8th International Workshop on Data Mining in Bioinformatics (BIOKDD 2008)

Held in conjunction with SIGKDD conference, August 24, 2008

KDD08
LAS VEGAS | 24-27 AUGUST 2008
The 14th ACM SIGKDD International Conference
on Knowledge Discovery & Data Mining

Workshop Chairs

Stefano Lonardi
Jake Y. Chen
Mohammed Zaki

BIOKDD ‘08: 2008 International Workshop on Data Mining in Bioinformatics Las Vegas, NV, USA

Held in conjunction with
14th ACM SIGKDD Conference on Knowledge Discovery and Data Mining

Stefano Lonardi
Dept. of Computer Science and Eng.
University of California
Riverside, CA 92521
stelo@cs.ucr.edu

Jake Y. Chen
School of Informatics
Indiana University
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 study of collecting, managing, interpreting, and disseminating biological data and knowledge. 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 new and exciting opportunities for data mining researchers in the post-genome era. 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 molecular pathway mapping, are still open.

Data mining approaches seem ideally suited for bioinformatics, because it can help researchers sift through large amounts of data to develop novel biological insights not obvious from conventional data analysis. The extensive databases of biological information create both challenges and opportunities for developing novel KDD methods. To highlight these avenues we organized the Workshops on Data Mining in Bioinformatics (BIOKDD 2001-2007), held annually in conjunction with the ACM SIGKDD Conference. This will be the 8th year for the workshop.

The goal of this year’s workshop call for papers (CFP) was to encourage KDD researchers to take on the numerous research challenges that post-genomics biology offers. In our call for papers, we promoted a theme “integrating complex biological systems and knowledge discovery”. 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:

- Phylogenetics and comparative Genomics
- DNA microarray 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

- Computational drug discoveries
- Biomedical text mining
- Biological data management techniques
- Semantic webs and ontology-driven biological data integration methods

PROGRAM

The workshop is a half day event in conjunction with the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Las Vegas, NV, August 24-27, 2008. 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/biokdd08.

This year, we accepted 8 papers out of 24 submissions into the workshop program and proceedings 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 three 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 Philip Yu, Ph.D. Professor and Wexler Chair in Information Technology, Department of Computer Science, University of Illinois at Chicago, Chicago, IL, USA. His talk's title is "Link Mining: exploring the power of links".

WORKSHOP CO-CHAIRS

- Stefano Lonardi, University of California, Riverside
- Jake Y. Chen, Indiana University – Purdue University, Indianapolis
- Mohammed J. Zaki, Rensselaer Polytechnic Institute (General Chair)

PROGRAM COMMITTEE

Alberto Apostolico (Georgia Tech & University of Padova), Ann Loraine (University of North Carolina, Charlotte), Chad Myers (University of Minnesota), Chandan K. Reddy (Wayne State University), Dong Xu (University of Missouri), Giuseppe Lancia (University of Udine, Italy), Isidore Rigoutsos (IBM T. J. Watson Research Center), Jason Wang (New Jersey Institute of Technology), Jie Zheng (NCBI), Jing Li (Case Western Reserve University), Knut Reinert (Freie Universitt Berlin, Germany), Li Liao (University of Delaware), Luke Huan (University of Kansas), Mehmet Koyuturk (University of Georgia), Muhammad Abulaish (Case Western Reserve University), Natasa Przulj (University of California, Irvine), Michael Brudno (University of Toronto), Muhammad Abulaish (Jamia Millia Islamia, India), Natasa Przulj (University of California, Irvine), Phoebe Chen (Deakin University, Australia), Rui Kuang (University of Minnesota), Seungchan Kim (Arizona State University), Si Luo (Purdue University), Simon Lin (Northwestern University), Walid G. Aref (Purdue University), Wei Wang (University of North Carolina, Chapel Hill), Xiaohua Hu, (Drexel University), Yaoqi Zhou (Indiana University), Yves Lussier (University of Chicago).

ACKNOWLEDGEMENT

We would like to thank all the program committee, contributing authors, invited speaker, and attendees for contributing to the success of the workshop. Special thanks are also extended to the SIGKDD '08 conference organizing committee, particularly Eamonn Keogh, for coordinating with us to put together the excellent workshop program on schedule.

WORKSHOP SCHEDULE AND INDEX TO PROCEEDING

8:20-8:30am: **Opening Remarks**

Session 1.

8:30-8:50am: **Talk 1:** *Function Prediction Using Neighborhood Patterns* **page 4-10**

- Petko Bogdanov and Ambuj K. Singh, University of California, Santa Barbara, USA

8:50-9:10am: **Talk 2:** *Statistical Modeling of Medical Indexing Processes for Biomedical Knowledge Information Discovery from Text* **page 11-19**

- Markus Bundschus, Mathaeus Dejori, Shipeng Yu, Volker Tresp, and Hans-Peter Kriegel, University of Munich, Germany

9:10-9:30am: **Talk 3:** *Information Theoretic Methods for Detecting Multiple Loci Associated with Complex Diseases* **page 20-28**

- Pritam Chanda, Aidong Zhang, Lara Sucheston and Murali Ramanathan, State University of New York, Buffalo, USA

9:30-9:50am: **Talk 4:** *A Fast, Large-scale Learning Method for Protein Sequence Classification* **page 29-37**

- Pavel Kuksa, Pai-Hsi Huang, and Vladimir Pavlovic, Rutgers University, USA

9:50-10:05am: **Coffee Break**

Session 2.

10:05-10:50am: **Keynote Talk:** *Link Mining: exploring the power of links*

- Philip Yu, University of Illinois, Chicago, USA

10:50-11:10am: **Talk 5:** *Catching Old Influenza Virus with A New Markov Model* **page 38-43**

- HamChing Lam and Daniel Boley, University of Minnesota, USA

11:10-11:30am: **Talk 6:** *GPD: A Graph Pattern Diffusion Kernel for Accurate Graph Classification with Applications in Cheminformatics* **page 44-52**

- Aaron Smalter, Luke Huan, Gerald Lushington and Yi Jia, University of Kansas, USA

11:30-11:50am: **Talk 7:** *Reinforcing Mutual Information-based Strategy for Feature Selection for Microarray Data* **page 53-60**

- Jian Tang, Shuigeng Zhou, Feng Li and Jiang Kai, Fudan University, China

11:50-12:10pm: **Talk 8:** *Graph-based Temporal Mining of Metabolic Pathways with Microarray Data* **page 61-70**

- Chang hun You, Lawrence B. Holder, Diane J. Cook, Washington State University, USA

12:10-12:20pm: **Concluding Remarks**

Function Prediction Using Neighborhood Patterns[*]

Petko Bogdanov[†]
Department of Computer Science, University of California, Santa Barbara, CA 93106
petko@cs.ucsb.edu

Ambuj Singh
Department of Computer Science, University of California, Santa Barbara, CA 93106
ambuj@cs.ucsb.edu

ABSTRACT

The recent advent of high throughput methods has generated large amounts of protein interaction data. This has allowed the construction of genome-wide networks. A significant number of proteins in such networks remain uncharacterized and predicting the function of these proteins remains a major challenge. A number of existing techniques assume that proteins with similar functions are topologically close in the network. Our hypothesis is that proteins with similar functions observe similar annotation patterns in their neighborhood, regardless of the distance between them in the interaction network. We thus predict functions of uncharacterized proteins by comparing their functional neighborhoods to proteins of known function. We propose a two-phase approach. First we extract functional neighborhood features of a protein using *Random Walks with Restarts*. We then employ a kNN classifier to predict the function of uncharacterized proteins based on the computed neighborhood features. We perform leave-one-out validation experiments on two *S. cerevisiae* interaction networks revealing significant improvements over previous techniques. Our technique also provides a natural control of the trade-off between accuracy and coverage of prediction.

Categories and Subject Descriptors

I.5 [**Pattern Recognition**]: Applications

General Terms

Methodology

Keywords

Protein Function Prediction, Feature Extraction, Classification, Protein Interaction Network

1. INTRODUCTION

The rapid development of genomics and proteomics has generated an unprecedented amount of data for multiple model organisms. As has been commonly realized, the acquisition of data is but a preliminary step, and a true challenge lies in developing effective means to analyze such data and endow them with physical or functional meaning [24]. The problem of function prediction of newly discovered genes has traditionally been approached using sequence/structure homology coupled with manual verification in the wet lab. The first step, referred to as computational function prediction, facilitates the functional annotation by directing the experimental design to a narrow set of possible annotations for unstudied proteins.

Significant amount of data used for computational function prediction is produced by high-throughput techniques. Methods like Microarray co-expression analysis and Yeast2Hybrid experiments have allowed the construction of large interaction networks. A protein interaction network (PIN) consists of nodes representing proteins, and edges representing interactions between proteins. Such networks are stochastic as edges are weighted with the probability of interaction. There is more information in a PIN compared to sequence or structure alone. A network provides a global view of the context of each gene/protein. Hence, the next stage of computational function prediction is characterized by the use of a protein's interaction context within the network to predict its functions.

A node in a PIN is annotated with one or more functional terms. Multiple and sometimes unrelated annotations can occur due to multiple active binding sites or possibly multiple stable tertiary conformations of a protein. The annotation terms are commonly based on an ontology. A major effort in this direction is the Gene Ontology (GO) project [11]. GO characterizes proteins in three major aspects: *molecular function, biological process* and *cellular localization*. Molecular functions describe activities performed by individual gene products and sometimes by a group of gene products. Biological processes organize groups of interactions into "ordered assemblies." They are easier to predict since they localize in the network. In this paper, we seek to predict the GO molecular functions for uncharacterized (target) proteins.

The main idea behind our function prediction technique is that function inference using only local network analysis but without the examination of global patterns is not general enough to cover all

[*]
BIOKDD '08, August 24, 2008, Las Vegas, Nevada, USA.
Copyright 2007 ACM 978-1-60558-302-0 ...$5.00.

[†] Corresponding author.

possible annotation trends that emerge in a PIN. Accordingly, we divide the task of prediction into the following sequence of steps: extraction of neighborhood features, accumulation and categorization of the neighborhood features from the entire network, and prediction of the function of a target protein based on a classifier. We summarize the neighborhood of a protein using *Random Walks with Restarts*. Coupled with annotations on proteins, this allows the extraction of histograms (on annotations) that serve as our features. We perform a comprehensive set of experiments that reveal a significant improvement of prediction accuracy compared to existing techniques.

The rest of the paper is organized as follows. Section 2 discusses the related work. Section 3 presents our methods. In Section 4, we present experimental results on two *S. cerevisiae* interaction networks, and conclude in Section 5.

2. RELATED WORK

According to a recent survey [22], most existing network-based function prediction methods can be classified in two groups: *module assisted* and *direct methods*. Module assisted methods detect network modules and then perform a module-wide annotation enrichment [16]. The methods in this group differ in the manner they identify modules. Some use graph clustering [23, 10] while others use hierarchical clustering based on network distance [16, 2, 4], common interactors [20] and Markov random fields [15].

Direct methods assume that neighboring proteins in the network have similar functional annotations. The *Majority* method [21] predicts the three prevailing annotations among the direct interactors of a target protein. This idea has later been generalized to higher levels in the network [13]. Another approach, *Indirect Neighbor* [7], distinguishes between direct and indirect functional associations, considering level 1 and level 2 associations. The *Functional Flow* method [19] simulates a network flow of annotations from annotated proteins to target ones. Karaoz et al. [14] propose an annotation technique that maximizes edges between proteins with the same function.

A common drawback of both the direct and module-assisted methods is their hypothesis that proteins with similar functions are always topologically close in the network. As we show, not all proteins in actual protein networks corroborate this hypothesis. The direct methods are further limited to utilize information about neighbors up to a certain level. Thus, they are unable to predict the functions of proteins surrounded by unannotated interaction partners.

A recent approach by Barutcuoglu et al. [3] formulates the function prediction as a classification problem with classes from the GO biological process hierarchy. The authors build a Bayesian framework to combine the scores from multiple Support Vector Machine (SVM) classifiers.

A technique called *LaMoFinder* [6] predicts annotations based on network motifs. An unannotated network is first mined for conserved and unique structural patterns called motifs. The motifs are next labeled with functions. Pairs of corresponding proteins in different motif occurrences are expected to have similar annotations. The method is restricted to target proteins that are part of unique and frequent structural motifs. A less conservative approach for pattern extraction (that is robust to noise in network topology) is needed for the task of whole genome annotation.

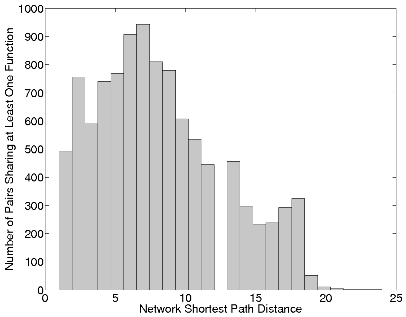


Figure 1: Proteins sharing annotations do not always interact in the *Filtered Yeast Interactome (FYI)* [12]. Similar functions are sometimes at large network distances.

