classify the four types of relationship polarity: positive, negative, neutral, and no-relationship (Table 1); whereas Support Vector Regression (SVR) models are employed to predict the relationship strength at three levels: weak (1), medium (2), and strong (3) (Table 2). To train such models, we utilize the annotated corpus as described in Section 2. We split the corpus into training and testing components. We then build SVMs and SVRs using different subsets of features and consequently test their performance over the testing component. Following this strategy, we determine the subset of features that can deliver an optimal performance for predicting the polarity and strength of a relationship. We next describe the different SVM and SVR models we have built to select an optimal feature set.

4.1 Polarity Analysis

Since SVM is a binary classifier, we use two different methods to build a multi-stage classifier to handle the four polarity classes.

4.1.1 A.1 One vs. All (1 vs. A)

In this method, we first build a SVM to separate positive relationships from negative, neutral and no-relationship. We then build a SVM to separate negative from neutral and no-relationship. Finally we separate neutral from no-relationship. Such an ordering is chosen on the basis of both manual observations and empirical evaluation.

4.1.2 A.2 Two vs. Two (2 vs. 2)

Here we build a SVM to separate neutral and no-relationships first from positive and negative ones. We then build two other SVMs to separate between neutral and no-relationship and between positive and negative relationships, respectively. The above combination strategy is based on analyses of our dataset, which shows that positive and negative relationships often exhibit similar characteristics when compared against the other two types.

4.2 Strength Analysis

For strength analysis we build the SVRs using the entire training set without categorizing the existing records according to polarity. We also tried a variation of this approach by building individual SVRs based on polarity but found that polarity has no effect on the strength analysis.

4.3 Kernel Selection

We evaluate four kernel functions when building a SVM or a SVR, including linear, sigmoid, polynomial and radial-bias function (RBF). We explore a variety of kernel combinations for building the multi-stage classifiers for polarity analysis as reported in the next section.

5. Results

In this section, we report the main results to demonstrate that the proposed feature space can effectively predict the polarity and strength of the relationships extracted from biomedical literature. These results also indicate that not every feature contributes equally to the two problems under study. The annotated corpus described in section 2 is used for our evaluation studies. (See Tables 4-5 for the corpus distribution according to polarity and strength.) We perform 10-fold cross validation throughout our evaluation. Classification accuracy is primarirly used to measure the performance of each SVM, whereas prediction accuracy is used for each SVR. We use the SVMLight package by [23] for our experiments.

5.1 Polarity Analysis Results

Table 6 lists the model accuracy for polarity analysis based on various feature combinations and kernel functions. To understand this table, let us take an example of the row with feature combination (2+5) for the One vs. All method. (2+5) represents the feature combination of POS-based unigrams with WordNet correction and unary semantics-based sequential structures. The column L1 lists the best kernel function used to separate the positive polarity from the rest, which is the polynomial kernel for the feature set (2+5). The column L2 gives the best kernel function used to separate negative from (neutral and no-relationship). The same level2 kernel is used to separate neutral from no-relationship. In the case of (2+5), the RBF kernel delivers the best result. The columns +, -, = and ! list the average accuracy for each of the polarities after 10-fold cross validation using the kernel functions under columns L1 and L2. The overall accuracy calculates the accuracy of a given model over all four polarity classes using 10-fold cross validation. This is listed under the column OA. Finally, the column SE indicates the standard error of the overall accuracy. The highlighted columns represent the best overall accuracy obtained after 10-fold cross validations including both one vs. all and two vs. two methods.

Table 6 Polarity classification results using the 2 SVM schemas and different feature sets. Column notations: L1--level 1 of the SVM, L2--level 2 of the SVM, L--linear kernel, P--polynomial kernel), R--RBF kernel, OA--overall accuracy, + (Positive), - (Negative), = (Neutral), ! (No-relationship). Feature notations: 1--Penn Treebank based Unigram), 2--unigrams with WordNet based POS correction, 3--Binary semantics-based features), 4--K-ary semantics-based features, 5--unary semantics based features, and 6--bigrams. The top three models are highlighted. Note that the standard error is that of the overall accuracy.

Feature Set	SVM: One vs. All Schema							
	Kernel		Average Accuracy					StdErr
	L1	L2	+	-	=	!	OA	
1	R	R	0.88	0.9	0.87	0.88	0.87	0.0289
1+3	L	P	0.83	0.9	0.73	1	0.84	0.0312
1+4	P	P	0.85	0.9	0.67	1	0.84	0.0303
1+5	L	L	0.78	0.8	0.73	1	0.8	0.0339
1+6	R	R	0.85	0.9	0.73	1	0.85	0.0145
2	R	R	0.87	0.8	0.87	0.88	0.86	0.027
2+3	L	R	0.8	0.9	0.93	1	0.86	0.0245
2+4	P	P	0.87	0.9	0.73	1	0.86	0.014
2+5	P	R	0.88	0.9	0.87	1	0.89	0.0222
2+6	R	R	0.85	0.8	0.87	1	0.86	0.031
2+3+6	R	R	0.8	0.7	0.93	1	0.84	0.0204
2+4+6	P	P	0.9	0.7	0.73	1	0.85	0.0177
2+5+6	L	R	0.83	0.7	0.8	1	0.82	0.0235
Feature Set	**SVM: Two vs. Two Schema**							
	Kernel		**Average Accuracy**					**StdErr**
	L1	L2	+	-	=	!	OA	
1	L	P	0.95	0.9	0.6	1	0.88	0.0189
1+3	R	P	0.98	0.9	0.4	1	0.85	0.0261
1+4	L	P	0.95	0.9	0.73	1	0.91	0.0234
1+5	R	P	0.98	0.9	0.4	1	0.85	0.0108
1+6	R	P	0.92	0.8	0.6	1	0.85	0.014
2	R	R	0.98	0.9	0.53	1	0.88	0.0239
2+3	L	P	0.95	0.8	0.4	1	0.82	0.0144
2+4	L	P	0.95	0.9	0.73	1	0.91	0.0118
2+5	R	P	0.97	0.9	0.47	1	0.86	0.0262
2+6	R	P	0.97	0.7	0.53	1	0.85	0.0189
2+3+6	R	L	0.93	0.9	0.53	1	0.85	0.0203
2+4+6	L	L	0.9	0.9	0.73	1	0.88	0.017
2+5+6	L	P	0.97	0.6	0.4	1	0.81	0.0139

From Table 6, one can observe that the positive polarity constantly has high accuracy for both methods as compared to the other polarities. One main reason we believe is that the annotated corpus has a large number of positive examples as shown in Table 4. We also observe that the "no-relationship" has a constantly high accuracy and is not influenced by other feature combinations. The reason behind this is that the unigrams found in the relationships that have "no relationship" polarity often contain unique negation terms such as "no" and 'not". Therefore unigrams alone are often sufficient. The accuracy of neutral relationships is low because neutral and positive relationships tend to contain

identical unigrams and exhibit similar semantics-based sequential structures. Hence SVM is not able to differentiate between positive and neutral relationships. To give an example the below sentence has features which are the same as a positive relationship but is labeled as neutral due to lack of contextual information. {\chemical P-ASA} {\relationship 2= increased} intracellular levels of {\chemical reactive oxygen species}. Finally we can also observe that the (1 vs. All) SVM schema behaves differently than the (2 vs. 2). The highest overall accuracy of (1 vs. all) is 0.89, generated by a feature combination of unigrams and unary semantics-based sequential features. We cannot however find any particular combination of kernels that constantly produce a high accuracy. In contrast, in the (2 vs. 2) schema, the linear and polynomial kernel combination tends to produce good results. In addition, (2 vs. 2) in general outperforms (1 vs. all) regardless of the feature set under use. We hence focus on the (2 vs. 2) scheme in the following discussion.

For the (2 vs. 2) SVM schema, one can observe from Table 6 that the highest overall accuracy is 0.91 and generated by a feature combination of POS-based unigrams and K-array semantics-based sequential features, regardless of whether WordNet is used to correct the POS of a unigram or not. In terms of the kernel functions, a linear kernel is used at the first level of the SVM model and a polynomial kernel at the second level. In addition, we notice that the inclusion of bigrams (the last four rows in Table 6) does not improve accuracy but rather reduce the accuracy. This agrees with results reported in previous studies.

Tables 7 and 8 present the confusion matrices resulted from the two best feature sets using the (2 vs. 2) SVM schema. These two tables elaborate the two optimal feature sets as highlighted in Table 6. Results are obtained by using a linear and a polynomial kernel at the two levels respectively and averaged over 10-fold cross validation. Notice that in certain cases, the model cannot ditinguish between positive and neutral polarities due to the reasons discussed above; while negative and no-relationship can often be correctly identified.

Table 7. Confusion matrix from using the first optimal set of features: Penn Treebank-based Unigrams and K-ary semantics-based sequential structures. This elaborates the second highlighted row in Table 6.

Actual / Predicted	Positive	Negative	Neutral	No-Rel.
Positive	39	1	4	0
Negative	0	9	0	0
Neutral	2	0	11	0
No-Rel	0	0	0	8

Table 8. Confusion matrix from using another optimal set of features: (Penn Treebank with WordNet correction)-based Unigrams and K-ary semantics-based sequential structures. This elaborates the third highlighted row in Table 6.

Actual / Predicted	Positive	Negative	Neutral	No-Rel.
Positive	40	1	3	0
Negative	0	9	0	0
Neutral	3	0	12	0
No-Rel	0	0	0	8

Regarding whether including co-references can improve the performance, contrary to our belief, using co-references actually decreases the accuracy levels as shown in Table 9. The main reason is that it might just add redundant or even incorrect information for creating semantic structures due to limitations in the current co-reference identification module. Table 9 shows that the overall accuracy drops when including co-references, regardless of the feature set being used to construct the (2 vs. 2) SVM classifier .

Table 9. The impact of co-references on the overall accuracy using the (2 vs. 2) SVM schema.

Features	POS-based Unigram +USS	POS-based Unigram +BTSS	POS-based Unigram +KSS	POS (Correction)-based Unigram +USS	POS (Correction)-based Unigram +BTSS	POS (Correction)-based Unigram +KSS
Co-reference	0.8	0.79	0.87	0.84	0.8	0.89
No Co-reference	0.85	0.85	0.91	0.86	0.82	0.91

Another seemingly counter-intuitive result is that constructing unigram features from neighborhood of a relationship-depicting phrase (RDP) actually degrades the performance as against from the RDP only. The main reason being that more than 50% of the unigrams formed using the neighborhood have a term and document frequency of 1 or 2. This in turn has introduced irrelevant features to the model. Table 10 shows that the overall accuracy reduces to 0.85 from 0.88 for the Two vs. Two Method when using the entire RDP neighborhood to form unigrams.