We hypothesize that the simultaneous activity of sometimes functionally diverse functional agents comprise higher level processes in different regions of the PIN. We refer to this hypothesis as *Similar Neighborhood*, and to the central idea in all direct methods as *Function Clustering*. Our hypothesis is more general, since a clique of similar function proteins can be equivalently treated as a set of nodes that observe the same functional neighborhood. Hence *Similar Neighborhood* is a natural generalization of *Function Clustering*. A justification for our approach is provided by Figure 1 which shows that proteins of similar function may occur at large network distances.

3. METHOD

Our approach divides function prediction into two steps: extraction of neighborhood features, and prediction based on the features. According to our *Similar Neighborhood* hypothesis, we summarize the functional network context of a target protein in the neighborhood features extraction step. We compute the steady state distribution of a *Random Walk with Restarts (RWR)* from the protein. The steady state is then transformed into a functional profile. In the second step, we employ a *k-Nearest-Neighbors (kNN)* classifier to predict the function of a target protein based on its functional profile. As confirmed by the experimental results, the desired tradeoff between accuracy of prediction and coverage of our algorithm can be controlled by k, the only parameter of the kNN classification scheme. Such a decoupled approach allows for the possibility that other kinds of neighborhood features can be extracted, and that other kinds of classifiers can be used.

3.1 Extraction of functional profiles

The extraction of features is performed in two steps. First, we characterize the neighborhood of a target node with respect to all other nodes in the network. Second, we transform this node-based characterization to a function-based one.

We summarize a protein's neighborhood by computing the steady state distribution of a *Random Walk with Restarts (RWR)*. We simulate the trajectory of a random walker that starts from the target protein and moves to its neighbors with a probability proportional to the weight of each connecting edge. We keep the random walker close to the original node in order to explore its local neighborhood, by allowing transitions to the original node with a probability of r, the restart probability [5].

The PIN graph is represented by its adjacency matrix $M_{n,n}$. Each element $m_{i,j}$ of M encodes the probability of interaction between proteins i and j. The outgoing edge probabilities of a each protein are normalized, i.e. M is row-normalized. We use the power method to compute the steady state vector with respect to each node. We term the steady state distribution of node j as the *neighborhood profile* of protein j, and denote it as $S^j, j \in [1, n]$. The neighborhood profile is a vector of probabilities $S^j_i, i \neq j, i, j \in [1, n]$. Component S^j_i is proportional to the frequency of visits to node i in the RWR from j. More formally, the power method is defined as follows:

$$S^j(t+1) = (1-r)M^T S^j(t) + rX. \tag{1}$$

In the above equation, X is a size-n vector defining the initial state of the random walk. In the above scenario, X has only one nonzero element corresponding to the target node. $S^j(t)$ is the neighborhood profile after t time steps. The final neighborhood profile is the vector S^j when the process converges. A possible interpretation of the neighborhood profile is an affinity vector of the target node to all other nodes based solely on the network structure.

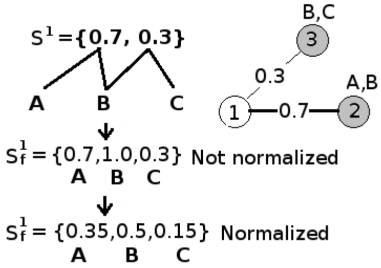

S¹={0.7, 0.3}
A
B
C
S¹f = {0.7,1.0,0.3} Not normalized
A B C
S¹f = {0.35,0.5,0.15} Normalized
A B C
B,C
3
0.3
A,B
1
0.7
2

Figure 2: Transformation of the neighborhood profile of node 1 into a functional profile. Node 2 is annotated with functions A and B and node 3 is annotated with functions B and C. The neighborhood profile of node 1 is computed and transformed using the annotations on the nodes into a functional profile.

As our goal is to capture the functional context of a protein, the next step in our feature extraction is the transformation of a neighborhood profile into a functional profile. The value S^j_i of node j to node i can be treated as affinity to the annotations of i. Figure 2 illustrates the transformation of a neighborhood profile to a functional profile. Assume that RWR performed from node 1 results in the neighborhood profile $(0.7, 0.3)$, where 0.7 corresponds to node 2, and 0.3 to node 3. Annotations on these two nodes are weighted by the corresponding values, resulting in the vector $(0.7, 1.0, 0.3)$ over functions A, B, and C, respectively. This vector is then normalized, resulting into the functional profile $(0.35, 0.5, 0.15)$.

More formally, based on the annotations of a protein, we define an annotation flag e_{ia} that equals 1 if protein i is annotated with function a and 0 otherwise. The affinity to each function a in the neighborhood profile is computed as:

$$S^j_f(a) = \sum_{i=1, i \neq j}^{n} S^j_i e_{ia}. \tag{2}$$

Vector S^j_f is normalized to yield the functional profile for node j.

3.2 Function prediction by nearest neighbor classification

The second step in our approach is predicting the annotations of a given protein based on its *functional profile*. According to our *Similar Neighborhood* hypothesis, proteins with similar functional profiles are expected to have similar annotations. An intuitive approach in this setting is to annotate a target protein with the annotations of the protein with most similar neighborhood. Alternatively, we can explore the top k similar proteins to a target protein and compute a consensus set of candidate functions.

We formulate function prediction as a multi-class classification problem. Each protein's profile is an instance (feature vector). Each instance can belong to one or more classes as some proteins have multiple functions. We choose a distance based classification approach to the problem, namely the k-Nearest-Neighbor (kNN) classifier. The classifier uses the L1 distance between the instances and classifies an instance based on the distributions of classes in its k nearest L1 neighbors.

The consensus set of predicted labels is computed using weighted voting. Annotations of a more similar neighborhood are weighted higher. The result is a set of scores for each function where a function's score is computed as follows:

$$F^j_a = \sum_{i=1}^{k} f(d(i,j)) e_{ia}, \tag{3}$$

where e_{ia} is an indicator value set to 1 if protein i is annotated with a, $d(j, i)$ is the distance between functional profiles of proteins i and j and $f(d(i, j))$ is a function that transforms the distance to score. We use a distance-decreasing function of the form $f(d) = \frac{1}{1+\alpha d}, \alpha = 1$. It has the desirable property of a finite maximum at 1 for $d = 0$, and anti-monotonicity with respect to d. As our experiments show, the accuracy did not change significantly when alternative distance transform functions are used.

It is worth mentioning that since the two steps of our approach are completely independent, different approaches can be adopted for feature extraction and classification. Additionally, it is possible to exploit possible dependencies between the dimensions of the functional profile for the purposes of dimensionality reduction.

4. EXPERIMENTAL RESULTS

4.1 Interaction and annotation data

We measure the performance of our method on two yeast protein interaction networks. As a high confidence interaction network, we use the *Filtered Yeast Interactome (FYI)* from [12]. This network is created by using a collection of interaction data sources, including high throughput yeast two-hybrid, affinity purification and mass spectrometry, *in silico* computational predictions of interactions, and interaction complexes from MIPS [18]. The network contains 1379 proteins and 1450 interactions. *FYI* is an unweighted network, since every edge is added if it exists in more than two sources [12]. When performing the random walk on this network, the walker follows a uniformly chosen edge among the outgoing edges.

The second yeast interaction network is constructed by combining *9* interaction data sources from the *BioGRID* [1] repository. The method of construction is similar to the ones used in [7, 19, 17]. The network consists of 4914 proteins and 17815 interactions among them. The *BioGRID* network contains weighted edges based on scoring that takes into account the confidence in each data source and the magnitude of the interaction.

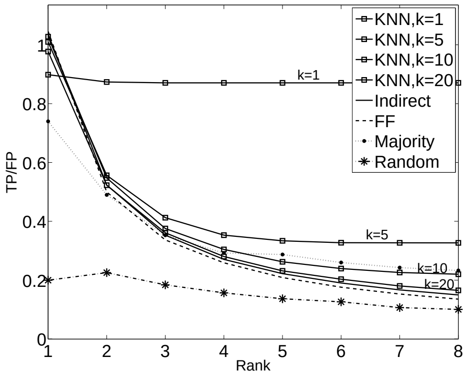
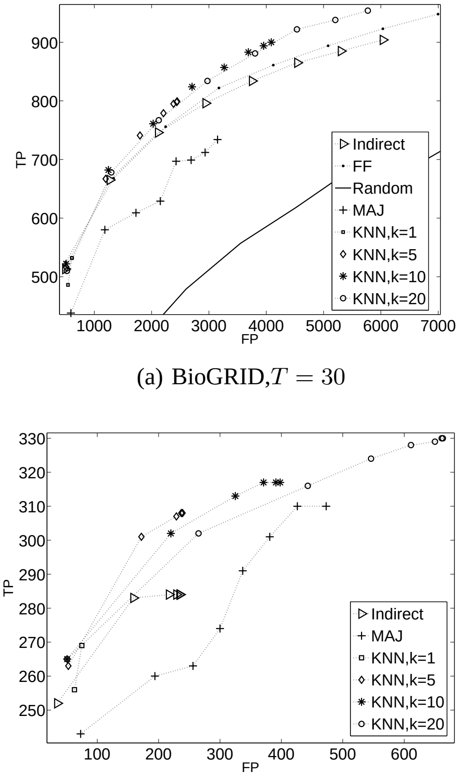


Figure 3: TP/FP ratio for the *BioGRID* network. All genes are labeled with exactly one annotation and the value of the frequency threshold is $T = 30$.



(a) BioGRID,$T = 30$

(b) FYI,$T = 20$

Figure 4: TP versus FP for the (a) *BioGRID* and (b) *FYI* networks. All genes are labeled with exactly one annotation and the frequency thresholds are set respectively to 30 and 20.

The protein GO annotations for *S. cerevisiae* gene products were obtained from the Yeast Genome Repository [9].

4.2 Existing techniques

We compare our *KNN* technique to *Majority (MAJ)* [21], *Functional Flow (FF)* [19] and *Indirect Neighbors (Indirect)* [7]. Majority scores each candidate function based on the number of its occurrences in the direct interactors. The scores of candidate functions in edge-weighted networks can be weighted by the probabilities of the connecting edges. Functional Flow [19] simulates a discrete-time flow of annotations from all nodes. At every time step, the annotation weight transferred along an edge is proportional to the edge's weight and the direction of transfer is determined by the difference of the annotation's weight in the adjacent nodes. The Indirect [7] method exploits both indirect and direct function associations. It computes *Functional Similarity* score based on *level 1* and *level 2* interaction partners of a protein. We used the implementation of the method as supplied by the authors, with weight function: FSWEIGHT and with minor changes related to the selection of informative functional annotations.

4.3 Experimental setup

The frequency of a functional annotation (class) is the number of proteins that are annotated with it. We call functions whose frequency exceeds a given threshold T as *informative*. An informative instance is a protein (represented by its functional profile) annotated with at least one informative class. For a given T, our training instance set contains all informative instances in the network. We exploit all available annotation information and predict functions at different levels of specificity. Unlike the approach in [8], we predict informative functions, even if their descendants are also informative.

We compare the accuracy of the techniques by performing leave-one-out validation experiments. We use leave-one-out validation because many annotations in the actual network are of relatively low frequency, and thus limiting the training set. Our classifier is working with actual networks, containing significant number of uncharacterized proteins and hence this is a realistic measure of the accuracy. Moreover, since the competing techniques implicitly use all available annoatations, leave-one-out provides a fair comparison to our method. In this setup, a target protein is held out (i.e. its annotations are considered unknown) and a prediction is computed using the rest of the annotation information in the network. All competing methods compute a score distribution for every class. We use the scores to rank the candidate functions and then analyze the accuracy for different ranks. An ideal technique would rank the true (held-out) annotation(s) as the top-most one. We penalize a technique for ranking false annotations above the actual ones. Additionally, we do not consider functions of zero score as actual predictions of the techniques.

A true positive (TP) prediction is a protein predicted as its actual label or any of the label's ontological descendants. This is also known as the *true path* prediction criterion and has been used in previous ontology-aware prediction studies [8]. The motivation for the true path criterion is the gradual characterization of a given protein with more specific terms as more wet-lab experiments are performed. We analogously define a false positive (FP) prediction as a prediction of a function that is not part of the annotation of a target protein.

Though we pose the annotation prediction as a classification problem, it is not a general classification task. A domain scientist would be more interested in the TP and FP predictions, than in the number of True Negatives (TN) and False Negatives (FN). TNs in the prediction setting cannot facilitate the wet-lab experiments since the space of all possible functions is large, hence characterizing a protein using positive predictions is more tractable compared to using

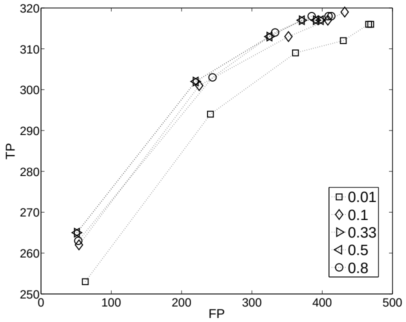


Figure 5: The effect of different restart probabilities r on the accuracy in the *FYI* network $(T = 20, k = 10)$.

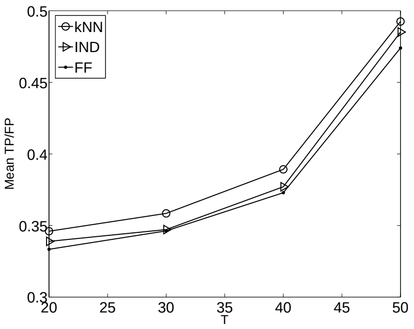


Figure 6: Effect of the informative functions threshold T. The average TP/FP ratio for each of the first 4 ranks is plotted for the *BioGRID* network. All genes are labeled with exactly one annotation and the value of k is set to 10.

negative ones.

The Receiver Operating Characteristic (ROC) is a commonly used metric for comparison of competing classification techniques. In an ROC setting, the True Positive Rate (TPR = TP/P) is plotted as a function of the False Positive Rate (FPR = FP/N). We show a variation of the ROC that skips the normalization terms, so that the actual number of false predictions is explicit in the plots.

4.4 Effect of parameters k, r, T and the distance conversion function $f(d)$

We first analyze the effect of the number of neighbors k in our kNN technique on the accuracy of the method. Figure 3 presents the ratio of TP/FP (accuracy ratio) as a function of the rank up to which labels are considered as predictions. We analyze this statistic for four different values of k and all competing techniques. We also examine the performance of a random predictor (*Random*) that uses solely the prior probabilities of the annotations.

The highest rank prediction for most of the methods produces roughly equal number of TP and FP. Compared to a random model, this accuracy is significantly higher as there are 18 candidate labels in this specific experiment ($T = 30$). The average number of classes for which 1NN gives predictions, i.e. classes that score greater than 0, is 1.2, hence the accuracy ratio of 1NN remains fairly stable for increasing ranks. The number of predictions increases with k and more TP are discovered at the price of more FP.

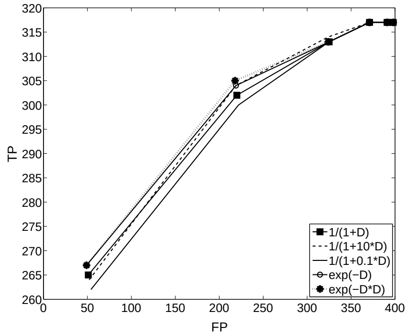


Figure 7: Effect of the distance to score conversion function on the accuracy of kNN. Our method is not sensitive to the exact form of the function (*FYI*, $T = 20$, $k = 10$).

A good predictor needs to be balanced with respect to its accuracy ratio and coverage, where the coverage is defined as the number of TP for increasing ranks, regardless of the FP introduced. According to this definition, kNN for small values of k can be regarded as high accuracy and low coverage method, and with increasing k, the coverage is increased at the price of lower accuracy. This effect can be observed in Figure 4(a). As k increases, the curves become less steep, however coverage improves. This effect of k is even more evident for the high confidence FYI network (Figure 4(b)). Traces for the FF and Random predictor are omitted for the FYI network for clarity since their performance is significantly dominated by the rest of the techniques.

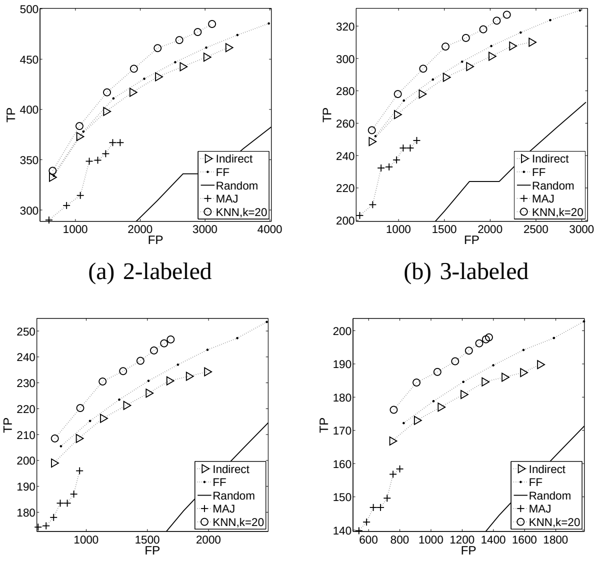
We next study the effect of the restart probability of the Random Walks on the quality of the functional neighborhoods. As evident from Figure 5, the classification accuracy is not sensitive to the value of restart, as long as it is not chosen extremely low or extremely high. Values of 0.5 and 0.33 result in identical performance. Hence for all experiments we use a restart value of 0.33.

The overall relative performance of the techniques for varying informative threshold T is presented in Figure 6. We vary T from 20 to 50 for the *BioGRID* network and compare the average accuracy ratio of the first four ranks. Our technique dominates for all values of T. Note that when predicting low frequency classes, a lower value for k would result in a better prediction accuracy. However, for this specific experiment, we use a uniform value of $k = 10$ for all T.

We experiment with different distance conversion functions $f(d)$ in order to assess the sensitivity of our method to this parameter. The accuracy for three versions of our fractional function $f(d) = \frac{1}{1+\alpha d}$, $\alpha = 0.1, 1, 10$ as well as two exponential functions e^{-d} and e^{-d^2} are presented in Figure 7. Our method is not sensitive to the exact form of function, however we do not exclude the possibility of learning the optimal function for a given dataset.

4.5 Prediction accuracy

The prediction accuracy for single-labeled proteins is presented in Figures 4(a) and 4(b). As we already discussed, kNN outperforms the competing techniques when predicting single classes.



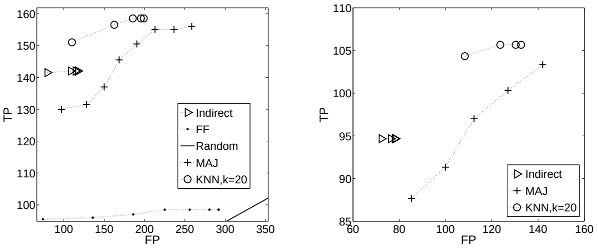
(a) 2-labeled

(b) 3-labeled

(c) 4-labeled

(d) 5-labeled

Figure 8: Performance comparison on the *BioGRID* network for (a) 2-, (b) 3-, (c) 4- and (d) 5-labeled proteins, $T = 30$.



(a) 2-labeled

(b) 3-labeled

Figure 9: Performance comparison on the *FYI* network for (a) 2- and (b) 3-labeled proteins, $T = 20$.

A significant number of proteins in most genomes perform several functions, and hence have multiple annotations. We thus would like to analyze the performance of our technique on multi-labeled proteins. In this experiment, we group the proteins by the cardinality of their label set and perform leave-one-out validation. In this case every TP label is counted as a TP/C fraction of a true positive, where C is the cardinality of the label set. We take a similar approach when counting the false positives. In this set of experiments, we vary the rank up to which a label is considered predicted starting from C. Figures 8(a)-8(d) present the accuracy curves for proteins labeled with two and more annotations in the *BioGRID* network. The difference in performance between our method and competing methods is preserved when predicting more than one label. Similar plots are shown for the small *FYI* network in Figures 9(a), 9(b).

4.6 Discussion

The semantics of both GO *process* and *localization* imply that same terms would interact and hence cluster in a PIN. According to its definition, a GO *process* is a conglomerate of GO *functions* performed in a sequence. Genes localized in the same compartment of the cell, i.e. share GO *localization* terms, are also expected to interact more than ones of different localization. On the contrary, a GO *function* describes a molecular activity without specifying where, when or in what context this activity takes place. An example of two *Kinases* interacting with *GTPases* and *Unfolded Protein Binding* genes is presented in Figure 10. They share a functionally diverse pattern in their neighborhood, which could be captured by our feature extraction step.

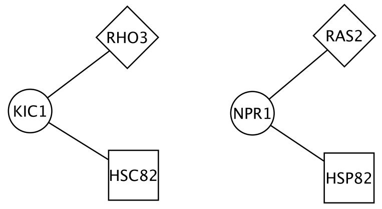


Figure 10: An example of two *Kinases* KIC1 and NRP1 that both interact with *GTPases* (RHO3 and RAS2) and *Unfolded Protein Binding* genes (HSC82 and HSP82)

We further analyzed the annotations of the three GO components in the high confidence *FYI* network. We call a label *related* to a target label if it is the same or any of the target's ontological ancestors. More than 73% of the *localization* and 64.2% of the *process* annotations interact with more related annotations than unrelated ones. This percentage for the *function* hierarchy is only 58%. The semantic uniqueness of the GO function hierarchy makes it harder for *Direct methods* to infer uncharacterized proteins and this is why we concentrate on this specific part of GO. Our experiments on *process* and *localization* did not reveal a significant advantage of our method over existing ones.

Our method is robust to the density of the interaction network. This is demonstrated by the consistent accuracy dominance of our technique over the competing ones on two yeast interaction networks of different size, density and origin. A possible explanation for the robustness is the preservation of the neighborhood pattens in networks of diverse size and origin, which we think is a promising further direction for exploration.

5. CONCLUSION

We proposed a novel framework for predicting functional annotation in the context of protein interaction network. It is comprised of two independent components: extraction of neighborhood features and prediction (formulated as classification) based on these features. The only parameter k to which our approach is sensitive provides an intuitive interface for control of the trade-off between accuracy and coverage of our method. Our method is robust to the density and size of a PIN and its prediction accuracy is higher than that of previous methods.

The predictive power of our method gives further insight about the topological structure of functional annotations in a genome-wide network. The commonly adopted idea that similar functions are network neighbors does not hold for all annotations. A different structural annotation trend emerges, namely functions that observe similar (but sometimes heterogeneous) functional neighborhoods. Our approach incorporates this idea and has a better predictive power.

6. ACKNOWLEDGMENTS

This work is supported in part by National Science Foundation IIS-0612327. We thank Prof. Limsoon Wong for providing the implementation of the *Indirect Neighbor* [7] technique.

7. REFERENCES

[1] Biogrid: General repository for interaction datasets. *http://www.thebiogrid.org/*, 2006.

[2] V. Arnau, S. Mars, and I. Marin. Iterative clustering analysis of protein interaction data. *Bioinformatics*, 2005.

[3] Z. Barutcuoglu, R. Schapire, and O. Troyanskaya. Hierarchical multi-label prediction of gene function. *Bioinformatics*, 2006.

[4] C. Brun, F. Chevenet, D. Martin, J. Wojcik, A. Guenoche, and B. Jacq. Functional classification of proteins for the prediction of cellular function from a protein-protein interaction network. *Genome Biology*, 5:R6, 2003.

[5] T. Can, O. Camoglu, and A. K. Singh. Analysis of protein interaction networks using random walks. *Proceedings of the 5th ACM SIGKDD Workshop on Data Mining in Bioinformatics*, 2005.

[6] J. Chen, W. Hsu, M. L. Lee, and S.-K. Ng. Labeling network motifs in protein interactomes for protein function prediction. *ICDE*, 2007.

[7] H. Chua, W. Sung, and L. Wong. Exploiting indirect neighbors and topological weight to predict protein function from protein-protein interactions. *Bioinformatics*, 2006.

[8] H. Chua, W. Sung, and L. Wong. Using indirect protein interactions for the prediction of gene ontology functions. *BMC Bioinformatics*, 2007.

[9] Y. G. Database. *http://www.yeastgenome.org/*.

[10] R. Dunn, F. Dudbridge, and C. Sanderson. The use of edge-betweenness clustering to investigate the biological function in protein interaction networks. *BMC Bioinformatics*, 2005.

[11] T. gene ontology consortium. Gene ontology: Tool for the unification of biology. *Nature*, 2000.

[12] J. Han, N. Bertin, and T. H. et Al. Evidence for dynamically organized modularity in the yeast protein-protein interaction network. *Nature*, 2004.

[13] H. Hishigaki, K. Nakai, T. Ono, A. Tanigami, and T. Takagi. Assessment of prediction accuracy of protein function from protein–protein interaction data. *Yeast*, 2001.

[14] U. Karaoz, T. M. Murali, S. Letovsky, Y. Zheng, C. Ding, C. R. Cantor, and S. Kasif. Whole-genome annotation by using evidence integration in functional-linkage networks. *PNAS*, 101:2888–2893, 2004.

[15] S. Letovsky and S. Kasif. Predicting protein function from protein/protein interaction data: a probabilistic approach. *Bioinformatics*, 19:i197–i204, 2003.

[16] K. Maciag, S. Altschuler, M. Slack, N. Krogan, A. Emili, J. Greenblatt, T. Maniatis, and L. Wu. Systems-level analyses identify extensive coupling among gene expression machines. *Molecular Systems Biology*, 2006.

[17] C. V. Mering, M. Huynen, D. Jaeggi, S. Schmidt, P. Bork, and B. Snel. String: a database of predicted functional associations between proteins. *Nucleic Acids Res.*, 2003.

[18] H. W. Mewes, D. Frishman, U. Guldener, G. Mannhaupt, K. Mayer, M. Mokrejs, B. Morgenstern, M. Munsterkotter, S. Rudd, and B. Weil. Mips: a database for genomes and protein sequences. *Nucleic Acids Res.*, 30:31–34, 2002.

[19] E. Nabieva, K. Jim, A. Agarwal, B. Chazelle, and M. Singh. Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. *Bioinformatics*, 21:i302–i310, 2005.