Table 10. The impact of unigrams constructed from two different neighborhoods on the overall accuracy.

Features	Relationship based Unigram	E-R-E/R-E/E-R boundary based Unigram
POS without correction	0.88	0.84
POS with correction	0.88	0.82

Finally, we have also compared the effect of the no-relationship polarity by comparing the overall accuracy of (i) combining no-relationship and neutral polarities into one category; and (ii) separating them into two classes. We observe that the overall accuracy of the former ranges between 0.71~0.81 as compared to 0.8~0.91 in the latter when they are separated. This demonstrates the necessity of introducing the "no-relationship" as its own class.

5.2 Strength Analysis Results

We have also built various SVR models using different feature combinations fro strength analysis. The average accuracy from 10-fold cross validation is shown in Table 11. Note that all the results are generated using a linear kernel based SVR.

From Table 11 we observe that all the feature combinations deliver approximately similar high-quality results. In other words, the addition of bigrams and semantics-based structural features does not improve the overall performance. This indicates that using unigrams alone is sufficient for the strength prediction task.

The main reason behind this phenomenon is that unlike polarity of a relationship, the strength of a relationship is often directly associated with the specific words used in the relationship-depicting phrase. For instance, the sentence "Soy consumption significantly reduces the risk of cancer." has a strong strength. The word "significantly" carries the most weight for the SVR model to make the correct prediction. This also explains why the addition of other semantics based features does not help in general. Another observation we have made is that the linear kernel constantly outperforms the other kernels. Finally, we notice that the medium strength achieves the highest accuracy as against

the other two strengths. This is highly likely a result of the biased training dataset, which contains a large number of medium strength relationships as shown in Table 5.

Table 11. Strength prediction results using different feature sets. Feature notations: 1--Penn Treebank based Unigram), 2--unigrams with WordNet based POS correction, 3--Binary semantics-based features), 4--K-ary semantics-based features, 5--unary semantics based features, and 6--bigrams. A linear kernel is used to general all the results. The standard error is that of the overall accuracy.

	Highest Average Accuracy				
Feature Set	Accuracy (Weak)	Accuracy (Medium)	Accuracy (Strong)	Overall Accuracy	Standard Error
1	0.83	0.98	1	0.96	0.0063
1+3	0.83	0.98	1	0.96	0.0063
1+4	0.83	0.98	1	0.96	0.0062
1+5	0.83	0.98	1	0.96	0.0062
1+6	0.83	0.98	0.89	0.95	0.0062
2	0.83	0.98	1	0.96	0.0061
2+3	0.83	0.98	1	0.96	0.0061
2+4	0.83	0.98	1	0.96	0.0061
2+5	0.83	0.98	1	0.96	0.0062
2+6	0.92	0.98	0.89	0.96	0.0062
2+6+3	0.92	0.98	0.89	0.96	0.0062
2+6+4	0.83	0.98	0.89	0.95	0.0065
2+6+5	0.83	0.98	0.89	0.95	0.0065

6. CONCLUSIONS AND ONGOING WORK

In this paper, we describe a novel feature space designed to effectively classify the polarity and strength of relationships extracted from biomedical abstracts. In addition to the conventional syntactic features such as unigrams and bigrams, we have also explored and constructed semantics-based sequential features. These features are constructed at three different levels: entity, phrase, and sentence. A wrapper-based method is then used to select the optimal feature sets for both polarity and strength prediction. Specifically, a multi-stage SVM classifier and an SVR predictor are built for polarity and strength prediction, respectively. Two different schemas, namely, (1 vs. all) and (2 vs. 2), are employed to build the multi-stage SVM. Finally, three different kernel functions are considered at different stage of this SVM classifier.

Our intensive evaluations have shown that for polarity prediction, the (2 vs. 2) schema in general works better than the (1 vs. all). It produces the highest polarity accuracy of 0.91 when both unigrams and semantics-based sequential structures (KSS) are used, with a standard error ranging between 0.01~0.02. On the other hand, we find that for strength prediction, unigrams solely can produce satisfying results. We obtain a high accuracy of 0.96, with the standard error ranging between 0.61%~0.63% for the strength analysis.

We are currently expanding our annotated corpus to facilitate further validation of the findings reported in this work. We are also integrating this module with other modules as shown in Figure 1 towards building a quantitative food-disease-gene network. Finally, we are creating an interactive user interface to visually present this network.

7. ACKNOWLEDGMENTS

This work is partially supported by the Mini Grant from the Center for Computing for Life Sciences (CCLS) at San Francisco State University.

We would also like to extend our sincere thanks to Yan Dong, Jason D'Silva, and Vilas Ketkar for helping annotate the corpus and providing constructive suggestions and comments during this study.

8. REFERENCES

[1] A. M. Cohen, "Using symbolic network logical analysis as a knowledge extraction method on MEDLINE abstracts", BMC Bioinformatics 2005 (in press).

[2] A Skusa and A Rüegg and J. Köhler. Extraction of biological interaction networks from scientific literature. Briefings in Bioinformatics, (6)3:263--276, 2005.

[3] B. Pang and L. Lee, "Opinion Mining and Sentiment Analysis", Now Publishers Inc, July 2008.

[4] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up? Sentiment classification using machine learning techniques," in Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pp. 79-86,2002.

[5] C. Friedman, P. Kra, H. Yu et al., "GENIES: A natural-language processing system for the extraction of molecular pathways from journal articles", Bioinformatics (2001), Vol. 17, Suppl. 1, pp. S74-82.

[6] C. W. Gay, M. Kayaalp, and A. R. Aronson. Semi-automatic indexing of full text biomedical articles. In AMIA Annu Symp Proc, pages 271{275, Lister Hill National Center for Biomedical Communications, National Library of Medicine, Bethesda, MD 20894, USA., 2005.

[7] DR Swanson, Fish oil, Raynaud's syndrome, and undiscovered public knowledge, Perspect. Bio. Med, v30, pp. 7-18, 1986

[8] http://archive.ics.uci.edu/ml/

[9] http://verbs.colorado.edu/~mpalmer/projects/

[10] http://wordnet.princeton.edu/

[11] http://www.cis.upenn.edu/~treebank/

[12] http://www.nal.usda.gov/fnic/foodcomp/Data/

[13] http://www.ncbi.nlm.nih.gov/pubmed/

[14] http://www.nlm.nih.gov/research/umls/

[15] I. Guyon and A. Elisseeff, "An Introduction to Variable and Feature Selection", Journal of Machine Learning Research , Vol. 3 (2003) , S. 1157-1182 .

[16] J. A. Mitchell, A. R.Aronson, J. G.Mork, L. C. Folk , S. M. Humphrey , J. M. Ward, "Gene indexing: Characterization and analysis of NLM's GeneRIFs", Proceedings of the AMIA Symposium 2003, 8th-12th November, Washington, DC, pp. 460-464.

[17] J. M. Messina, V. Persky, D. R. Kenneth and et al., "Soy intake and cancer risk: A review of the in vitro and in vivo data", Nutrition and Cancer, Volume 21, Issue 2 1994 , pages 113 - 131

[18] K. Dave, S. Lawrence, and D. M. Pennock, "Mining the peanut gallery: Opinion extraction and semantic classification of product reviews," in Proceedings of WWW, pp. 519-528, 2003.

[19] P. Melville, W. Gryc and R. D. Lawrence, "Sentiment analysis of blogs by combining lexical knowledge with text classification", KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data miningNew York, NY, USA: ACM (2009) , p. 1275--1284.

[20] P. Turney, "Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews," in Proceedings of the Association for Computational Linguistics (ACL), pp. 417-424, 2002.

[21] R. Feldman, Y. Regev, E. Hurvitz, and M. Finkelstein-Landau. Mining the biomedical literature using semantic analysis and natural language processing techniques. Drug Discovery Today: BIOSILICO, 1(2), May 2003.

[22] R. Swaminathan, A. Sharma, H. Yang and V. Ketkar, "On building a quantative food-disease-gene network", the 2nd International Conference on Bioinformatics and Computational Biology (BICoB 2010)

[23] T. Joachims, Text Categorization with Support Vector Machines: Learning with Many Relevant Features. Proceedings of the European Conference on Machine Learning, Springer, 1998.

[24] X. Ding, B. Liu, P. S. Yu, "A holistic lexicon-based approach to opinion mining", WSDM '08: Proceedings of the international conference on Web search and web data mining (2008), pp. 231-240.

[25] Y. Niu, X. Zhu, J. Li, and G. Hirst, "Analysis of polarity information in medical text", in Proceedings of the American Medical Informatics Association 2005 Annual Symposium, 2005.

Discovering Coherent Value Bicliques In Genetic Interaction Data

Gowtham Atluri
Dept of Comp Sc and Engg
Univ of Minnesota, Twin Cities
Minneapolis, MN USA
gowtham@cs.umn.edu

Jeremy Bellay
Dept of Comp Sc and Engg
Univ of Minnesota, Twin Cities
Minneapolis, MN USA
bellay@cs.umn.edu

Gaurav Pandey
Dept of Comp Sc and Engg
Univ of Minnesota, Twin Cities
Minneapolis, MN USA
gaurav@cs.umn.edu

Chad Myers
Dept of Comp Sc and Engg
Univ of Minnesota, Twin Cities
Minneapolis, MN USA
cmyers@cs.umn.edu

Vipin Kumar
Dept of Comp Sc and Engg
Univ of Minnesota, Twin Cities
Minneapolis, MN USA
kumar@cs.umn.edu

ABSTRACT

Genetic Interaction (GI) data provides a means for exploring the structure and function of pathways in a cell. Coherent value bicliques (submatrices) in GI data represents functionally similar gene modules or protein complexes. However, no systematic approach has been proposed for exhaustively enumerating all coherent value submatrices in such data sets, which is the problem addressed in this paper. Using a monotonic range measure to capture the coherence of values in a submatrix of an input data matrix, we propose a two-step Apriori-based algorithm for discovering all nearly constant value submatrices, referred to as Range Constrained Blocks. By systematic evaluation on an extensive genetic interaction data set, we show that the coherent value submatrices represent groups of genes that are functionally related than the submatrices with diverse values. We also show that our approach can exhaustively find all the submatrices with a range less than a given threshold, while the other competing approaches can not find all such submatrices.