[20] M. P. Samanta and S. Liang. Predicting protein functions from redundancies in large-scale protein interaction networks. *PNAS*, 100:12579–12583, 2003.

[21] B. Schwikowski, P. Uetz, and S. Fields. A network of protein-protein interactions in yeast. *Nature*, 2000.

[22] R. Sharan, I. Ulitsky, and R. Shamir. Network-based prediction of protein function. *Molecular Systems Biology*, 2007.

[23] V. Spirin and L. Mirny. Protein complexes and functional modules in molecular networks. *PNAS*, 2003.

[24] G. X. Yu, E. M. Glass, N. T. Karonis, and N. Maltsev. Knowledge-based voting algorithm for automated protein functional annotation. *PROTEINS: Structure, Function, and Bioinformatics*, 61:907–917, 2005.

Statistical modeling of medical indexing processes for biomedical knowledge information discovery from text

Markus Bundschus
Institute for Computer Science
University of Munich
Oettingenstr. 67
80538 Munich, Germany
bundschu@dbs.ifi.lmu.de

Mathaeus Dejori
Integrated Data Systems Dep.
Siemens Corporate Research
755 College Road East
Princeton, NJ 08540, USA
mathaeus.dejori@siemens.com

Shipeng Yu
CAD & Knowledge Solutions
Siemens Medical Solutions
51 Valley Stream Parkway
Malvern, PA 19355, USA
shipeng.yu@siemens.com

Volker Tresp
Information &
Communications, IC4
Siemens CT
Otto-Hahn-Ring 6
81739 Munich, Germany
volker.tresp@siemens.com

Hans-Peter Kriegel
Institute for Computer Science
University of Munich
Oettingenstr. 67
80538 Munich, Germany
kriegel@dbs.ifi.lmu.de

ABSTRACT

The overwhelming amount of published literature in the biomedical domain and the growing number of collaborations across scientific disciplines results in an increasing topical complexity of research articles. This represents an immense challenge for efficient biomedical knowledge discovery from text. We present a new graphical model, the so-called Topic-Concept Model, which extends the basic Latent Dirichlet Allocation framework and reflects the generative process of indexing a PubMed abstract with terminological concepts from an ontology. The generative model captures the latent topic structure of documents by learning the statistical dependencies between words, topics and MeSH (Medical Subject Headings) concepts. A number of important tasks for biomedical knowledge discovery can be solved with the here introduced model. We provide results for the extraction of the hidden topic-concept structure from a large medical text collection, the identification of the most likely topics given a specific MeSH concept, and the extraction of statistical relationships between MeSH concepts and words. Moreover, we apply the introduced generative model to a challenging multi-label classification task. A benchmark with several classification methods on two independent data sets proves our method to be competitive.

Keywords

Document Modeling, topic modeling, multi-label classification, ontologies

BIOKDD '08 Las Vegas, NV, USA
Copyright 2008 ACM 978-1-60558-302-0 ...$5.00.

1. INTRODUCTION

In the last decade, powerful new biomedical research tools and methods have been developed, resulting in an unprecedented increase of biomedical data and literature. High-throughput experiments, such as DNA microarrays or protein arrays, produce large quantities of high-quality data, leading to an explosion of scientific articles published in this field. Thus, automated extraction of useful information from large document collections has become an increasingly important research area [12, 11]. To ensure an efficient access to this steadily increasing source of bibliographic information, it is required to efficiently index incoming articles, i. e. to label unstructured free text with a structured machine readable annotation. Articles selected for inclusion in PubMed[1], for example, are indexed with concepts from the Medical Subject Headings[2] (MeSH) thesaurus to facilitate later retrieval. This additional meta information provides a rich source of knowledge, which can be exploited for biomedical knowledge discovery and data mining tasks and this is the focus of this work.

Recently, powerful techniques such as Probabilistic Latent Semantic Analysis (PLSA) [15] or Latent Dirichlet Allocation (LDA) [7] have been proposed for automated extraction of useful information from large document collections. Applications include automatic topic extraction, query answering, document summarization, and trend analysis. Generative statistical models such as the above mentioned ones, have been proven effective in addressing these problems. In general, the following advantages of topic models are highlighted in the context of document modeling: First, topics can be extracted in a complete unsupervised fashion, requiring no initial labeling of the topics. Second, the resulting representation of topics for a document collection is interpretable and last but not least, each document is usually expressed by a mixture of topics, thus capturing the topic combinations that arise in documents [15, 7, 14]. In the

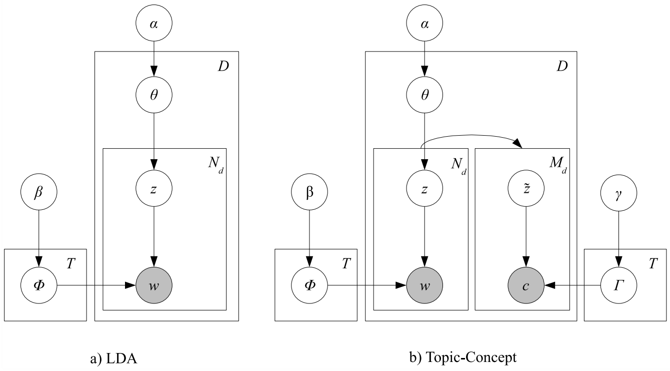
[1] http://www.ncbi.nlm.nih.gov/pubmed/
[2] http://www.nlm.nih.gov/mesh/

biomedical domain, the classical LDA model has been applied to the task of finding life span related genes from the Caenorhabditis Genetic Center Bibliography [5] and to the task of identifying biological concepts from a protein-related corpus [33]. Depending on the addressed generative process, the LDA framework has been extended e. g. to model the dependencies between authors, topics and documents [30] or the dependencies between author and recipients [20]. Further approaches include the modeling of images and their corresponding captions [6] as well as the modeling of dependencies between topics and named entities [25].

In this paper, we introduce another extension of the LDA framework, the so-called Topic-Concept (TC) model, to resemble the generative process of creating an indexed PubMed abstract. The approach simultaneously models the way how the document is generated as well as the way how the document is subsequently indexed with MeSH concepts (see figure 1 for a comparison with the classical LDA approach). We refer to MeSH as a terminological ontology, where relations are partially described as subtype-supertype relations and where the concepts are described by concept labels or synonyms [2].

By modeling the indexing process of PubMed abstracts, we can answer a range of important queries for knowledge discovery about the content of biomedical text collections. With such a model, we can provide a bird's eye view of biomedical topics discussed in a large document collection associated with prominent MeSH concepts (i. e. uncovering the hidden topic-concept structure in a biomedical text collection). In contrast to the classical LDA, this results in a richer representation of topics, since topics are not solely represented by their most likely words. Instead, topics in the TC model are, in addition to the words, associated with their most likely MeSH terms (see section 3.2.1). Furthermore, we can identify several types of statistical relationships between different classes of document entities (i. e. words, MeSH concepts and topics). We provide results for identifying statistical relationships between concepts and words based on the topics (see section 3.2.2). Another interesting use case we consider, is the estimation of the most likely topics given a MeSH concept. This results in a fast overview over the topics in which a specific MeSH term is most likely to be involved (see section 3.2.2). Last but not least, we can use the TC model for multi-label classification. To validate the predictive power of the here presented model, we apply our generative method to a challenging multi-label classification problem with 108 classes. A benchmark on two independent corpora against (1) a multi-label naive Bayes classifier, (2) a method currently used by the National Library of Medicine (NLM) and (3) a state-of-the-art multi-label support vector machine (SVM) shows encouraging results.

The remainder of the paper is organized as follows: In Section 2 we describe the extension of the classical LDA towards the TC model. Section 3.1 describes the experimental setup. Afterwards results are presented and a concluding discussion is given.



a) LDA

b) Topic-Concept

Figure 1: Graphical model for a) LDA and b) Concept-LDA in plate notation. Shaded nodes represent observed random variables, unshaded nodes represent latent random variables.

2. METHODS

In the following we will describe two generative models, the first simulating the process of document generation alone and the second simulating both the process of document generation and the process of document indexing. Let $\mathbf{D} = \{d_1, d_2, ..., d_D\}$ be a set of documents, where D denotes the number of documents in the corpus. A document d is represented by a vector of N_d words, w_d, where each word w_i is chosen from a vocabulary of size N. In the second model, a document d is additionally described by a vector of M_d MeSH concepts c_d, where each concept c_i is chosen from a set of MeSH concepts of size M. The collection of D documents is defined by $\mathbf{D} = \{(w_1, c_1), ..., (w_D, c_D)\}$.

2.1 Classical Latent Dirichlet Allocation (LDA) model

The Latent Dirichlet Allocation model (LDA) is based upon the idea that the probability distribution over words in a document can be expressed as a mixture of topics, where each topic is expressed as a mixture of words [7]. In LDA, the generation of a document collection is modeled as a three step process. First, for each document, a distribution over topics is sampled from a Dirichlet distribution. Second, for each word in the document, a single topic is chosen according to this distribution. Finally, a word is sampled from a multinomial distribution over words specific to the sampled topic. The hierarchical Bayesian model shown (using plate notation) in Figure 1(a) depicts this generative process. θ represents the document-specific multinomial distribution over T topics, each being drawn independently from a symmetric Dirichlet prior α. Φ denotes the multinomial distribution over N vocabulary items for each of T topics being drawn independently from a symmetric Dirichlet prior β. For each of the N_d words w in document d, z denotes the topic responsible for generating that word, drawn from θ, and w is the word itself, drawn from the topic distribution Φ conditioned on z. According to the graphical model representation, the probability distribution over N vocabulary items for the generation of word w_i within a given document is specified as

$$p(w_i) = \sum_{t=1}^{T} p(w_i|z_i = t)p(z_i = t) \qquad (1)$$

where $z_i = t$ represents the assignment of topic t to the ith word, $p(w_i|z_i = t)$ is given by the topic-word distribution Φ and $p(z_i = t)$ by the document-topic distribution θ.

Table 1: Corpora statistics for the two data sets used in this paper.

	random 50K	**genetics-related**
Documents	50.000	84.076
Unique Words	22.531	31.684
Total Words	2.369.616	4.293.992
Unique MeSH Main Headings	17.716	18.350
Total MeSH Main Headings	470.101	912.231

2.2 Extension to the Topic-Concept (TC) Model

The Topic-Concept model extends the LDA framework by simultaneously modeling the generative process of *document generation* and the process of *document indexing.* In addition to the three steps mentioned above, two further steps are introduced to model the process of document indexing. For each of the M_d concepts in the document a topic $\tilde{z}$ is uniformly drawn based on the topic assignments for each word in the document. Finally, each concept c is sampled from a multinomial distribution over concepts specific to the sampled topic. This generative process corresponds to the hierarchical Bayesian model shown in Figure 1(b). In this model, Γ denotes the vector of multinomial distribution over M concepts for each of T topics being drawn independently from a symmetric Dirichlet prior γ. After the generation of words, a topic $\tilde{z}$ is drawn from the document specific distribution, and a concept c is drawn from the $\tilde{z}$ specific distribution Γ. The probability distribution over M MeSH concepts for the generation of a concept c_i within a document is specified as:

$$p(c_i) = \sum_{t=1}^{T} p(c_i|\tilde{z}_i = t)p(\tilde{z}_i = t|\mathbf{z}) \tag{2}$$

where $\tilde{z}_i = t$ represents the assignment of topic t to the ith concept, $p(c_i|\tilde{z}_i = t)$ is given by the concept-topic distribution Γ. The topic for the concept is selected uniformly out off the assignments of topics in the document model, i.e., $p(\tilde{z}_i = t|\mathbf{z}) = \text{Unif}(z_1, z_2, \ldots, z_{N_d})$ leading to a coupling between both generative components.

The generative process of the Topic-Concept model is essentially the same as the Correspondence LDA model proposed in [6] with the difference that the Topic-Concept model imitates the generation of documents and their subsequent annotation, while [7] models the dependency between image regions and captions.

2.3 Learning the Topic-Concept Model from Text Collections

Estimating Φ, θ and Γ provides information about the underlying topic distribution in a corpus and the respective word and MeSH concept distributions in each document. Given the observed documents, the learning task is to infer these parameters for each document. Instead of estimating the parameters directly [16, 6] we follow the idea of [14] and estimate Φ and θ from the posterior distribution over the assignments of words to topics $p(\mathbf{w}|\mathbf{z})$. As the posterior cannot computed directly we resort to a Gibbs sampling strategy generating samples from the posterior by repeatedly drawing a topic for each observed word from its probability conditioned on all other variables. In the LDA model, the algorithm goes over all documents word by word. For each word w_i, a topic z_i is assigned by drawing from its distribution conditioned on all other variables

$$\begin{aligned} p(z_i = t|w_i = n, \mathbf{z_{-i}}, \mathbf{w_{-i}}) &\propto \\ p(w_i = n|z_i = t)p(z_i = t) &\propto \\ \frac{C^{WT}_{nt} + \beta}{\sum_{n'} C^{WT}_{n't} + N\beta} \frac{C^{DT}_{dt} + \alpha}{\sum_{t'} C^{DT}_{dt'} + T\alpha} & \end{aligned} \tag{3}$$

where $z_i = t$ represents the assignments of the ith word in a document to topic t, $w_i = n$ represents the observation that the ith word is the nth word in the lexicon, and $\mathbf{z_{-i}}$ represents all topic assignments not including the ith word. Furthermore, C^{WT}_{nt} is the number of times word n is assigned to topic t, not including the current instance, and C^{DT}_{dt} is the number of times topic t has occurred in document d, not including the current instance. Additionally, in the Topic-Concept model, the posterior $p(c|\tilde{\mathbf{z}})$ is approximated by assigning for each concept c_i, a topic $\tilde{z}_i$ from the following distribution

$$\begin{aligned} p(\tilde{z}_i = t|c_i = m, \tilde{\mathbf{z}}_{\mathbf{i}}, \mathbf{z_{-i}}, \mathbf{w_{-i}}) &\propto \\ p(c_i = m|\tilde{z}_i = t)p(\tilde{z}_i = t|\mathbf{z}) &\propto \\ \frac{C^{CT}_{mt} + \gamma}{\sum_{m'} C^{CT}_{m't} + M\gamma} \frac{C^{TD}_{td}}{N_d} & \end{aligned} \tag{4}$$

where $\tilde{z}_i = t$ represents the assignments of the ith concept in a document to topic t, $c_i = m$ represents the observation that the ith concept in the document is the mth concept in the lexicon, and $\mathbf{z_{-i}}$ represents all topic assignments not including the ith concept. Furthermore, C^{CT}_{mt} is the number of times concept m is assigned to topic t, not including the current instance, and C^{TD}_{td} is the number of times topic t has occurred in document d, not including the current instance.

For any single sample we can estimate Φ, θ and Γ using

$$\hat{\Phi}_{nt} = \frac{C^{WT}_{nt} + \beta}{\sum_{n'} C^{WT}_{n't} + N\beta} \tag{5}$$

$$\hat{\theta}_{dt} = \frac{C^{DT}_{dt} + \alpha}{\sum_{t'} C^{WT}_{dt'} + T\alpha} \tag{6}$$

$$\hat{\Gamma}_{mt} = \frac{C^{CT}_{mt} + \gamma}{\sum_{m'} C^{CT}_{m't} + M\gamma} \tag{7}$$

Instead of estimating the hyperparameters α, β and γ, we fix them to $50/T$, 0.001 and $1/M$ respectively in each of the experiments. The values were chosen according to [30, 14].

3. EXPERIMENTS AND RESULTS

3.1 Experimental setting

Two large PubMed corpora previously generated by [23, 24] were used in the experiments. The first data set is a collection of PubMed abstracts randomly selected from the MEDLINE 2006 baseline database provided by the NLM.

Table 2: Selected topics, learned from the genetics-related corpus ($T = 300$). For each topic the fifteen most probably words and MeSH terms are listed with their corresponding probabilities.

Topic 6				Topic 17			
Word	Prob.	Mesh Term	Prob.	Word	Prob.	Mesh Term	Prob.
ethic	0.043	Humans	0.150	viru	0.118	Humans	0.06
research	0.039	United States	0.038	viral	0.064	HIV-1	0.06
issu	0.023	Informed Consent	0.017	infect	0.058	HIV Infections	0.059
public	0.014	Ethics, Medical	0.011	hiv-1	0.047	Virus Replication	0.045
medic	0.013	Personal Autonomy	0.001	virus	0.035	RNA, Viral	0.042
health	0.013	Decision Making	0.001	hiv	0.033	Animals	0.027
moral	0.013	Ethics, Research	0.008	replic	0.033	DNA, Viral	0.027
consent	0.012	Great Britain	0.008	immunodef.	0.025	Cell-Line	0.023
practic	0.012	Human Experimentation	0.007	envelop	0.012	Genome, Viral	0.022
concern	0.011	Public Policy	0.007	aids	0.012	Viral Proteins	0.020
polici	0.001	Morals	0.007	particl	0.011	Molecular Sequence Data	0.017
conflict	0.008	Biomedical Research	0.006	capsid	0.011	Anti-HIV Agents	0.016
right	0.008	Research Subjects	0.006	host	0.011	Viral Envelope Proteins	0.013
articl	0.008	Social Justice	0.006	infecti	0.010	Drug Resistance, Viral	0.012
accept	0.008	Confidentiality	0.006	antiretrovir	0.001	Acquired Immunodef. Synd.	0.011
Topic 16				**Topic 26**			
Word	Prob.	Mesh Term	Prob.	Word	Prob.	Mesh Term	Prob.
phosphoryl	0.130	Phosphorylation	0.123	breast	0.372	Breast Neoplasms	0.319
kinas	0.118	Prot.-Serine-Threonine Kin.	0.075	cancer	0.323	Humans	0.120
activ	0.060	Proto-Oncogene Prot.	0.060	women	0.032	Middle Aged	0.024
akt	0.060	Proto-Oncogene Proteins c-akt	0.047	tamoxifen	0.028	Receptors, Estrogen	0.023
tyrosin	0.036	1-Phosphatidylinositol 3-Kin.	0.047	mcf-7	0.026	Tamoxifen	0.022
protein	0.029	Humans	0.043	estrogen	0.012	Antineopl. Agents, Hormon.	0.017
phosphatas	0.025	Signal Transduction	0.038	mda-mb-231	0.007	Aged	0.016
signal	0.025	Animals	0.028	adjuv	0.007	Carcinoma, Ductal, Breast	0.013
pten	0.024	Protein Kinases	0.021	statu	0.007	Chemotherapy, Adjuvant	0.013
pi3k	0.022	Tumor Suppressor Proteins	0.016	hormon	0.007	Mammography	0.012
pathwai	0.020	Phosphoric Monoester Hydrol.	0.016	tam	0.006	Breast	0.012
regul	0.018	Enzyme Activation	0.015	aromatas	0.006	Adult	0.011
serin	0.015	Cell Line, Tumor	0.014	ductal	0.006	Neoplasm Staging	0.010
inhibit	0.015	Enzyme Activation	0.001	mammari	0.006	Aromatase Inhibitors	0.009
src	0.015	Mice	0.013	postmenop.	0.005	Receptors, Progesterone	0.009

The collection consists of $D = 50.000$ abstracts, $M = 17.716$ unique MeSH main headings and $N = 22.531$ unique word stems. Word tokens from title and abstract were stemmed with a standard Porter stemmer [27] and stop words were removed using the PubMed stop word list [3]. Additionally, word stems occurring less than five times in the corpus were filtered out. Note that no filter criterion was defined for the MeSH vocabulary.

The second data set contains $D = 84.076$ PubMed abstracts, with $M = 18.350$ unique MeSH main headings and a total of $N = 31.684$ unique word stems. The same filtering steps were applied as described above. This corpus is composed of genetics-related abstracts from the MEDLINE 2005 baseline corpus. The here introduced bias towards genetics-related abstracts resulted from using NLM's Journal Descriptor Indexing Tool by applying some genetics-related filtering strategies [23]. See [23, 24] for more information about both corpora. In the following, the data sets are referred to as *random 50K* data set and *genetics-related* data set respectively. For the extraction of statistical relationships between the various document entities and for uncovering the hidden-topic concept structure, we decided to use the larger genetics-related corpus with all 18.350 MeSH main headings (see section 3.2.1 and section 3.2.2), while for the multi-label classification task, we used both corpora in a pruned setting (see next section 3.1.1).

Parameters for the Topic-Concept model were estimated by averaging samples from ten randomly-seeded runs, each running over 100 iterations, with an initial burn-in phase of 500 iterations (resulting in a total of 1.500 iterations). We found 500 iterations to be a convenient choice by observing a flattening of the log likelihood. The training time ranged from ten to fifteen hours depending on the size of the data set, the number of used MeSH concepts as well as on the predefined number of topics (run on a standard Linux PC with Opteron Dual Core processor, 2.4 GHz).

3.1.1 Multi-label classification task

In this setting, we prune each MeSH descriptor to the first level of each taxonomy-subbranch resulting in 108 unique MeSH concepts ($M = 108$). For example, if a document is indexed with *Muscular Disorders, Atrophic [C10.668.550]*, the concept is pruned to *Nervous System Diseases [C10]*. Therefore, the task is to assign at least one of the 108 classes to an unseen PubMed abstract. Note that from a machine learning point of view, this is a challenging 108 multi-label classification problem and corresponds to other state-of-the-art text classification problems such as the Reuters text classification task [19], where the number of classes is approximately the same. In the pruned setting of our task, we have on average 9.6/10.5 (random 50K/genetics-related) pruned

[3] http://www.ncbi.nlm.nih.gov/entrez/query/static/help/pmhelp.html#Stopwords

Table 3: Selected MeSH concepts from the *Disease* and the *Drug & Chemicals* subbranch with the 20 most probable word stems estimated based on a topic-concept model learned from the genetics-related corpus ($T = 300$). The font size of each word stem encodes its probability given the corresponding MeSH concept. The number in brackets is euqal to the number of times, the MeSH terms occurs in the corpus

Diseases

Myelodysplastic Syndromes (208)	Pulmonary Embolism (39)
acut aml bcr-abl blast chronic cml flt3 hematolog imatinib leukaemia leukem leukemia lymphoblast marrow mds myelodysplast myeloid patient relaps syndrom	activ associ case clinic diagnos diagnosi diagnost factor incid men mortal patient platelet preval protein rate risk studi women year

Drugs & Chemicals

Erythropoietin (85)	Paclitaxel (309)
abnorm anaemia anemia caus cell defect defici disord epo erythrocyt erythroid erythropoietin g6pd hemoglobin increas model normal patient sever studi	advanc agent anticanc cancer chemotherapi cisplatin combin cytotox drug effect median paclitaxel patient phase regimen respons sensit surviv toxic treatment

MeSH labels per document. Parameter estimation remains the same as mentioned in the previous paragraph.

In particular, we are interested in evaluating the classification task in a user-centered or semi-automatic scenario, where we want to recommend a set of classes for a specific document (e. g. a human indexer gets recommendations of MeSH terms for a document). Thus, we decided to follow the evaluation of [13] and average the effectiveness of the classifiers over documents rather than over categories. In addition, we weight recall over precision and use the F2-macro measure, because it reflects that human indexers will accept some inappropriate recommendations as long as the major fraction of recommended index terms will be correct [13].

3.2 Results

3.2.1 Uncovering the hidden topic-concept structure

Table 2 illustrates several different topics (out of 300) from the genetics-related corpus, obtained from a particular Gibbs sampler run after the 1.500th iteration. Each table shows the fifteen most likely word stems assigned to a specific topic and its corresponding most likely MeSH main headings. To show the descriptive power of our learned model, we chose four topics describing different aspects of biomedical research. Topic 6 is ethics-related, topic 16 is related to a special biochemical process, namely signal transduction, and the last two topics represent aspects of specific disease classes. Topic 26 represents a topic centered around breast cancer, while topic 17 refers to HIV. The model includes several other topics related to specific diseases, biochemical processes, organs and other aspects of biomedical research like e. g. Magnetic Resonance Spectroscopy. Recall that the here investigated corpus is biased towards genetics-related topics, thus, some topics can describe quite specific aspects of genetics research. More generic topics in the corpus are related to terms, common to almost all biomedical research areas including terminology, describing experimental setups or methods. In general, the extracted topics are, of course, dependent on the corpus seed. The full list of topics with corresponding word and MeSH distributions is available at `www.dbs.ifi.lmu.de/~bundschu/TCmodel_supplementary/TC_structure.txt`.

It can be seen that the word stems already provide an intuitive description of specific aspects. Furthermore, the topics gain more descriptive power by their associated MeSH concepts, providing an accurate description in structured from. Note that the standard topic models are only able to represent topics with the single word descriptions. In contrast, the TC model provides a richer representation of topics by additionally linking topics to concepts from a terminological ontology. We found that the topics obtained from different Gibbs sampling runs were relatively stable. A variability in terms of ranking of the words and MeSH terms in the topics can be observed, but overall the topics match very closely. For studies about topic stability in aspect models, please refer to [29].

3.2.2 Extraction of statistical relationships

Besides uncovering the hidden topic-concept structure, we apply the model to derive statistical relations between MeSH concepts and word stems, thus bridging the gap between natural free text and the structured semantic annotation. The derived relations could be e. g. used for improving word sense disambiguation [18]. In Table 3, four MeSH concepts from the *Disease* and the *Drug & Chemicals* subbranch and their twenty most probable word stems are shown. For each MeSH concept, the distribution over words is graphically represented by varying the font size for each word stem with respect to the probability. Given a concept c, the conditional probability for each word is estimated by $p(w|c) \propto \sum_t p(w|t)p(t|c)$, which is computed from the learned model parameters. The word distributions describe the corresponding MeSH concept in an intuitive way, capturing the topical diversity of certain MeSH concepts. Note

Table 4: Selected MeSH concepts from the *Disease* and the *Drug & Chemicals* subbranch with the three most probable topics estimated based on a topic-concept model learned from the genetics-related corpus ($T = 300$). Topics are illustrated here by the twenty most probable word stems.