1. INTRODUCTION

Genetic Interaction (GI) data provides a means for exploring the structure and function of pathways in a cell [18]. The development of technologies like Synthetic Genetic Array (SGA) and Epistatic MiniArray (E-MAP), have enabled large-scale measurement of quantitative interactions in *S. Cerevisiae* [20]. These technologies measure the interaction between two genes in terms of the fitness of a cell when a pair of genes are knocked out relative to the expected fitness when there is no interaction between the pair of genes. Specifically, two genes A and B are said to interact genetically if the fitness of a large set of yeast cells (colony) after the deletion of both genes (say F_{AB}) differs from the expected fitness if the effects of A and B were independent, i.e., the product of the fitnesses after the deletion of A (say F_A) and B (say F_B) individually [20]. Thus two genes interact if $\epsilon \neq 0$ in the following equation.

$$\epsilon = F_{AB} - F_A F_B \quad (1)$$

The magnitude of this score, i.e., $|\epsilon|$ represents the strength of the genetic interaction between A and B. In addition, if $\epsilon > 0$, the interaction is called a "positive" or "alleviating" interaction, and $\epsilon < 0$ denotes a "negative" or "aggravating" interaction. A GI interaction data set can be represented as an adjacency matrix G, where the value of each element G_{ij} is the interaction score between the query gene g_i (row) and the array gene g_j (column), calculated using Equation 1.

Previous studies on analyzing genetic interaction networks has noted striking structure present in these networks. For example, [13, 21] have noted the presence of nearly complete bipartite subgraphs involving similar type of interactions. The two sets of genes in each of the bipartite subgraphs typically represent pairs of functionally complementary pathways or protein complexes. Previous efforts to discover these bipartite subgraphs in GI data are limited to finding bipartite subgraphs with interactions having same sign [13, 21] i.e., they look for bicliques such that all interactions are positive (or all interactions are negative) without being concerned about the variation in the magnitude of the interactions. It has been observed that bicliques with coherent (i.e., similar values) positive interaction scores represent protein complexes or modules of genes involved in similar biological functions [3, 18]. In this paper we address the problem of discovering such bicliques i.e. submatrices with coherent values in a GI data matrix.

This problem of discovering a submatrix with coherent values is very similar in nature to the biclustering problem ([15]) that is addressed in the domain of micorarray data analysis. In biclustering, the goal is to find a subset of the gene (rows) constituting a gene expression data set that have coherent values across a subset of the conditions (columns). Several algorithms have been proposed in the literature for finding such biclusters. These algorithms vary in their definition of "coherence", and thus focus on different types of

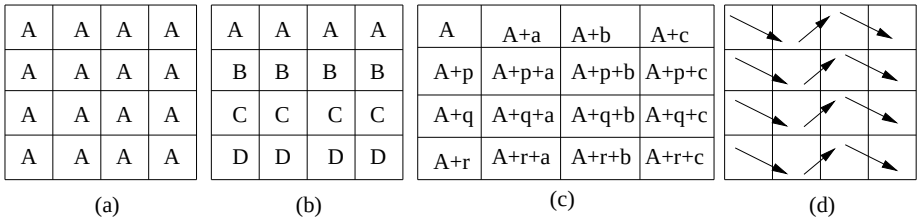


Figure 1: Types of biclusters: (a) Biclusters with constant values (b) Biclusters with constant rows (c) Biclusters following an additive model (d) Biclusters with coherent evolutions.

biclusters corresponding to this definition. [15] have classified the biclusters found by these algorithms into four categories, as shown in Figure 1. These categories include (i) biclusters with constant values, (ii) biclusters with constant rows or columns, (iii) biclusters following an additive (or multiplicative) model, and (iv) biclusters with coherent evolutions. The problem we address in this paper is the same as "finding constant value biclusters" as defined in [15].

Several biclustering algorithms, such as CC ([9]), ISA ([6]), SAMBA ([19]), OPSM ([5]) and co-clustering ([10]) , have been proposed to find different types of these biclusters. However, these approaches suffer from three common limitations. (i) Most of these approaches either adopt top-down greedy schemes that start from all rows and columns, and then iteratively eliminate rows and/or columns to optimize their objective function ([10, 9]), or start with a random initial seed and use heuristics to converge to the final bicluster ([6, 5]). Due to the use of these heuristics, these algorithm are unable to search the space of all possible biclusters exhaustively. (ii) The objective of these approaches is different from finding coherent value submatrices. For example, CC finds constant row biclusters which have low mean squared residue score and SAMBA finds maximum weight bicliques. (iii) Small biclusters tend to get overshadowed by noise and/or by larger biclusters due to the top-down nature of the search. In particular, these techniques are not meant to find constant value biclusters that are of interest to us.

Interestingly, pattern mining algorithms developed in association analysis ([2, 8, 11]) also produce biclusters in binary market-basket-type data, where each row is a transaction that indicates the purchase of items (represented along columns) in a store. A pattern is a group of items (itemset) purchased together in atleast a given fraction of transactions and it can be represented as a submatrix with supporting transactions as rows and items in the itemset as columns, with all the values included being 1. So, these patterns are essentially similar to constant value biclusters that we seek to discover. However, they only work with binary data sets. Recently, [16] have extended these algorithms to find constant row/column biclusters in real-valued data, but their approach still can not discover constant value biclusters exclusively. Although constant row biclusters may include constant value biclusters, these need to be identified by post-processing, as we discussed in the evaluation section, this is not an effective way to find coherent value biclusters.

In this paper, we present a novel framework to exhaustively discover all RCBs in a given GI dataset. For this, we define the notion of a coherent submatrix whose values are within a pre-specified (relative) range, and refer to it as a *Range Constrained Block* (RCB). The measure of coherence used, named the *Range* measure, is monotonic in nature, and thus makes it possible to develop an Apriori-like algorithm ([1, 2]) to enumerate all RCBs whose value for the *Range* measure is lower than the user-specified threshold. This algorithm is guaranteed to recover all such coherent submatrices in the given data set.

The rest of the paper is organized as follows. We discuss some related approaches for the bicluster discovery problem in Section 2. We present the RCB discovery framework in Section 3. Section 4 details the quantitative evaluation of RCBs. We conclude with suggestions for future work in Section 5.

2. RELATED APPROACHES

Although our work is the first systematic approach for the problem of finding constant value biclusters, this problem can be approached in other ways also. One of the most straightforward approaches would be to binarize the data matrix and use the Apriori algorithm [1] to find binary frequent patterns, which are also biclusters. However, this is not an ideal approach for our problem, since all the values are represented by 1 or 0, and thus even if such a pattern is found, there is no guarantee on the coherence of the values included in a bicluster so found. This problem is shared by Ma *et al.*'s approach [14] for finding highly connected subgraphs from a bipartite graph representation of GI data. Ma *et al.*'s approach further faces the problem of being non-exhaustive due to the heuristic search algorithm employed. Another possible approach is to generate multiple binary matrices with each matrix having 1s for values that are within in a small range (window). This approach can not find biclusters that have values that are in two adjacent windows but still in a small range. Below, we discuss three related approaches that focus on finding biclusters directly from real-valued data. Note that these methods were originally developed for microarray data, but the formulations and underlying principles apply directly to other types of data also.

2.1 Range Support Patterns (RAP)

Pandey *et al.* [16] recently proposed an association analysis approach for finding constant row/column biclusters (Figure 1(b)) directly from real-valued data. Here, they defined the *RangeSupport* measure of an itemset as the sum of the contributions of each transaction where the values of these items are within a pre-specified (relative) range threshold α, and are of the same sign. This contribution is defined to be the minimum of the absolute value among the items for a transaction that satisfies both these conditions, and zero otherwise. This definition makes *RangeSupport* anti-monotonic, and an Apriori-like algorithm is then used to mine constant row/column biclusters from the given data

set. This approach has several desirable properties, such as the exhaustive enumeration of all biclusters of this type, the possibility of overlaps between biclusters and the ability to discover small biologically meaningful biclusters. However, these biclusters are only guaranteed to be coherent over one of the dimensions (row or column), but not necessarily both the dimensions, as is required for constant value biclusters.

2.2 Cheng and Church's algorithm

Cheng and Church [9] (CC) proposed the first algorithm, which we refer to as CC, to find biclusters in microarray data. They used the mean squared residue (MSR) measure to capture the coherence of expression values among a set of genes across a subset of all the conditions, and focused on finding biclusters with low MSR values. However, since enumerating all such biclusters is an NP-hard problem, a greedy heuristic approach to discover such biclusters is used. This approach first starts with the entire matrix M and iteratively removes rows or columns that provided maximum reduction in the MSR score until the MSR score is below a user specified threshold, or a certain number of iterations is reached. Provisions are also made for finding overlapping biclusters. However, this algorithm faces several challenges in finding constant value biclusters. First, since a heuristic search algorithm is employed, it can not be guaranteed that all biclusters with an MSR lower than the specified threshold will be found. Also, CC generally finds biclusters of large sizes since the termination criteria are generally satisfied early in the search process. Finally, CC tends to find several biclusters with almost neutral (zero) values in them, since they have MSR=0, which may not be useful if these biclusters need to be analyzed further.

2.3 SAMBA

Tanay *et al.* [19] proposed the SAMBA algorithm for finding biclusters, which they define as a group of genes that jointly respond to a group of conditions. A gene is said to respond to a condition if its expression level changes significantly relative to its expression under normal conditions. The given gene expression data matrix is represented as a bipartite graph with genes and conditions as the two sets of vertices. An edge e connects gene u to condition v with weight 1 if the expression level of u is significant under v and -1 otherwise. The algorithm then tries to find maximal weight subgraphs, all of whose edges of the same sign, in this weighted bipartite graph using a heuristic search algorithm. The genes and conditions constituting these subgraphs are output as biclusters. It can be seen that, similar to binary pattern mining, SAMBA ignores the importance of the real values once it is determined if a value is significant or not. Thus, the coherence of values constituting the resultant biclusters is not guaranteed. Furthermore, SAMBA can not guarantee finding all possible maximal weight subgraphs, which is an NP-hard problem.

In summary, although various algorithms have been proposed for finding different types of biclusters, none of them exhaustively finds constant value biclusters, which are the focus of our work. The challenges faced by these approaches for this problem are reflected in the experimental results discussed in Section 4.

3. RCB DISCOVERY APPROACH

In this section we introduce an Apriori-like framework to mine RCBs from a real valued data set. For this, we first define a *range* measure to capture the semantics of an RCB and prove that it is monotonic. We then introduce a diagonal representation of a square sub-matrix, and describe how it can be used to efficiently mine square RCBs using an Apriori-like algorithm. This algorithm discovers a rectangular RCB in the form of multiple, overlapping square RCBs. Finally, we present an Apriori-like algorithm to merge these square RCBs, at the end of each level in the previous algorithm, into rectangular RCBs. Note that although the RCB mining framework is defined below for a data matrix that has items of the same type on both of its dimensions, it can be also be used for a data set that has different types of items along the two dimensions.