MeSH term	Topic	Word stems
Myelodysplastic Syndromes (208)	Topic 46 ($p = 0.20$)	leukemia acut myeloid aml mds lymphoblast leukaemia blast leukem patient myelodysplast marrow syndrom malign flt3 bone promyelocyt hematolog mll granulocyt
	Topic 75 ($p = 0.02$)	transplant donor recipi graft stem allogen reject autolog cell immunosuppress allograft marrow surviv hematopoiet condit receiv acut gvhd engraft diseas
	Topic 25 ($p = 0.01$)	chromosom aberr transloc cytogenet delet abnorm rearrang genom karyotyp gain loss region arm breakpoint trisomi mosaic duplic cgh case imbal
Erythropoietin (85)	Topic 177 ($p = 0.30$)	defici adren anemia malaria parasit plasmodium mosquito falciparum erythrocyt cortisol erythropoietin caus g6pd insuffici adrenocort acth anaemia epo anophel develop
	Topic 14 ($p = 0.14$)	cell stem progenitor hematopoiet differenti embryon lineag hsc adult marrow bone erythroid cd34+ precursor potenti cd34 marker hematopoiesi msc self-renew
	Topic 140 ($p = 0.07$)	activ nf-kappab factor nuclear transcript express cell induc inhibit constitut ap-1 regul c-jun suppress p65 kappa curcumin transloc nfkappab c-fo

that there are many other opportunities to access statistical relations between MeSH concepts and words. One could e. g. use measurements like relative frequency or χ^2 statistics. It may be that the TC model captures relationships that can't be captured in a simpler way, but this evaluation is out of scope of the here presented work. We provide all word clouds for all MeSH terms occurring in the corpus from the *Disease* and the *Drug & Chemicals* subbranch as supplementary data (`www.dbs.ifi.lmu.de/~bundschu/TCmodel_supplementary/`).

Another important use case we consider, is the task of estimating the most likely topics given a specific MeSH term with respect to a seed corpus. This results in a fast overview over the topics in which a specific MeSH term is most likely to be involved. Table 4 shows two such examples extracted from the genetics-related corpus. Because of lack of space, we only represent the topics by the most likely word stems (the associated MeSH terms for the topics can be investigated in the supplementary file, mentioned in section 3.2.1). The first example shows the three most likely topics for the MeSH term *myelodysplastic syndromes*. Myelodysplastic syndromes, also called pre-leukemia or 'smoldering' leukemia, are diseases in which the bone marrow does not function normally and not enough blood cells are produced [26]. This fact is reflected by the most likely topic for this MeSH term (Table 4, Topic 46). Furthermore, a state-of-the-art treatment of this disease, is bone marrow transplantation. First, all of the bone marrow in the body is going to be destroyed by high-doses of chemotherapy and/or radiation therapy. Then healthy marrow is taken from a donor (i. e. another person) and is given to the patient [26]. This is described by the second most likely topic (Table 4, Topic 75). Topic 25 constitutes that Myleodysplastic syndromes have a genetic origin and that gene and chromosome aberrations are a likely cause of this disease [26].

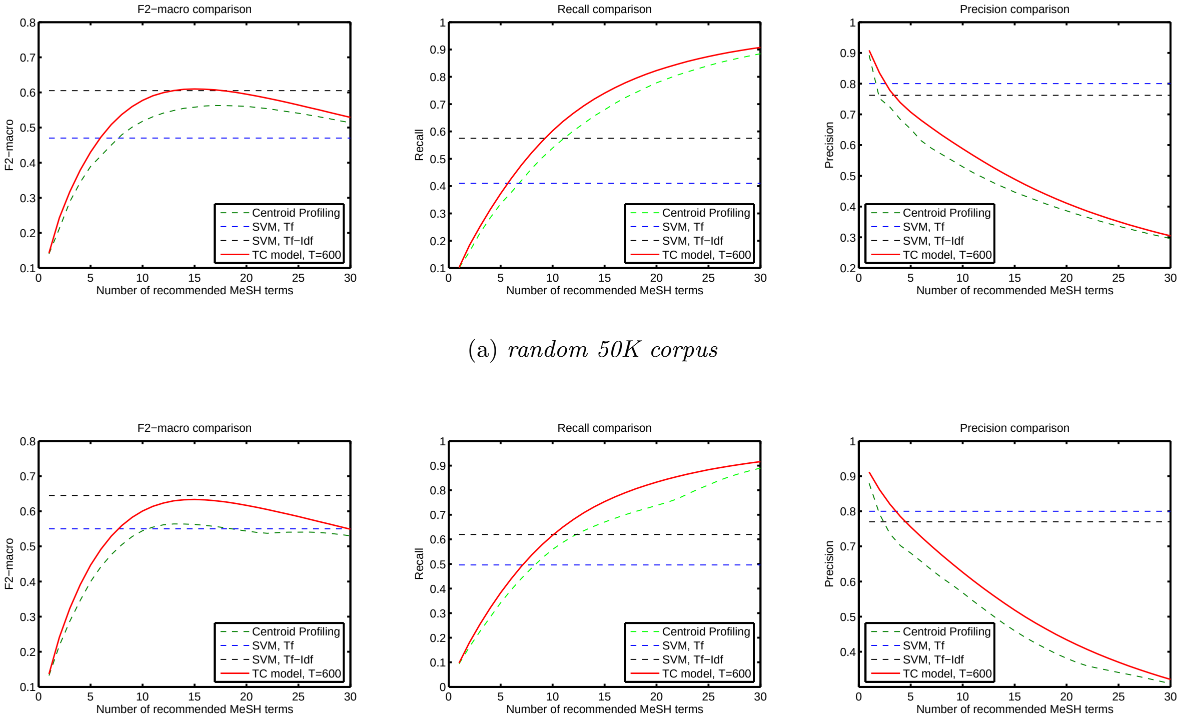
The second MeSH term in table 4, *Erythropoietin* (EPO), is a hormone which is produced by the kidney and liver. It is known to regulate red blood cell production. In the mined genetics-related corpus, the most likely topic (Table 4, Topic 177) states that erythropoietin could be used as a treatment during malaria infection [9] and this is a current issue of ongoing research [3, 31]. Erythropoietin is known to directly promote the generation of neuronal stem cells from progenitors, which is reflected by Topic 14. Last but not least, Topic 140 provides information about the gene regulatory context of EPO. NF-kappaB, e. g. , regulates EPO [8], while EPO in turn regulates expression of c-jun and AP-1 [28].

A full list of all MeSH terms and its most likely associated topics is available online. (`www.dbs.ifi.lmu.de/~bundschu/TCmodel_supplementary/mesh_associated_topics.pdf`).

3.2.3 Multi-label classification

In what follows, we will first describe the used benchmark methods and then present the results for the multi-label classification problem with 108 classes for the genetics-related corpus and the random 50K corpus. The prediction results of the Topic-Concept model are benchmarked against a method currently used by the NLM [17], which we refer to as *centroid profiling*, a multi-label naive Bayes classifier and a multi-label SVM. For both data sets and all methods, 5-fold cross-validation was conducted.

In [17] classification is tackled by computing for each word token w_i and each class label y_m, in a training corpus, a term frequency measure $TF_{i,m} = w_{i,y_m} / \sum_{m=1}^{M} w_{i,y_m}$ with M equals to the total number of classes. Thus, $TF_{i,m}$ measures the number of times a specific word w_i co-occurs with the class label y_m, normalized by the total number of times the word w_i occurs. As a consequence, each word token in the training can be represented by a profile consisting of the term frequency distribution over all M classes. When index-



(a) *random 50K corpus*

(b) *genetics-related corpus*

Figure 2: F2-macro, recall and precision plots for discipline-based indexing. Results are plotted according to the number of top n recommended MeSH terms. In average every document has 9.6 such assignments in our experimental setting. (a) Plots for the randomly selected data set. (b) Plots for the genetics-related data set

ing a new unseen document, the centroid over all profiles for the word tokens in the test document is computed. This centroid represents the ranking of all class labels for the test document. This method was chosen, because it is currently used by the NLM in a classification task to predict so-called journal descriptors [17].

According to [22], naive Bayes classifiers are a very successful class of algorithms for learning to classify text documents. For the multi-label naive Bayes classifier, we assumed a bag of words representation like for the Topic-Concept model and trained it for each of the 108 labels. We used the popular multinomial model for naive Bayes [21].

The multi-label SVM setting was implemented according to [19]. In this setting, a linear kernel is used and the popular so-called binary method is used to adapt the SVM to a multi-label setting. It has been shown that this setting produced very competitive results on a large-scale text classification task on the RCV1 Reuters corpus [19]. LIBLINEAR, a part of the LIBSVM package [10] is used for the implementation. Two different weighting schemes are evaluated: Term frequency (Tf) as well as cosine-normalized Term frequency-inverse document frequency (Tf-Idf).

In the TC-model, the prediction of concept terms for unseen documents can be formulated as follows: Based on the word-topic and concept-topic count matrices learned from an independent data set, the likelihood of a concept c given the test document d is $p(c|d) = \sum_t p(c|t)p(t|d)$. The first probability in the sum, $p(c|t)$, is given by the learned topic-concept distribution (see Equation 7). The mixture of topics for the document $p(t|d)$ is estimated by drawing for each word token in the test document a topic based on the learned word-topic distribution $p(w|t)$ (see Equation 5). Therefore, the TC model directly predicts a ranked list of class recommendations, in contrast to the classical task of topic models in text classification problems, where they are usually used for dimensionality reduction and afterwards standard classifiers are applied [7].

We now discuss experimental results using 5-fold cross-validation. Figure 2 plots F2-macro measure, recall and precision against the number of recommended MeSH terms. Figure 2(a) shows results for the random 50K data set and Figure 2(b) for the genetics-related data set respectively. Our TC model and the centroid profiling method provide as output a ranked list of recommendations. In order to be able to compare these two methods with the other classifiers, a thresholding strategy is needed [32]. We decided to use the simple rank-based thresholding (*Rcut*) [32] and evaluate the results until a cut-off value of 30 (Recall that each document has in average 9.6 (random 50K) and 10.5 (genetics-related) MeSH entries in our experimental setting. The Topic-Concept model was trained with two different number of topics on both data sets ($T = 300$, $T = 600$ for the 50K random corpus and $T = 300$, $T = 600$ for the genetics-related corpus). For clarity, we only show the results for $T = 600$ here, since experimental validation showed that the number of topics is not very sensitive to the overall performance. We also exclude the NB classifier from the figure for clarity (F-measure 0.58 and 0.60 for random 50K and genetics-related). In terms of F2-macro, recall and precision, the Topic-Concept model clearly outperforms the centroid profiling. The naive Bayes classifier already yields quite competitive results. Regarding F2-macro, the TC models reach their optimum at 15 returned recommendations for both data sets (0.61 (random 50K)/0.635 (genetics-related)). At a cut-off value of 15 recommendations, centroid profiling reaches a F2-macro of 0.558 for the random 50K data set (optimum at 17 recommendations with 0.562) and 0.562 for the genetics-related corpus (optimum at 13 recommendations with 0.564). Using a cut-off value which equals to the number of average MeSH assignments (rounded-up) in the two training corpora the F2-macro is for the best TC models 0.59 (random 50K) and 0.61 (genetics-related), while the centroid profiling reaches only 0.517 (random 50K) and 0.55 (genetics-related) at this cut-off value. Note that using the average number of MeSH assignments is the most simple way to determine an appropriate cut-off value. A more analytical way of determining the threshold would be to set up an independent development set for the given corpus and to maximize the F2-macro measure according to the number of recommendations. Other approaches e. g. use a default length of 25 recommended index terms [1] for unpruned MeSH recommendation. The evaluation of the multi-label SVM shows that the performance is very sensitive to the used term weighting scheme (see Figure 2). When using Tf-Idf, the SVM is approximately on par with the TC model in terms of F2-macro on both data sets (F2-macro SVM, Tf-Idf is 0.60 (random 50K) and 0.645 (genetics-related)). The SVM is clearly superior in terms of precision due to its discriminative nature. When considering recall, the TC model outperforms the SVM with Tf-Idf, effective from a cut-off value of recommended MeSH terms, which is the average number of MeSH terms in the training corpora.

4. CONCLUSION AND OUTLOOK

This study presents a new probabilistic topic model for modeling medical text indexing processes. The so-called Topic-Concept model automatically learns the relation between words, MeSH terms, documents and topics from large text corpora of PubMed abstracts. The method uses a generative probabilistic process to learn the just mentioned relationships by extracting the latent topic structure. Gibbs sampling is used to learn the Topic-Concept model.

The TC model uncovers novel information from a biomedical text corpus, including the extraction of the hidden topic-concept structure, using all occurring unique MeSH terms in the corpus (18.350 distinct MeSH terms). In contrast to standard topic models, where topics are solely represented by their most likely words, the here extracted topic-concept structure can be interpreted as a richer representation of topics by additionally linking to concepts from the MeSH thesaurus. Thus, the enriched topic representation provides important additional information from a terminological ontology. Other use cases we explore, are the extraction of statistical relationships between words and MeSH terms as well as between topics and MeSH terms. The just mentioned applications can have impact on several other closely related areas such as information retrieval or information extraction (see e. g. [25]).