3.1 Range measure

We defined RCB as a submatrix that has all values within a given range. This range can simply be defined as a difference between the maximum and minimum value of the submatrix. However, since most real data sets have a wide range of values, we use a relative form of range to make its definition more versatile. Formally, if G is any real valued positive data matrix, for any submatrix G_{IJ}, where $I = i_1, i_2, \ldots, i_k$ and $J = j_1, j_2, \ldots, j_l$ constitute its two dimensions, and whose each element is g_{ij} ($i \in I$ and $j \in J$), the range of G_{IJ} is defined in a straight-forward manner as:

$$range(G_{IJ}) = \frac{\max_{i \in I, j \in J}(g_{ij}) - \min_{i \in I, j \in J}(g_{ij})}{\min_{i \in I, j \in J}(g_{ij})} \quad (2)$$

However, another complicating aspect of real-valued data sets is that they contain both positive and negative values. For example, in genetic interaction data, positive and negative values represent different types of interactions, as discussed earlier. This factor needs to be incorporated into the definition of *range*, so that the resultant RCBs are coherent not only in values, but also in their signs. We ensure this by enforcing this constraint into the definition of the *range* measure as formulated in Equation 3. Here, the range of a submatrix that includes both positive and negative values is simply set to infinity, so that it is not considered as an RCB. Note that this constraint is supported by research on genetic interactions, where it has been shown that groups of genes (modules) having interactions of same type (sign) are more functionally related than those involved in very different types of interactions [21].

$$r(G_{IJ}) = \begin{cases} range(abs(G_{IJ})) \\ \quad (if\ g_{ij} > 0\ \ \forall\, i \in I,\ \forall\, j \in J \\ \quad or \\ \quad g_{ij} < 0\ \ \forall\, i \in I,\ \forall\, j \in J) \\ \infty \quad (otherwise) \end{cases} \quad (3)$$

Using this definition, it can be shown that the *range* measure has a monotonicity property, as shown by the following.

THEOREM 1. *Range measure is monotonic*

PROOF. Consider a submatrix G_{IJ} of a matrix G, where $I = i_1, i_2, \ldots, i_k$ and $J = j_1, j_2, \ldots, j_l$ are the two dimensions of the submatrix and $r(G_{IJ}) \in [0, \infty)$. Let $I' = I \cup i_{k+1}$ and $J' = J \cup j_{l+1}$.

The range of the submatrix $r(G_{I'J'})$ will fall into one of the following:

• The elements in $G_{I'J'}$ have different signs: Now, $r(G_{I'J'}) =$

∞. Since $r(G_{IJ}) \in [0, \infty)$, $r(G_{I'J'}) \geq r(G_{IJ})$.
• The elements in $G_{I'J'}$ have the same sign: Two sub-cases are possible in this scenario:
– $max(G_{I'J'}) = max(G_{IJ})$ and $min(G_{I'J'}) = min(G_{IJ})$. Then $r(G_{I'J'}) = r(G_{IJ})$.
– $max(G_{I'J'}) \geq max(G_{IJ})$ and/or $min(G_{I'J'}) \leq min(G_{IJ})$. Then $r(G_{I'J'}) \geq r(G_{IJ})$.
Thus, $r(G_{IJ})$ is monotonic. □

Due to this monotonicity property, the *range* measure can be used in a bottom-up Apriori-like algorithm to enumerate the all the RCBs in a given data matrix that satisfy the given range constraint. Note that traditional frequent pattern mining algorithms focus on patterns with support greater than a user-specified threshold, while we discover RCBs with range lower than the user-specified threshold, thus enabling us to ensure coherence in both the dimensions simultaneously. However, due to the complexities in this search process discussed below, we adopt a two-step process, in which first all the square submatrices that qualify to be an RCB are enumerated, and then, these square RCBs are merged to form rectangular RCBs of arbitrary sizes. We describe the individual components of this process below.

3.2 Challenges in finding RCB patterns using the standard Apriori like approaches

This process of finding RCBs, a search in a combination of two dimensions, is a non-trivial problem and is computationally hard compared to the problem of frequent itemset discovery. In discovering frequent itemsets, the Apriori algorithm starts with a single items that are frequent. These individual items are then merged to form candidate size-2 itemsets and their supported is computed. All frequent pairs are further merged to form candidate itemsets of size-3 and their support is computed. This process is repeated until no more bigger itemsets can be found. One can design a similar approach for finding RCBs that hold a range constraint (r) in a given genetic interaction matrix with m rows and n columns as follows: all $m \times n$ individual elements in the matrix are considered as candidate size-1 $\times$ 1 RCBs. Each element that has a non-zero value is considered as a size-1$\times$1 RCB, because range (r) for zero valued elements is ∞. Now, for each of the size-1 $\times$ 1 RCBs a row or column is added to form candidate size-1$\times$2 (or candidate size-2$\times$1) RCBs. The range measure can then be computed to determine size-1 $\times$ 2 (or size-2 $\times$ 1) RCBs. This approach for finding an $m \times n$ RCB involves enumeration of $(2^m - 1)(2^n - 1)$ smaller submatrices in the process of discovering it. Where as, finding a size-n itemset involves enumerating $(2^n - 1)$ smaller itemsets.

Thus, RCB discovery is a combinatorial search in $m \times n$ space, whereas traditional frequent pattern mining is a search in n-dimensional space. As a result searching for RCBs in matrix with large dimensionality can be computationally inefficient if done in a simplistic fashion. In the following subsection we present an approach to represent a square RCB in the form a one-dimensional vector which helps in discovering square RCBs efficiently.

3.3 Diagonal representation of square RCBs

We make use of the observation that a square sub-matrix can be represented by the indices along the diagonal. Consider a square sub-matrix G_{IJ}, where $I = i_1, i_2, \ldots, i_k$ and $J = j_1, j_2, \ldots, j_k$ are its two dimensions. This sub-matrix is can be represented by its diagonal $\{(i_1, j_1), (i_2, j_2), \ldots (i_k, j_k)\}$. In other words, we can write it as $\{d_{i_1 j_1}, d_{i_2 j_2}, \ldots d_{i_k j_k}\}$, where $d_{i_m, j_n} = (i_m, j_n)$ for $\forall i_m \in I,\ j_n \in J$. This *diagonal-set* for a matrix can be considered analogous to an *itemset* in the traditional association analysis. This representation makes it easier to represent a size k square sub-matrix as a one-dimensional vector of pairs of indices of length k. Using this representation, all square RCBs can now be enumerated efficiently in a manner similar to discovering frequent itemsets in binary datasets. Thus the diagonal representation facilitates efficient Apriori-based search for RCBs by mapping the two-dimensional search space into one-dimensional search space.

In the following sub-section we present an Apriori like algorithm that makes use of the diagonal representation for discovering square RCBs. Since RCBs can be rectangular, we then present an efficient algorithm that merges the square RCBs of same size into rectangular RCBs in an Apriori-like fashion.

3.4 Mining Square RCBs

As the first step of our RCB discovery process, we use the following Apriori-like algorithm to discover all square RCBs in a given data matrix for a user-specified range threshold.

Algorithm 1: 2-D Square RCB Approach

Input:
i. G, a real valued data matrix of size $|m \times n|$, with items $I = \{i_1, i_2, \ldots i_m\}$ and $J = \{j_1, j_2, \ldots j_n\}$ along the two dimensions
ii. δ, a range threshold
Output:
All square submatrices $G_{I'J'}$ in G with $r(G_{I'J'}) \leq \delta$

$k = 1$
$S_k = \{d_{ij} | g_{ij} \neq 0\}$ // Find all size $|1 \times 1|$ RCBs
while $F_k \neq \emptyset$ **do**
  $k = k + 1$
  $CS_k = Apriori - gen(S_{k-1})$
    // Generate all size k candidate RCBs
  for each candidate $cs_k \in CS_k$ **do**
    compute $r(cs_k)$ using Eq. 3
  end
  $S_k = \{cs_k | cs_k \in CS_k \wedge r(cs_k) \leq \delta\}$
end
Result = $\bigcup S_k$

This algorithm takes a real valued data matrix and a user-specified range threshold as input and enumerates all square sub-matrices in the given matrix for which the range constraint holds. To begin, since the *range* measure defined in Equation 3 is ∞ for any sub-matrix that has all zero elements, each non-zero element in the given data matrix is treated as a level-1 square RCB. At level-2, each level-1 RCB is paired with another level-1 RCB that has both indices greater than itself to form candidate level-2 square RCBs. Now, all the candidates that satisfy the range constraint are output as level-2 square RCBs. At the next level, a candidate level-3 square RCB is generated from two level-2 square RCBs using $Apriori - gen$ [2], a method used to efficiently generate candidate level-k itemsets from level-$k-1$ frequent itemsets. $Apriori - gen$ constructs a new candidate level-3 square RCB by combining two level-2 RCBs if they have one element of the diagonal-set in common. More generally, a candidate level-$(k+1)$ square RCB is generated

from two level-k square RCBs if their diagonal-sets overlap in $k-1$ elements. Let $\{d_{i_1j_1},\ d_{i_2j_2},\ \ldots\ d_{i_{k-1}j_{k-1}},\ d_{i_kj_k}\}$ and $\{d_{i_1j_1},\ d_{i_2j_2},\ \ldots\ d_{i_{k-1}j_{k-1}},\ d_{i_{k+1}j_{k+1}}\}$ be two level-k square RCBs. Then, a level-$(k+1)$ candidate square RCB is obtained by merging these two level-k square RCBs that have $k-1$ elements of their diagonal-sets overlapping. Finally, the candidate RCBs that satisfy the range constraint are enumerated as the level-$(k+1)$ square RCBs. This process is continued until no more sub-matrices satisfy the range constraint.

3.5 Combining Square RCBs into Rectangular RCBs

Algorithm 1 can be used to discover all square RCBs in a given data matrix. However, a naturally existing rectangular RCB of size $m \times n$ $(m > n)$ will result in $\binom{m}{n}$ square RCBs that share the same n indices along the shorter dimension. As each rectangular sub-matrix is broken into multiple square sub-matrices that share the shorter dimension, we need a method to join them. It is important to note that all the squares that share the same dimension may not form a rectangular RCB, but some combinations of these squares could potentially hold the range constraint to form rectangular RCBs. So, we use the following Apriori-like algorithm to combine these square RCBs into rectangular ones that satisfy the range constraint.

Algorithm 2: MINING RECTANGULAR RCBS

Input:
i. G, a real valued data matrix of size $|m \times n|$, with items $I = \{i_1, i_2, \ldots i_m\}$ and $J = \{j_1, j_2, \ldots j_n\}$ along the two dimensions
ii. δ, a range threshold
iii. All maximal square RCBs at level-l that are enumerated by Algorithm 1
Output:
All rectangular submatrices $G_{I'J'}$ whose smallest dimension is of length k in G with $r(G_{I'J'}) \leq \delta$

for each set of square RCBs $S = \{s_1, s_2, \ldots s_t\}$ that have one dimension of the square common **do**
  $k = 1$
  $R_k = \{s_i | \forall s_i \in S\}$
  while $R_k \neq \emptyset$ **do**
    $k = k + 1$
    $CR_k = Apriori - gen(R_{k-1})$
      // Generate all size k candidate rectangular RCBs
    for each candidate $cr_k \in CR_k$ **begin**
      compute $r(cr_k)$ using Eq. 3
    end
    $R_k = \{cr_k | cr_k \in CR_k \wedge r(cr_k) \leq \delta\}$
  end
  Result $= \bigcup R_k$
end

Algorithm 2 takes all the maximal square RCBs at level-l of Algorithm 1 as input and for each group of square RCBs that have the same set of items across one dimension, it first considers these square RCBs as level-1 rectangular RCBs, analogous to level-1 itemsets in Apriori. It then enumerates all possible combinations of level-1 rectangular RCBs using $Apriori - gen$. Let $s_{(S,T_i)}$ and $s_{(S,T_j)}$ be the square RCBs that have one dimension (S) in common and the other dimension $(T_i, T_j \in T)$ that is different. A candidate level-2 rectangular RCB is obtained by merging the dimension that is not common $(T_i \cup T_j)$ and by retaining the common dimension (S). So, the set of level-2 candidates is represented as $cr_{(S,(T_i \cup T_j))}$. The candidates that satisfy the range threshold are treated as level-2 rectangular RCBs R_2. Similarly, at any level-k, $Apriori - gen$ is used to find candidate level-k rectangular RCBs CR_k from level $k-1$ rectangular RCBs R_{k-1}. All candidates that satisfy the range constraint are enumerated as level-k rectangular RCBs R_k. This process is iterated until no more candidate rectangular RCBs satisfy the range threshold.

Thus, using Algorithm 2 all arbitrary size RCBs are enumerated that satisfy the user-specified range constraint. The correctness of the overall RCB discovery algorithm is ensured, since only the candidates that pass the range threshold are returned as RCBs. Theorem 2 proves the completeness of this algorithm.

THEOREM 2. *RCB approach discovers all valid RCBs at a given range r in a given data set G.*

PROOF. We prove this by induction. We first prove that all valid square RCBs will be discovered by Algorithm 1. Let G be the input data matrix. In the first level, all non-zero elements in G are considered as level-1 square RCBs, since the range of a non-zero element is zero and that of a zero element is ∞. So, level-1 is complete. Now, consider the set of level-k square RCBs S_k and assume that it is complete. Since, Algorithm 1 uses $Apriori - gen(S_k)$ to enumerate all possible candidate level-$(k+1)$ square RCBs CS_{k+1} and tests them for the range constraint, level-$(k+1)$ is also complete. By induction, Algorithm 1 generates the complete set of square RCBs at any level.

We now prove that Algorithm 2 generates all possible rectangular RCBs from the set of level-l square RCBs. Since Algorithm 2 finds rectangular RCBs from each group of level-l square RCBs that share one dimension, we focus on proving that Algorithm 2 generates all possible rectangular RCBs for one such group. Since Algorithm 2 considers all level-l square RCBs that have one dimension in common, the set of level-1 rectangular RCBs R_1 is complete. Now, consider level-k rectangular RCBs R_k and assume that it is complete. Since $Apriori - gen(R_k)$ is used to generate all candidate level-$(k+1)$ rectangular RCBs CR_{k+1} and each candidate in CR_{k+1} is tested for the range constraint, level-$(k+1)$ in Algorithm 2 is complete. By induction, Algorithm 2 generates the complete set of rectangular RCBs at any level.

This proves the completeness of our algorithm. □

We now discuss the performance of this RCB discovery algorithm when tested on genetic interaction data.

4. EXPERIMENTAL RESULTS

In this section we evaluate the proposed RCB discovery approach on a genetic interaction data set and compare its performance with other approaches discussed in Section 2, namely binary frequent patterns(FP), RAP, CC and SAMBA.

4.1 Experimental Design

We tested our proposed scheme on a dataset of genetic interactions consisting of weighted positive interactions, among 500 query (row) and 3893 array (column) genes. The values in this data set belong to the interval $[0, 357]$. As the values close to zero correspond to neutral interactions and

are also prone to distortion due to noise, we used a sparsification threshold γ, and replaced the values less than γ by zero. Sparsification threshold used in our experiment vary between 30 and 50. Sparsified matrices are used for mining maximal RCB biclusters at varying range thresholds between 0.3 and 1.5 and different aspects of the performance of the RCB mining approach are analyzed. Note that only RCBs larger than 3×3 are analyzed to simplify the discussion since there are far too many blocks of smaller sizes (many of which could be spurious due to noise in the data).

To assess the relative utility of our proposed scheme with respect to the approaches discussed in Section 2, namely binary frequent patterns(FP), RAP, CC and SAMBA, we also generated biclusters using them. Binary frequent patterns are generated by first constructing a binary matrix $G_{1/0}$ from the data matrix G, where each element $g_{1/0_{ij}}$ is 1 if its corresponding element in G, $g_{ij} \geq \gamma$, and 0 otherwise. Borgelt's implementation [7] of Apriori algorithm [2] is used to discover frequent patterns (biclusters) from the binary matrix $G_{1/0}$. The lowest possible support threshold with which this implementation could run without 'running out of memory' was chosen at different sparsification thresolds, and is reported in Table 1 as σ. The RAP code[1] is then used to discover biclusters on the matrix G using support thresholds determined using the median support of each item. Since RAP is meant to find one-dimensional constant row biclusters (Figure 1(b)), we used it to discover biclusters from both the dimensions (array and query genes respectively) of G, in order to ensure completeness for comparison. The RAP and binary patterns (biclusters) obtained from these transformed matrices are filtered out if the length of any one dimension is less than 3, since we limit our analysis to RCB biclusters of size 3×3 or more. Note that only closed patterns obtained from Apriori and RAP patterns are considered for further analysis, since they represent all distinct patterns. Note that maximal patterns found by RCB (since they are computed in two dimensions) correspond to closed itemsets. In addition, we also generated biclusters from this data set sparsified using $\gamma = 40$ using the SAMBA algorithm implemented in the Expander tool [17] with the parameter *#probes* set to 10 and 100. Finally, we also generated 100 biclusters using CC with two parameter settings $\delta = 0.3$ and $\delta = 0.5$ as specified in the BiCAT tool [4]. CC when run on a sparsified version of the data discovered biclusters filled with zeros, owing to the fact that MSE for a bicluster with zeros is zero. So, to help CC find reasonbale biclusters the original version of the data matrix was used. All these experiments are run on an eight-processor computer with total 32 GB memory, running Linux.

4.2 Quantitative Evaluation of RCBs

Table 1 provides a global overview of the performance of the different algorithms. In this subsection, we discuss some of these aspects in detail. This table presents the different parameter settings, number of biclusters found, variation in the size of biclusters (denoted as $L - M$, where $L = |I| \times |J|$ for the smallest bicluster G_{IJ} and M is computed similarly for the largest bicluster) and range (r as in Eq 3).

From Table 1 it can be seen that the number of RCBs, FP patterns discovered at low sparsification threshold γ are more in number due to the density of the values in the matrix. As the sparsification threshold is increased, RCB, FP and RAP discovery approaches discover fewer patterns. It is important to note that FP and RAP approaches use support to contain the complexity of the search space, where as RCB uses range measure to contain the complexity. Since no coherence in values is ensured on FP and RAP patterns they have high average range, where as the patterns discovered by RCB patterns have the average range less than the specified thresholds. The CC and SAMBA patterns are generally very large in size because of the top-down approach that is employed and also contain many 0 values within them. So, we do not consider these for further analysis.

Title	Parameter Settings	# biclusters	Size of biclusters	Average Range (r)
RCB biclusters				
RCB1	$\delta = 0.3,\ \gamma = 30$	8794	9-18	0.2617
RCB2	$\delta = 0.5,\ \gamma = 30$	107799	9-27	0.4352
RCB3	$\delta = 0.3,\ \gamma = 40$	991	9-15	0.2600
RCB4	$\delta = 0.5,\ \gamma = 40$	12099	9-24	0.4333
RCB5	$\delta = 0.7,\ \gamma = 40$	44044	9-39	0.6041
RCB6	$\delta = 1,\ \gamma = 40$	127884	9-51	0.8541
RCB7	$\delta = 0.7,\ \gamma = 50$	6847	9-30	0.5920
RCB8	$\delta = 1,\ \gamma = 50$	16182	9-45	0.8237
RCB9	$\delta = 1.2,\ \gamma = 50$	24584	9-48	0.9840
RCB10	$\delta = 1.5,\ \gamma = 50$	32049	9-57	1.1722
Binary Patterns				
FP1	$\sigma = 6,\ \gamma = 30$	581916	30-102	4.6778
FP2	$\sigma = 4,\ \gamma = 40$	142480	12-92	2.3624
FP3	$\sigma = 3,\ \gamma = 50$	30663	9-75	2.1087
RAP biclusters (on query genes)				
RAP1	$\sigma = 500,\ \delta = 0.5$	146	15-45	1.8999
RAP2	$\sigma = 500,\ \delta = 0.7$	610	15-72	1.9673
RAP3	$\sigma = 500,\ \delta = 1$	1758	15-93	2.1386
RAP biclusters (on array genes)				
RAP4	$\sigma = 257,\ \delta = 0.5$	2662	9-27	1.5870
RAP5	$\sigma = 257,\ \delta = 0.7$	9138	9-33	1.7070
RAP6	$\sigma = 257,\ \delta = 1$	24920	9-40	1.8494
CC biclusters				
CC1	$\delta = 0.3$	100	187-3249	∞
CC2	$\delta = 0.5$	100	273-2940	∞
SAMBA biclusters				
SAMBA1	# probes = 10	10	234-1314	∞
SAMBA2	# probes = 100	349	120-1450	∞

Table 1: Statistics of biclusters generated at various parameter settings from the GI data set

In summary, our RCB mining approach is able to discover a larger number of coherent blocks from the genetic interaction matrix, which have low average range compared to the competing approaches. Furthermore, RCB biclusters also cover more interactions in a GI dataset as shown in Section 4.5.