The Topic-Concept model can be easily applied to text classification tasks. Even though the here proposed method is generative, the experimental evaluation on a challenging multi-label classification problem on two independent data sets with 108 class labels against discriminative methods proves our method to be competitive in terms of F2-macro and even superior in terms of recall. In contrast to most text categorization algorithms, the here proposed model provides a ranking of recommended index terms for prediction tasks. Up to now, the choice of the number of returned recommended index terms is user-defined. Using a simple cut-off value which is equal to the number of average index terms assigned in a training collection, already yields competitive results.

In the current setting, our model neglects the hierarchical property of the MeSH thesaurus. The extension of the underlying generative process for capturing the hierarchy of terminological ontologies is a matter of ongoing research. To further tune prediction performance, we are also considering an expansion of the generative Topic-Concept model to a supervised topic model for multi-label classification as lately proposed by [4] for multi-class classification problems.

Funding

The work presented here was partially funded by the German Federal Ministry of Economy and Technology (BMWi) under the THESEUS project.

Acknowledgements

The authors wish to thank Aurélie Névéol from the NLM for answering questions about her research as well as for providing the data sets.

5. REFERENCES

[1] A. R. Aronson, J. G. Mork, C. W. Gay, S. M. Humphrey, and W. J. Rogers. The nlm indexing initiative's medical text indexer. In *Medinfo 2004*, pages 268–272. IOS Press, 2004.

[2] C. Biemann. Ontology learning from text: A survey of methods. *LDV-Forum*, 20(2):75–93, 2005.

[3] A.-L. L. Bienvenu, J. Ferrandiz, K. Kaiser, C. Latour, and S. Picot. Artesunate-erythropoietin combination for murine cerebral malaria treatment. *Acta tropica*, February 2008.

[4] D. Blei and J. Mcauliffe. Supervised topic models. In J. C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*. MIT Press, Cambridge, MA, 2008.

[5] D. M. Blei, K. Franks, M. I. Jordan, and I. S. Mian. Statistical modeling of biomedical corpora: mining the caenorhabditis genetic center bibliography for genes related to life span. *BMC Bioinformatics*, 7(1), May 2006.

[6] D. M. Blei, M. I. Jordan, J. Callan, G. Cormack, C. Clarke, D. Hawking, and A. Smeaton. Modeling annotated data. *SIGIR Forum*, (SPEC. ISS.):127–134, 2003.

[7] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, January 2003.

[8] G. Carvalho, C. Lefaucheur, C. Cherbonnier, D. Metivier, A. Chapel, M. Pallardy, M.-F. Bourgeade, B. Charpentier, F. Hirsch, and G. Kroemer. Chemosensitization by erythropoietin through inhibition of the nf-[kappa]b rescue pathway. *Oncogene*, aop(current), December 2004.

[9] C. Casals-Pascual, R. Idro, N. Gicheru, S. Gwer, B. Kitsao, E. Gitau, R. Mwakesi, D. J. Roberts, and C. R. Newton. High levels of erythropoietin are associated with protection against neurological sequelae in african children with cerebral malaria. *Proceedings of the National Academy of Sciences*, 105(7):2634–2639, February 2008.

[10] C.-C. Chang and C.-J. Lin. Libsvm: a library for support vector machines, 2001.

[11] K. B. Cohen and L. Hunter. *Natural language processing and systems biology*, pages 147–174. Springer, December 2004.

[12] 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.

[13] 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.

[14] T. L. Griffiths and M. Steyvers. Finding scientific topics. *Proc Natl Acad Sci U S A*, 101 Suppl 1:5228–5235, April 2004.

[15] T. Hofmann. Probabilistic latent semantic analysis. In *Proc. of Uncertainty in Artificial Intelligence, UAI'99*, Stockholm, 1999.

[16] T. Hofmann. Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning*, V42(1):177–196, January 2001.

[17] S. Humphrey, C. Lu, W. Rogers, and A. Browne. Journal descriptor indexing tool for categorizing text according to discipline or semantic type. In *AMIA Annu Symp Proc*, 2006.

[18] S. M. Humphrey, W. J. Rogers, H. Kilicoglu, D. Demner-Fushman, and T. C. Rindflesch. Word sense disambiguation by selecting the best semantic type based on journal descriptor indexing: preliminary experiment. volume 57, pages 96–113, 2006.

[19] D. Lewis, Y. Yang, T. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.

[20] A. Mccallum, A. Corrada-Emmanuel, and X. Wang. Topic and role discovery in social networks. 2005.

[21] A. Mccallum and K. Nigam. A comparison of event models for naive bayes text classification, 1998.

[22] T. M. Mitchell. *Machine Learning*. McGraw-Hill Science/Engineering/Math, March 1997.

[23] A. Névéol, S. E. Shooshan, S. M. Humphrey, T. C. Rindflesch, and A. R. Aronson. Multiple approaches to fine-grained indexing of the biomedical literature. In R. B. Altman, K. A. Dunker, L. Hunter, T. Murray, and T. E. Klein, editors, *Pacific Symposium on Biocomputing*, pages 292–303. World Scientific, 2007.

[24] A. Névéol, S. E. Shooshan, J. G. Mork, and A. R. Aronson. Fine-grained indexing of the biomedical literature: Mesh subheading attachment for a medline indexing tool. In *Proc. AMIA Symp*, 2007.

[25] D. Newman, C. Chemudugunta, and P. Smyth. Statistical entity-topic models. In *KDD '06: Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 680–686, New York, NY, USA, 2006. ACM Press.

[26] S. D. Nimer. Myelodysplastic syndromes. *Blood*, 111(10):4841–4851, May 2008.

[27] M. F. Porter. An algorithm for suffix stripping. pages 313–316, 1997.

[28] S. R. Seong, J. W. Lee, Y. K. Lee, T. I. Kim, D. J. Son, D. C. Moon, Y. W. Yun, d. o. . Y. Yoon, and J. T. Hong. Stimulation of cell growth by erythropoietin in raw264.7 cells: association with ap-1 activation. *Archives of pharmacal research*, 29(3):218–223, March 2006.

[29] M. Steyvers and T. Griffiths. *Probabilistic Topic Models*. 2007.

[30] M. Steyvers, P. Smyth, M. Rosen-Zvi, and T. Griffiths. Probabilistic author-topic models for information discovery. In *KDD '04: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 306–315, New York, NY, USA, 2004. ACM Press.

[31] L. Wiese, C. Hempel, M. Penkowa, N. Kirkby, and J. A. L. Kurtzhals. Recombinant human erythropoietin increases survival and reduces neuronal apoptosis in a murine model of cerebral malaria. *Malaria Journal*, 7:3+, January 2008.

[32] Y. Yang. An evaluation of statistical approaches to text categorization. *Information Retrieval*, 1(1/2):69–90, 1999.

[33] B. Zheng, D. C. Mclean, and X. Lu. Identifying biological concepts from a protein-related corpus with a probabilistic topic model. *BMC Bioinformatics*, 7, 2006.

Information Theoretic Methods for Detecting Multiple Loci Associated with Complex Diseases

Pritam Chanda*
Dept. of Computer Science and Engineering, State University of New York, Buffalo, USA
pchanda@cse.buffalo.edu

Aidong Zhang
Dept. of Computer Science and Engineering, State University of New York, Buffalo, USA
azhang@cse.buffalo.edu

Lara Sucheston
Dept. of Biostatistics, State University of New York, Buffalo, USA
lsuchest@buffalo.edu

Murali Ramanathan
Dept. of Pharmaceutical Sciences, State University of New York, Buffalo, USA
murali@buffalo.edu

ABSTRACT

Gene-gene interactions play important roles in the etiology of complex multi-factorial diseases. With the advancements in genotyping technology, large genetic association studies based on hundreds of thousands of single-nucleotide polymorphisms are a popular option for the study of complex diseases. Association studies using locus by locus analyses till remains the primary method although the study of gene-gene interactions has become more common using regression based methods. However, regression based methods are computationally heavy and model complexity increases rapidly with the increase in number of loci and also with the number of possible allelic states at each locus. Information theoretic approaches offer many potent capabilities and advantages for the analyses of gene-gene interactions. In this paper, we develop and explore the effectiveness of two information theoretic metrics in identifying gene-gene interactions using extensive simulations on four different gene-gene interaction models. We propose a forward selection algorithm using the metrics and evaluate its performance using the rheumatoid arthritis dataset from Genetic Analysis Workshop-15. We demonstrate that our metrics are capable of analyzing a diverse range of epidemiological data sets containing evidences for gene-gene interactions.

Keywords

Gene-gene interaction, information theory, entropy, complex diseases.

1. INTRODUCTION

Complex interactions involving a number of genes and multiple single nucleotide polymorphisms (SNP) and environmental factors are known to be associated with the risk of developing diseases such as cancer, autoimmune disease and cardiovascular disease. Advances in high throughput genotyping methods and the completion of human genome project have made generating large scale dense genetic maps of the human genome for epidemiological studies feasible [1, 2]. The successful identification of critical gene-gene and gene-environment interactions can provide the scientific basis for preventative and curative measures to help individuals with particular genetic susceptibilities. The additional information from these methods improves the prospects for uncovering potentially undiscovered genes involved in complex interactions underlying the genetic etiology of multi-factorial diseases.

Single-locus based association analysis methods fail to detect all the loci affecting the disease susceptibility when observable marginal effects at each locus are small [3, 4]. Analysis of two-locus models modeling the interaction involving a pair of locus and the disease phenotype have been studied by several researchers and shown to be computationally feasible when involving hundreds of thousands of loci [1, 2, 5, 6]. Traditionally, regression based methods such as multiple and logistic regression has been used for analysis of genetic models in which two susceptibility loci jointly influence the risk of developing a disease [7-10]. These methods involve the comparison of likelihoods of models incorporating different sets of disease model parameters that allows inferences to be drawn regarding the nature of the joint effect of the loci. However, regression based methods are computationally intensive and model complexity increases rapidly with the increase in number of loci and also with the number of possible allelic states at each locus.

Information theoretic methods are among the most promising approaches for genetic disease association studies [11, 12]. Information-theoretic methods are not only based on strong theoretical backgrounds but are also versatile and are independent of the underlying genetic models. But only limited research has been done on leveraging these strengths for analysis of multi-locus disease association studies. Several reports have used the Kullback-Leibler divergence (KLD) for genetic analysis. The KLD being a measure of the 'distance' between two distributions, it has been applied for 2-group comparisons such as those used to evaluate ancestry informative markers [11, 13, 14], as multi-locus linkage disequilibrium (LD) measure to identify of tag SNPs [11] and for analytical visualization [15-17]. Information theory based statistics have been proposed for genome-wide data analysis to test for allelic associations [18] and in identifying and visualizing gene-gene and gene-environment interactions [17].

In this paper, we critically evaluate the effectiveness of two information theoretic metrics in detecting statistical gene-gene and gene-environment interactions in complex disease models. We choose some well-studied two-locus models of statistical

*Corresponding author

BIOKDD'08, August 24, 2008, Las Vegas, Nevada, USA.
Copyright 2008 ACM 978-1-60558-302-0/08/0008…$5.00.

interactions and present their simulation studies to evaluate the power of information theoretic methods in identifying gene-gene incorporated in these models for numerous settings of the parameters (e.g. allele frequencies, linkage disequilibrium) involved. Using the metrics, we propose a simple forward selection algorithm to effectively scan data sets containing large number of loci and additional environmental variables (covariates). Finally, we assess the performance of the algorithm using the simulated rheumatoid arthritis dataset from Genetic Analysis Workshop 15 (GAW15) [19].

2. METHODS

In this section, we define the information theoretic metrics that we shall use for detecting gene-gene interactions. Then we describe in details several two-locus disease models we have used to evaluate the effectiveness of the information theoretic measures in detecting the interactions incorporated in the disease models.

2.1 Terminology and Representation

Let $S = \{X_1, X_2,\ldots, X_n\}$ be the set of genetic variables to be analyzed where X_i denotes the random variable representing the genotypes at locus L_i. We assume L_i is biallelic (with alleles *A* and *a*) with three possible genotypes (*AA*, *Aa*, *aa*). The uncertainty of X_i is given by Shannon's entropy [20] as,

$$H(X_i) = -\sum_{x\in\{AA,Aa,aa\}} P(X_i = x)\log_2 P(X_i = x)$$

Let C be the random variable representing the disease status (phenotype variable). The mutual information between each X_i and C is denoted by $I(X_i;C)$ measures the mutual dependence of the two variables. It quantifies the distance between the true joint distribution of X_i and C and the joint distribution when X_i and C are independent. The *m-way total correlation* involving variables $\left\{X_1, X_2,\ldots, X_m\right\} \subseteq S$ is defined as [21] ,

$$TC(X_1;X_2;\ldots;X_m) = \sum_{i=1}^{m} H(X_i) - H(X_1X_2\ldots X_m)$$

The *TC* is the amount of information shared among the variables in the set. A *TC* value that is zero indicates independence, i.e. knowing the value of one variable tells you nothing about the others. The maximal value of *TC* occurs when one variable is completely redundant with the others; i.e., knowing one variable provides complete knowledge regarding all the others.