4.3 Statistical significance of RCBs

One of the important steps in analyzing real-life data, such as the GI data in our study, is to assess the validity and significance of the entities being mined from them. A common method for doing this is to randomize the data and mine the same type of entities from this randomized data set. A meaningful analysis should find significantly more of these entities from the real data as compared to the randomized version. We performed such an analysis for RCB mining from GI data. We found 12099 RCBs in the RCB4 set using $\delta = 0.5$ and $\gamma = 40$ from the GI data set. Also, we generated 30 randomized versions of this data set by randomly permuting the entries in each row, and derived RCBs from them using the same parameter settings as RCB4. Two observations can be made from these results. First, very few RCBs (348

[1]http://vk.cs.umn.edu/gaurav/rap/

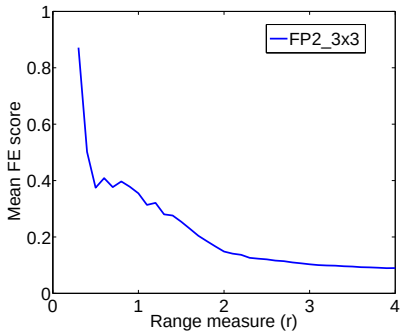


Figure 2: Relationship between Range (r) and Functional Relatedness (FE).

on average) are found from the randomized data sets. Second, the sizes of these RCBs are substantially smaller than the sizes of those in the RCB4 set, with most of the RCBs in the RandRCB4 (over 90%) being 3×3 blocks. The results of this analysis indicate that the products of our RCB mining approach are indeed statistically significant. The biological significance of some of these blocks is discussed in the next section.

4.4 Functional Evaluation

Since the groups of genes constituting a submatrix with coherent values are expected to be functionally coherent, we evaluate this using a measure of functional relatedness derived from sources of information about gene function that are independent of GI data. In particular, we use Functional-coExpression (FE) that is derived from 40 different micro-array data sets [12]. Here the probability for two genes to be co-annotated to the same Gene Ontology Biological Process (GO BP) function is computed on their levels of co-expression in these datasets. We refer interested readers to the corresponding paper for details on this measure, but stress that the basic purpose is to quantify the degree of functional relatedness of two genes.

Genes constituting both the dimensions of a submatrix are said to be functionally related, if each gene-pair which is a combination of one gene from each group has high functional-coExpression score. So, for any given submatrix, we compute the functional relatedness as the mean of the FE score for each interacting gene pair covered by the submatrix. Although it is possible to evaluate the relationship between range and the FE score on RCB patterns, the average range of the binary patterns is higher (as shown in Table 4.2) and so we use the binary patterns to evaluate this relationship. We evaluated the relationship between the functional relatedness and the range measure r, by enumerating all possible 3×3 size blocks for randomly chosen 10,000 patterns in FP2, which we refer to as FP2_3×3.

The FE score and the range are computed for each such 3×3 size block enumerated. The median of the FE scores for corresponding range values are presented in Figure 2. It can be seen that the blocks with a small range value, have high FE score and the blocks with large range value has low mean FE score. This indicates that the groups of genes representing the coherent submatrices are more functionally related than the groups of genes representing the less coherent submatrices.

4.5 Comparison of our RCB finding algorithm with post-processing of FP and RAP

It is also possible to enumerate all possible submatrices from binary patterns and RAP patterns and select the submatrices that satisfy a given range threshold. To demonstrate the effectiveness of RCB, we enumerated all possible submatrices that satisfy the range constraints for FP1, FP2, RAP1, RAP2 and RAP3. As the size of the CC and SAMBA patterns are typically large relative to the FP and RAP patterns, enumerating all possible submatrices is infeasible. So, we restrict our analysis to FP and RAP patterns. For FP1, the range threshold $\delta = 0.3$ and $\delta = 0.5$ are used to compare them with RCB1 and RCB2 respectively. For FP2, the range threshold $\delta = 0.3$, $\delta = 0.5$, $\delta = 0.7$ and $\delta = 1$ are used to compare them with RCB3, RCB4, RCB5 and RCB6 respectively. For RAP patterns the δ that was used in Table 1 was chosen. For each set of patterns, the number of genes covered in both the dimensions, total number of interactions covered i.e. the area in the data matrix that the discovered patterns cover, time taken are tabulated in Table 2.

Title	Range (δ)	# Genes covered	# Interactions covered	Time taken (in hours)
RCB biclusters				
RCB1	0.3	(408, 2437)	26664	1.62
RCB2	0.5	(484, 3391)	54842	3.2
RCB3	0.3	(216, 765)	4959	0.29
RCB4	0.5	(327, 1594)	16550	0.41
RCB5	0.7	(371, 1986)	22516	0.8
RCB6	1	(415, 2234)	26054	7.12
Binary Patterns				
FP1a	0.3	(293, 641)	6169	0.2
FP1b	0.5	(433, 1263)	22947	0.32
FP2a	0.3	(170, 421)	2642	0.06
FP2b	0.5	(286, 981)	10858	0.07
FP2c	0.7	(340, 1262)	16394	0.1
FP2d	1	(384, 1447)	20034	0.3
RAP biclusters (on query genes)				
RAP1	0.5	(53, 303)	1467	0.04
RAP2	0.7	(89, 756)	4959	0.06
RAP3	1	(111 ,1123)	8607	0.23
RAP biclusters (on array genes)				
RAP4	0.5	(156, 277)	2404	0.13
RAP5	0.7	(212, 502)	5987	0.28
RAP6	1	(280, 658)	9648	0.71

Table 2: Comparison of RCB with FP and RAP (with post processing) at various range thresholds.

RCBs cover more number of genes in both dimensions than FP and RAP patterns. Specifically, comparing the coverage of the sets RCB1 and FP1a, RCB1 covers approximately twice as many genes covered by FP1a in the query dimension and four times as many genes in the array dimension. They also cover four times as many interactions covered by the FP1a. This difference is relatively less at high sparsification thresholds γ, due to the sparse nature of the resulting binary matrix. The coverage of the genes and interactions is much less for the RAP patterns. This is due to the fundamental difference between the RCB approach and the general frequent pattern based approach. The RCB approach builds blocks in a bottom up fashion starting with a 1×1 block and gradually increasing its size in either dimensions while using range measure to control the complexity of the search space. On the other hand, FP based approaches start with single item that can have some support, which monotonically decreases as the size of itemset increases. The use of high support thresholds needed to contain the complexity resulting from the high density of the

data prevents FP based approaches from discovering small patterns that RCB can capture. For example, FP1 patterns are generated using $\sigma = 6$ which means each pattern generated should have atleast 6 genes on the query dimension. On the other hand, the coherent blocks of smaller sizes (of the order 3×3) exist in a large number compared to the bigger blocks in the data set (as shown in Table 4.2). So, these patterns cannot be discovered using the traditional frequent pattern based approaches especially at low sparsification thresholds, whereas RCB can discover all such patterns for a given range threshold δ.

On the other hand, from Table 2 the time taken for RCB generally appears to be larger than that of FP patterns, but note that the RCB patterns usually cover many more number of interactions than the FP patterns. Considering the set of biclusters in RCB6 and FP2d, RCB6 appears to require more time, but FP2d covers less number of interactions that of RCB6 because of a support threshold of 4, which causes it to miss many patterns of size 3×3. Note that the lowest possible support is being used to generate the FP patterns without 'running out of memory'. Similarly, the time taken for discovering RCB2 is more than that of FP1b. However, the number of interactions covered by RCB2 is more than twice as many as FP1b, also due to the use of high support to contain the complexity resulting from the density of the matrix at sparsification level $\gamma = 30$. This indicates that the RCB is an efficient and systematic approach to discover all the submatrices with range less than a given threshold than the other approaches.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a novel association analysis framework for mining (nearly) constant value submatrices from real valued genetic interaction datasets. We evaluated the proposed RCB discovery approach and compared its performance with other approaches, namely binary frequent patterns (FP), RAP, CC and SAMBA. Our results show that the gene modules representing the biclusters with similar values are more functionally related than the gene modules representing biclusters with diverse values. Furthermore, our approach can exhaustively find all the biclusters with range r less than a given threshold. This is not possible with other approaches, even when they are coupled with an exhaustive post-processing phase to enumerate submatrices with range within a given δ. Finally, we have shown that the RCBs discovered are statistically significant and are also biologically meaningful. This work can benefit from further research in many directions. The process of discovering RCBs can be made faster using specialized data structures and algorithms, such as hash trees. Our approach like other association analysis based approaches, provides a large number of patterns, many of which may be slight variation of the other patterns. Summarization techniques such as those in [22] will be helpful for the effective utilization of RCB patterns in practical settings.

6. ACKNOWLEDGEMENTS

This work was supported by NSF grants #IIS0916439, #CRI-0551551, a University of Minnesota Rochester Biomedical Informatics and Computational Biology Program Traineeship Award. Access to computing facilities was provided by the Minnesota Supercomputing Institute.

7. REFERENCES

[1] R. Agrawal et al. Mining association rules between sets of items in large databases. In *Proc. SIGMOD*, pages 207–216, 1993.

[2] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. In *Proc. VLDB*, pages 487–499, 1994.

[3] L. Avery and S. Wasserman. Ordering gene function: The interpretation of epistasis in regulatory hierarchies. *Trends in genetics*, 8(9):312, 1992.

[4] S. Barkow et al. BicAT: a biclustering analysis toolbox. *Bioinformatics*, 22(10):1282–1283, 2006.

[5] A. Ben-Dor et al. Discovering Local Structure in Gene Expression Data: The Order-Preserving Submatrix Problem. *JCB*, 10(3-4):373–384, 2003.

[6] S. Bergmann et al. Iterative signature algorithm for the analysis of large-scale gene expression data. *Physical Review*, 67(3):031902, 2003.

[7] C. Borgelt. Efficient implementations of apriori and eclat. In *FIMI*, 2003.

[8] A. Ceglar and J. F. Roddick. Association mining. *ACM Comput. Surv.*, 38(2):5, 2006.

[9] Y. Cheng and G. Church. Biclustering of Expression Data. In *Proc. ISMB Conference*, pages 93–103, 2000.

[10] I. Dhillon et al. Information-theoretic co-clustering. In *Proc. SIGKDD*, pages 89–98, 2003.

[11] J. Han et al. Frequent pattern mining: current status and future directions. *DMKD*, 15:55–86, 2007.

[12] C. Huttenhower, M. Hibbs, C. Myers, and O. G. Troyanskaya. A scalable method for integration and functional analysis of multiple microarray datasets. *Bioinformatics*, 22(23):2890–2897, 2006.

[13] R. Kelley and T. Ideker. Systematic interpretation of genetic interactions using protein networks. *Nature biotechnology*, 23(5):561–566, 2005.

[14] X. Ma et al. Mapping genetically compensatory pathways from synthetic lethal interactions in yeast. *PLoS ONE*, 3(4):e1922, 2008.

[15] S. C. Madeira and A. L. Oliveira. Biclustering algorithms for biological data analysis: a survey. *IEEE/ACM TCBB*, 1(1):24–45, 2004.

[16] G. Pandey et al. Association Analysis Approach to Biclustering. In *Proc. SIGKDD*, 2009.

[17] R. Shamir et al. EXPANDER – an integrative program suite for microarray data analysis. *BMC bioinformatics*, 6(1):232, 2005.

[18] R. St Onge et al. Systematic pathway analysis using high-resolution fitness profiling of combinatorial gene deletions. *Nature Genetics*, 39(2):199–206, 2007.

[19] A. Tanay et al. Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, 18(90001):S136–S144, 2002.

[20] A. Tong et al. Global mapping of the yeast genetic interaction network. *Science*, 303(5659):808–813, 2004.

[21] I. Ulitsky et al. From E-MAPs to module maps: dissecting quantitative genetic interactions using physical interactions. *Molecular Systems Biology*, 4(1), 2008.

[22] C. Wang and S. Parthasarathy. Summarizing itemset patterns using probabilistic models. In *KDD*, page 735, 2006.

New Privacy Threats in Healthcare Informatics: When Medical Records Join the Web

Fengjun Li[†], Jake Y. Chen[‡], Xukai Zou[‡], Peng Liu[†]
[†] College of IST, The Pennsylvania State University, University Park, PA, USA
[‡] Department of Computer and Information Science, IUPUI, Indianapolis, IN, USA
{fli,pliu}@ist.psu.edu, jakechen@iupui.edu, xkzou@cs.iupui.edu

ABSTRACT

In this paper, we study how patient privacy could be compromised from electronic health records (EHRs), especially with the help of today's information technologies. Current research on privacy protection is centralized around EHR: protecting patient information from being abused by authorized users or being accessed by unauthorized users. Limited efforts have been devoted to studying the attacks performed by manipulating information from external sources, or by joining information from multiple sources. Particularly, we show that (1) healthcare information could be collected by associating and aggregating information across multiple online sources including social networks, public records and search engines. Through attribution, inference and aggregation attacks, user identity and privacy are very vulnerable. (2) People are highly identifiable even when the attacker only possess inaccurate information. With real-world case study and experiments, we show that such attacks are valid and threatening. We claim that too much information has been made available electronic and available online that people are very vulnerable without effective privacy protection.

General Terms

Security

Keywords

Privacy, Healthcare informatics, EHR, social networks

1. INTRODUCTION

In recent years, with the development of healthcare informatics, a large amount of medical/healthcare records have been digitalized (in EHRs), for example, 43.9% of the US medical offices have adopted full or partial EHR systems by 2009 [7]. Since medical records are considered to be extremely sensitive, people start to concern on their privacy with digitalized healthcare data. Security and privacy becomes an important and popular topic in healthcare informatics research. Existing research on protecting user privacy in healthcare information systems could be summarized into three categories: (1) Defending against internal abuse of electronic health data, e.g. hospital personnel with authorization to access patients' records disclosing some of the private information for non-medical purposes. (2) Defending against unauthorized access to electronic health data, e.g. attackers hacking into hospital's databases or eavesdropping over the network communications. (3) Defending against re-identification attacks against published electronic health records, e.g. adversaries with access to de-identified healthcare data that are published for research purposes discovering the identities of record owners from a set of unprotected quasi-identifiers.

Meanwhile, as the Web gains its popularity and touches many aspects of our daily life, it becomes the largest open-access source of personal information. First, large amount of public records have been made accessible online, including phone books, voter registration, birth/death records, etc. Although some of them enforce certain restrictions to defend against abusers, it is still relatively easy or inexpensive to crawl/download such records. Second, more recently, online social network sites such as Facebook and MySpace have emerged to successfully attract a huge number of users, who willingly put their personal information to online social network sites to share with people. Unfortunately, with the new sophistication of information retrieval techniques and the advancement of searching techniques in search engines, it becomes unexpectedly easy to conduct Web-scale extraction of users' personal information that is readily available in various online social networks (e.g., [1, 8, 13, 3, 4]). As a result, malicious or curious adversaries could easily take advantage of these techniques to collect others' private information, which is readily available from online public records or various social networks.

In this way, the attackers possess powerful weapons and rich knowledge, which are somehow provided by the victims themselves, and are truly beyond the assumptions in the research literature. In this paper, we ask the question: "*when an attacker possesses a small amount of (possibly inaccurate) information from healthcare-related sources, and associate such information with publicly-accessible information from online sources, how likely the attacker would be able to discover the identity of the targeted patient, and what the potential privacy risks are.*"

To take a first step in answering this broad question, we study: (1) how user information from multiple online sources could be associated and utilized to compromise user privacy;

(2) how user identity could be identified by comparing approximate information with public databases.

2. ATTACKS ON HEALTHCARE RECORDS

With the broad adoption of electronic health records, security and privacy becomes extremely critical. Current researches on protecting patient privacy are centralized around the protection of EHRs by protecting patient information from being: (1) abused by authorized users; (2) accessed by unauthorized parties; or (3) re-identified from healthcare data published for research purposes.

To protect health care related information, regulations for disclosure are set and protected by law [2]. However, healthcare related personnel may violates privacy rules by disclosing or stealing private healthcare records for unauthorized usages, as depicted in [16]. This is a typical abuse/infraction with authorized data access. More often, the attackers do not have authorization for data access. They either eavesdrop or wiretap private information in transit or penetrate into EHR systems to get control of valuable health data. However, such types of attacks are often underestimated [18]. We believe such underestimation is partially from a fundamental misunderstanding that information revealed by carelessness or misuse is only one piece of the big picture and will not cause severe privacy disclosure. In later this paper, we will elaborate the severeness of such type of attacks in current information-rich context with an intuitive example.

Recently, there has been an increasing demand to publish the immense volume of EHRs for secondary purposes, such as research, government management, payment, and other marketing usages [14]. A typical EHR consists of a set of identifier attributes (e.g. *name*, *SSN*), quasi-identifier attributes (e.g. *gender*, *zipcode*), and sensitive attributes (e.g. *diseases*). Since privacy of record owners becomes a major concern, EHRs need to be de-identified [6] or anonymized [15] before data publishing. However, even with de-identified or anonymized data, sensitive attributes that pertains to an individual may be learned from other non-sensitive attributes in combination with external knowledge (e.g. voter registration list, phone books, etc.). The risks of such re-identification attacks have been intensively studied, which shows that the amounts and types of an attacker's external knowledge play an important role in reasoning about privacy in data publishing [11, 9, 12, 5]. However, it is not easy if not impossible for a data publisher to know upfront what external knowledge the attackers possess. Therefore, current research on privacy-preserving data publishing studies the problem from a theoretical perspective by making assumptions on attacker's background knowledge, quantifying external knowledge regardless of its content, and sanitizing the data to ensure the amount of disclosure is below a specified threshold [12, 5]. As a result, such protection, on one hand, does not take into account that large amount of external knowledge are accessible to the adversaries from various online sources (e.g. social networks), on the other hand, it might greatly distort the data and its secondary usages. Therefore, I believe it is of great importance to investigate the types and amounts of external knowledge that a powerful attacker possesses or infers from the immense volume of electronic data from *multiple online resources*. It not only provides evidence for efficient and optimal data sanitization, but also raises public concerns and awareness on the severeness of privacy threats and calls for effective protection.

3. ATTACKS FROM EXTERNAL SOURCES

Recently, online social networks are becoming extremely popular. Participants often voluntarily disclose personal information with surprising details. For example, *LinkedIn* users list their educational and working experiences to seek for potential career opportunities, and *MedHelp* users share details about their life and medical experiences expecting suggestions from others. A fundamental misunderstanding is that it is unlikely to link information of the same individual from different online resources. Unfortunately, with the sophistication of searching and information retrieval techniques, it is feasible for an attacker to *aggregate* personal information of a target user on different online resources, by associating unprotected but identifiable or semi-identifiable attributes (e.g. identical account names or email address of a careless user) [10]. Meanwhile, with governmental and industrial efforts, a large amount of public records have been digitalized and made available online. Most of them are indexed by commercial search engines, while others require a minimum subscription fee for full access. Adversaries could easily access and utilize such information to compromise others' privacy. Especially, it is possible to aggregate and associate information from multiple (possibly medical-related) external sources to identify patients from their poorly-anonymized data and reconstruct their complete profiles including identifiers and quasi-identifiers, as well as sensitive medical information.

Figure 1 demonstrates an example from a real-world case study: "Jean" (whose full name has been discovered but removed here for privacy protection) has type II diabetes, so she actively participates in two medicare social networks, *MedHelp* (www.medhelp.org) and *MP and Th1 Discussion Forum* (www.curemyth1.org). Her profile in *MP and Th1*, as shown in Figure 1 (1), contains birthdate, occupation, location, email addresses, and a text field about her interests on medical information. Her profile in *MedHelp*, as shown in Figure 1 (2), includes gender, age, location, and a text, from which we can learn astonishing details about Jean's medical conditions and history, e.g. *Diabetes II*, and *Ac1=5*, etc. More private attributes of Jean (e.g. times of doctor visit or diagnoses, prescription and medication) could be extracted from her postings on the two sites, respectively.

As shown in Figure 1, we compare all the attributes from both profiles: (1) Jean used identical (and relatively unique) username on both sites; (2) both profiles show Jean's current location - a small town with approximately 15K population; (3) birthdate shown in Profile 1 is consistent with the age shown in Profile 2; (4) Profile 1 shows "my husband" that indicates the owner is a female, which is consistent with the gender shown in Profile 2; and (5) both profiles show the same disease and symptoms. With all the evidences, we are able to link the two profiles at a certain confidence level, and associate the attributes from both profiles to the same individual. Further more, with the email address provided in profile 1, we are able to get profile 4 through Web search engines (note that email addresses are always considered as identifiers). Profile 4 includes a phone number (later it turns out to be a cell phone number) and a P.O. Box address, which also shows the same city as in Profiles 1 and 2. With the phone number from profile 4, we further discovered Profile 3, which is a job-related page containing Jean's cell and home phone numbers. Profiles 3 and 4 both contains the full name of "Jean", and we have a good hint on her occupation.

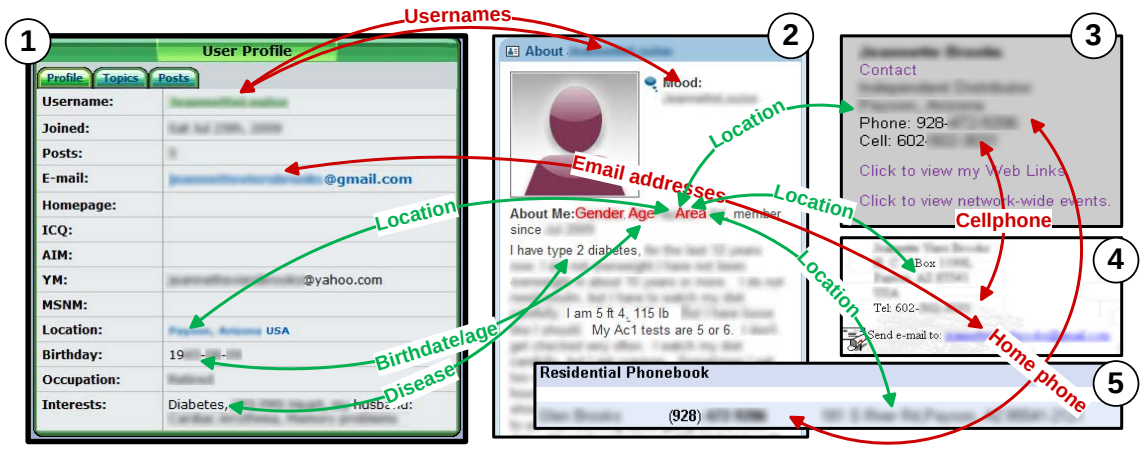


Figure 1: A real-world example of cross-site information aggregation.

Finally, with the home phone number, we are able to locate Jean's record in the residential phonebook, which shows her husband's name and their full home address. On the other hand, even without Profiles 3, 4 and 5, an attacker could also utilize public records to get more information about Jean: with the attribute set {gender, birthday, location}, Jean's identity (e.g. full name, address, and phone number) is recoverable from public birth records, voters registration records or online phone books.

By associating five profiles, we have collected Jean's full name, date of birth, husband's name, home address, home phone number, cell phone number, two email addresses, occupation, medical information including lab test results. With her full name, more information about Jean is subsequently discovered from various social networks. Finally, when Jean's hospital publishes de-identified patient records to support medical research, the attacker with external knowledge obtained from above process is highly likely to re-identify Jean's record.

The example reveals a serious privacy issue in both social networks and healthcare informatics. The entire process includes three steps: *attribution*, *inference*, and *aggregation* attacks. In attribution, identifiable, semi-identifiable or sensitive attributes are learned/extracted from various sources over the web. Particularly, three types of online resources are considered in the example: (1) public-accessible online databases: voters registration records, phone books, birth and death records, (2) online social network sites with explicit identifiable attributes (e.g. *LinkedIn*, *Facebook*, etc.) as well as specified healthcare-related social networks (e.g. *MedHelp*); and (3) commercial search engines, which index a good portion of the web. In inference, more attributes are further discovered from social activities and relationships through statistical learning or logical reasoning. In aggregation, records retrieved from different sources that potentially pertain to the same individual are linked under strong or weak evidences, in which strong evidences include matching identifiers or quasi-identifiers, and weak evidences are similarities identified from a statistical perspective. As we have shown in the example, the attacks are very valid and do not require excessive resources or techniques. Therefore, people are very vulnerable under such attacks, if they do not careful protect their online identities. A powerful privacy protection tool is expected to defend against such attacks.

4. ATTACKS WITH APPROXIMATE INFORMATION

Besides privacy attacks against digitalized medical records and healthcare information systems, adversaries also seek to obtain valuable information with non-technical kind of intrusions such as insider incidents or social engineering. With a vague definition, insider incidents often involve abuses such as inside personnel accidental leaking or stealing information, using pirated software, or accessing questionable webpages. Social engineering relies on people's unawareness of valuable information and carelessness in protection and becomes one of the major attacks towards user privacy. However, in most cases, information obtained from non-digital channels are not accurate due to the difficulty of accessing information, human capabilities or errors. For example, in today's medicine practice, many doctors record patients' medical information (e.g. symptoms, diagnoses, prescriptions, etc) with a audio recorder, and hire external companies to convert recordings into digital records. In the process, an adversary may steal the recording and learn detailed medical conditions of a patient, however, he may learn inaccurate information about patient's identity (e.g. he may not be able to get the correct spelling of the patient's name from doctor's voice). One may assume that the inaccuracy of attackers' knowledge may bring difficulty for them to compromise user identity or privacy. Unfortunately, such inaccuracy could be corrected by collaborating with external information sources, and the privacy risks causes by such attacks should no longer be ignored.

Here is a simple but representative example: Dr. Bob treats Alice in the hospital, while Malory eavesdrops the conversation, or peeps the record. Malory possesses the full prescription with an inaccurate version of Alice's last name (due to Dr. Bob's squiggling handwriting). Mallory does not know Alice, so he starts his attack by first looking into the phonebook for all "similar" names in the neighborhood. The question is: *What is Malory's opportunity of accurately recovering Alice's full name*?

To further articulate this problem, we define *k-approximate-anonymity* as follows:

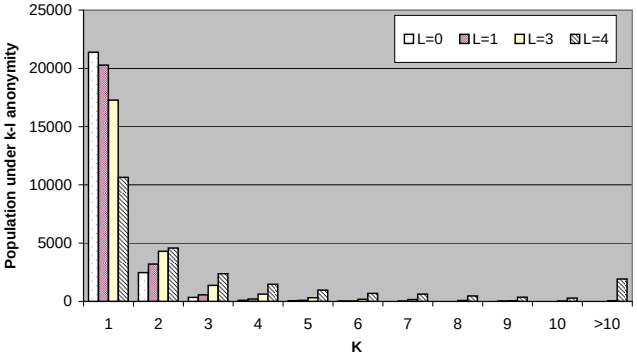


Figure 2: Population under *k-l-anonymity*.

Definition 1 (k-approximate-anonymity) *Given a dataset D, and a distance function $dist(r_1, r_2)$ that returns the distance for any two records on the dataset; for any record r, if there exists $k-1$ records r_x that $dist(r, r_x) <= l$ where l is a preset threshold, we conclude that D satisfies k-approximate-anonymity or k-l-anonymity with dist.*

In the above definition, when $l = 0$, it becomes the original *k-anonymity*. It basically says that when Mallory possesses approximate information on a target, he cannot distinguish the target from $k - 1$ other records in the database.

To simulate the above scenario, we have designed an experiment to study the identifiability of real names in the presence of inaccurate information from the attackers. We first implement a crawler to download the public residential phone book. In a few days, it successfully collects 24399 records from State College area, which covers approximately 64% of the population (according to 2000 cencus data). In each record, we have phone number, first and last names, and full residential address. In the experiments, we use full name as identifiers, and use the Levenshtein distance (edit distance) [17] as the distance function. For different threshold l, we show the population whose names are protected under *k-l-anonymity* in Figure 2.

From the figure, we can see that, with larger l, people are less identifiable with their names. However, overall, most (more than 70%) people are uniquely identifiable even when LD=2, and . It means that even though Malory gets an inaccurate name of the target, he has a good chance to correct the mistake and limit the target to a small range with the help of digital phonebooks. Even when Malory gets four letters wrong in the name, in more than 80% of the cases, his target is limited to no more than 5 candidates, i.e. he only needs to further examine no more than 5 records to identify the target. As we expected, people with longer names or unusual names are more vulnerable, while people with shorter or more popular names are less identifiable, especially when the attacker possesses inaccurate information.

5. CONCLUSION

In this position paper, we study the privacy vulnerabilities when medical records join with the Web. First, we show that multiple information sources (e.g. social networks and public records) could be utilized by the attackers. With attribution, inference and aggregation attacks, the attacks are capable of reconstructing very comprehensive user profiles, with various types of highly sensitive and private information (e.g. names, phone numbers, birth dates, diseases, lab test results, etc). On the other hand, we show that people are very identifiable if the attackers are equipped with information retrieval and data mining techniques. Even though an attacker only possesses a piece of inaccurate information, he is still highly likely to identify the target with the help of external information sources.

6. REFERENCES

[1] L. A. Adamic and E. Adar. How to search a social network. *Social Networks*, 27(3):187–203, 2005.

[2] G. J. Annas. Hippa regulations: a new era of medical-record privacy. *The New England Journal of Medicine*, 348(15):1486–1490, 2003.

[3] S. Barnes. A privacy paradox: Social networking in the united states. *First Monday*, 11(9), 2007.

[4] J. Caverlee and S. Webb. A large-scale study of myspace: Observations and implications for online social networks. In *Proceedings of the International Conference on Weblogs and Social Media*, 2008.

[5] B.-C. Chen, R. Ramakrishnan, and K. LeFevre. Privacy skyline: Privacy with multidimensional adversarial knowledge. In *VLDB*, pages 770–781, 2007.

[6] K. E. Emam. Heuristics for de-identifying health data. *IEEE Security & Privacy*, 6(4):58–61, 2008.

[7] C.-J. Hsiao, P. C. Beatty, E. S. Hing, D. A. Woodwell, E. A. Rechtsteiner, and J. E. Sisk. Electronic medical record/electronic health record use by office-based physicians: United states, 2008 and preliminary 2009. National Ambulatory Medical Care Survey, Dec 2009.

[8] C. Lampe, N. Ellison, and C. Steinfield. A face(book) in the crowd: social searching vs. social browsing. In *CSCW '06*, 2006.

[9] N. Li, T. Li, and S. Venkatasubramanian. t-closeness: Privacy beyond k-anonymity and l-diversity. In *ICDE*, 2007.

[10] B. Luo and D. Lee. On protecting private information in social networks: A proposal. In *ICDE Workshop on Modeling, Managing, and Mining of Evolving Social Networks (M3SN)*, 2009.

[11] A. Machanavajjhala, D. Kifer, J. Gehrke, and M. Venkitasubramaniam. *l*-diversity: Privacy beyond *k*-anonymity. *TKDD*, 1(1), 2007.

[12] D. J. Martin, D. Kifer, A. Machanavajjhala, J. Gehrke, and J. Y. Halpern. Worst-case background knowledge for privacy-preserving data publishing. In *ICDE*, pages 126–135, 2007.

[13] S. Preibusch, B. Hoser, S. Gürses, and B. Berendt. Ubiquitous social networks - opportunities and challenges for privacy-aware user modelling. In *Proceedings of Workshop on Data Mining for User Modeling*, 2007.

[14] C. Safran, M. Bloomrosen, and W. H. et.al. Toward a national framework for the secondary use of health data: an american medical informatics association white paper. *Journal of American Medical Informatics Association*, 2007.

[15] L. Sweeney. k-anonymity: A model for protecting privacy. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 10(5):557–570, 2002.

[16] C. Valli. The insider threat to medical records: Has the network age changed anything? In *SAM 2006*. CSREA, 2006.

[17] R. A. Wagner and M. J. Fischer. The string-to-string correction problem. *J. ACM*, 21(1):168–173, 1974.

[18] P. A. H. Williams. The underestimation of threats to patient data in clinical practice. In *3rd Australian Information Security Management Conference*, pages 117–122, Perth, WA, 2005.