Phenotype associated information: Given a set of genetic variables $\left\{X_1, X_2,\ldots, X_m\right\} \subseteq S$, since we are interested in learning the information shared among the genetic variables with the phenotype variable, the informative part of *TC* over all the variables (including the phenotype variable) is obtained by subtracting from it the *TC* representing the interdependencies among the genetic variables in the absence of the phenotype variable *C*. Therefore,

$$TC(X_1;X_2;\ldots,X_m;C) - TC(X_1;X_2;\ldots;X_m)$$
$$= I(X_1X_2\ldots X_m;C)$$

Thus the information shared among the genetic variables with the phenotype variable *C* is given by the mutual information between *C* and the joint distribution of the *m* genetic variables. We call this measure of information *phenotype associated information* (*PAI*). In terms of the probabilities of the variables, it can be represented as,

$$PAI(X_1;X_2;\ldots;X_m;C) = I(X_1X_2\ldots X_m;C)$$
$$= \sum_{V} p(x_1,x_2,\ldots,x_m,c)\log_2\left(\frac{p(x_1,x_2,\ldots,x_m,c)}{p(x_1,x_2,\ldots,x_m)p(c)}\right)$$

where V represents the set of possible values the set $\{X_1, X_2,\ldots, X_m, C\}$ takes.

Pooled phenotype associated information: Let $\{X_1, X_2,\ldots,X_m\}$ be a set of genetic variables taking on genotypes values from the set G. Let g be any genotype in G. Denote the set of genotypes $G\backslash\{g\}$ by $\widetilde{G}$. We pool the genotypes in set $\widetilde{G}$ into a single genotype $\widetilde{g}$ such that $p(\widetilde{g}) = \sum_{x\in\widetilde{G}} p(x)$. Let X be a binary random variable on the set $\{g, \widetilde{g}\}$.

Then the pooled PAI is given by

$$PAI_{pooled}(X_1;X_2;\ldots,X_m;C)$$
$$= \frac{1}{|G|}\left(\sum_{g\in G}\left(\sum_{x\in\{g,\widetilde{g}\},c\in C} p(x,c)\log_2\left(\frac{p(x,c)}{p(x)p(c)}\right)\right)\right)$$

Here, for each genotype $g \in G$, we pool the remaining genotypes into one genotype group for the purpose of calculating the contribution of g to the PAI_{pooled}. Thus this metric consists of the average contribution from each genotype in explaining the disease phenotype when the phenotype information in the remaining genotypes are pooled into one single block of genotype

2.2 Forward Selection Algorithm

The above two metrics can be used to design a simple stepwise forward selection algorithm. The algorithm takes as input the set of genetic variables, the phenotype variable, the metric to use *M*, and algorithm parameters ω and τ, which represent the number of combinations retained in each iteration of the search and the number of iterations to execute, respectively. At each iteration, the algorithm greedily tries to search for ω combinations of genetic variables of increasing sizes that has the highest values of *M*. It starts by calculating $M(X_i;C)\forall i \in 1\ldots n$. Top ω combinations with the highest values of *M* are retained. Let this set of variables be denoted by S_1. In the next step, $M(X_i;X_j;C)$ $\forall X_i \in S_1$, $\forall j \in 1\ldots n$, $(j \neq i)$ is calculated.

Again the top ω combinations with the highest values of M in are retained in the set S_2. The above steps are repeated τ times. To assess the computation complexity involved, let m be the sample size of the data and n be the number of variables (excluding the phenotype variable). Lines 2-4 take $O(nm^2)$ computations because computation of the metric consumes $O(m^2)$ computations. Lines 7-19 take $O(\tau n \omega m^2 + \tau n\omega^2)$ computations since computations of M are repeated for τ (*for* loop in Line 7) times n (*for* loop in Line 9) times ω (*for* loop in Line 10) computations, and line 17 take $O(n\omega^2)$ computations. We recommend setting τ to a value in the range 2-5 since we very rarely find gene-gene statistical interactions involving more than 5 variables at a time. The value of ω can be chosen according to the size of the data and the amount of time the researcher is willing to spend for the search. For example, with 100,000 markers we recommend setting it to 50 to successfully capture the interactions within a reasonable timeframe.

```
Algorithm : Forward Selection Algorithm ($S, C, \omega, \tau, M$)
Input $S$(Set of genetic variables), $C$(Phenotype variable), $\omega$(# of variable combinations to retain at each iteration), $\tau$(# of iterations), $M$(The chosen metric, $PAI$ or $PAI_{pooled}$)
Output $Q$(Significant combinations with metric values)
1.  $Z \leftarrow \phi$   $Q \leftarrow \phi$
2.  for each variable $V \in S$ do
3.      $Z \leftarrow Z \cup \{V, M(V;C)\}$
4.  endfor
5.  $W_1 \leftarrow$ Top $\omega$ combinations from $Z$ ranked by $M$
6.  $Q \leftarrow Q \cup W_1$
7.  for $i \leftarrow 2$ to $\tau$ do
8.     $Z \leftarrow \phi$
9.     for each variable $V \in S$ do
10.       for each combination $v \in W_{i-1}$ do
11.           if $V$ is not included in the combination $v$
12.             $v \leftarrow v \cup \{V\}$
13.             $Z \leftarrow Z \cup \{v, M(v;C)\}$
14.           endif
15.       endfor
16.    endfor
17.    $W_i \leftarrow$ Top $\omega$ combinations from $Z$ ranked by $M$
18.    $Q \leftarrow Q \cup W_i$
19. endfor
20. Retain only those combinations in $Q$ that are found to be
    significant using permutation or bootstrapping methods
21. return $Q$
```

2.3 Disease Models

We focus on two-locus gene-gene interaction models that attempts to mimic biological interactions. In an effort to classify the types of interaction in the case of two biallelic loci, Li and Reich [6] have enumerated 512 possible two-locus models and identified a fewer number of non-redundant two-locus models. Four widely used models are selected in this paper for demonstrating the effectiveness of the information theoretic metrics in detecting gene-gene interactions. Each model specifies the penetrance of the disease given the genotypes of the two interacting loci. Let the two loci be denoted by L_1 and L_2. We assume each loci is biallelic with three possible genotypes. Let the two alleles at loci L_1 be A and a and the genotypes are aa, Aa and AA. Let the two alleles at loci L_2 be B and b and the genotypes are bb, Bb and BB. Let λ_{aa} , λ_{Aa} , λ_{AA} be the marginal penetrances at L_1 and λ_{bb} , λ_{Bb} , λ_{BB} be the marginal penetrances at L_2. Denote the joint penetrances for each genotype of the two loci by μ_{aabb}, μ_{aaBb}, μ_{aaBB}, μ_{Aabb}, μ_{AaBb}, $\mu_{AaBB,}$ μ_{AAbb}, μ_{AABb}, and μ_{AABB}. Then,

$$P(Disease \mid Genotype = g) = \mu_g$$

where $g \in \{aabb, aaBb, aaBB, Aabb, AaBb, AaBB, AAbb, AABb, AABB\}$

The marginal penetrances at each locus is given by,

Locus L_1

$$\lambda_{aa} = \mu_{aabb}P(bb) + \mu_{aaBb}P(Bb) + \mu_{aaBB}P(BB)$$

$$\lambda_{Aa} = \mu_{Aabb}P(bb) + \mu_{AaBb}P(Bb) + \mu_{AaBB}P(BB)$$

$$\lambda_{AA} = \mu_{AAbb}P(bb) + \mu_{AABb}P(Bb) + \mu_{AABB}P(BB)$$

Locus L_2

$$\lambda_{bb} = \mu_{aabb}P(aa) + \mu_{Aabb}P(Aa) + \mu_{AAbb}P(AA)$$

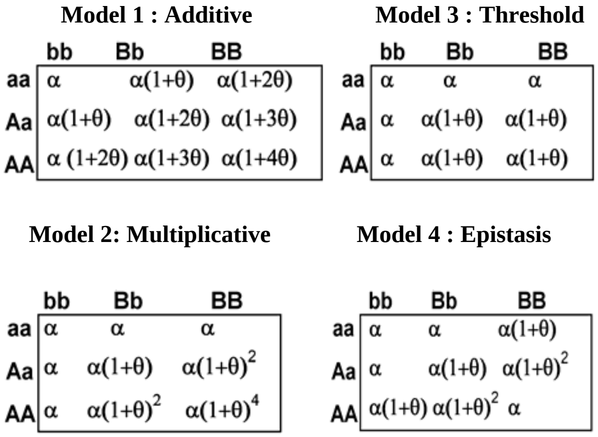
$$\lambda_{Bb} = \mu_{aaBb}P(aa) + \mu_{AaBb}P(Aa) + \mu_{AABb}P(AA)$$

$$\lambda_{BB} = \mu_{aaBB}P(aa) + \mu_{AaBB}P(Aa) + \mu_{AABB}P(AA)$$

And, the overall population prevalence of the disease is given by,

$$\begin{aligned}P(Disease) &= \mu_{aabb}P(aabb) + \mu_{aaBb}P(aaBb) + \mu_{aaBB}P(aaBB)\\ &+ \mu_{Aabb}P(Aabb) + \mu_{AaBb}P(AaBb) + \mu_{AaBB}P(AaBB)\\ &+ \mu_{AAbb}P(AAbb) + \mu_{AABb}P(AABb) + \mu_{AABB}P(AABB)\end{aligned}$$

The genotype frequencies can be calculated using the allele frequencies at each loci under Hardy Weinberg equilibrium assumptions. The four models are summarized below. Model 1 is an additive model that has a baseline penetrance for genotype *aabb* and it increases in an additive fashion with each copy of the disease causing allele in the genotype. Model 2 incorporates a multiplicative interaction with a baseline value that increases the chance of disease multiplicatively when at least one disease causing allele from each locus is present [2]. Model 3 is similar to Model 2 but specifies a threshold of disease probabilities and requires at least one copy of the disease causing allele from each locus of the corresponding genotype to have higher penetrance [2, 6]. However, increase in the number of disease causing alleles does not increase the chances of disease. Model 4 incorporates epistatic effects such that chances of disease increases from baseline value only for the genotypes containing two or three disease associated alleles from both the loci.



Model 1 : Additive

	bb	Bb	BB
aa	α	α(1+θ)	α(1+2θ)
Aa	α(1+θ)	α(1+2θ)	α(1+3θ)
AA	α (1+2θ)	α(1+3θ)	α(1+4θ)

Model 3 : Threshold

	bb	Bb	BB
aa	α	α	α
Aa	α	α(1+θ)	α(1+θ)
AA	α	α(1+θ)	α(1+θ)

Model 2: Multiplicative

	bb	Bb	BB
aa	α	α	α
Aa	α	α(1+θ)	$α(1+θ)^2$
AA	α	$α(1+θ)^2$	$α(1+θ)^4$

Model 4 : Epistasis

	bb	Bb	BB
aa	α	α	α(1+θ)
Aa	α	α(1+θ)	$α(1+θ)^2$
AA	α(1+θ)	$α(1+θ)^2$	α

Table 1. Penetrances for the four models across the genotypes of the two loci.

Given fixed values of the disease prevalence and the allele frequencies at each locus, for each model, the marginal effects at each locus are bounded by some maximum value ξ and the interaction effects (θ and α) are solved for by working backwards using the above equations. For example, consider Model 2. Let allele frequencies at locus L_1 and L_2 be P_A and P_a, and P_B and P_b, respectively. Then the marginal penetrances and the disease prevalence are given by the following equations:-

$$\lambda_{aa} = \alpha$$

$$\lambda_{Aa} = \alpha(1+\theta)^2 P_B^2 + 2P_B P_b \alpha(1+\theta) + \alpha P_b^2$$

$$\lambda_{AA} = \alpha(1+\theta)^4 P_B^2 + 2P_B P_b \alpha(1+\theta)^2 + \alpha P_b^2$$

$$\lambda_{bb} = \alpha$$

$$\lambda_{Bb} = \alpha(1+\theta)^2 P_A^2 + 2P_A P_a \alpha(1+\theta) + \alpha P_a^2$$

$$\lambda_{BB} = \alpha(1+\theta)^4 P_A^2 + 2P_A P_a \alpha(1+\theta)^2 + \alpha P_a^2$$

$$P(Disease) = \alpha(P_a^2 P_b^2 + 2P_a^2 P_B P_b + P_a^2 P_B^2 + 2P_A P_a P_b^2 + P_A^2 P_b^2) + 4\alpha(1+\theta)P_A P_a P_B P_b + \alpha(1+\theta)^2(2P_A^2 P_B P_b + 2P_A P_a P_B^2) + \alpha(1+\theta)^4 P_A^2 P_B^2$$

The bounds on the marginal effect sizes at each locus are specified as

$$\lambda_{AA}/\lambda_{Aa} \le \xi, \quad \lambda_{Aa}/\lambda_{aa} \le \xi,$$

$$\lambda_{BB}/\lambda_{Bb} \le \xi, \quad \lambda_{Bb}/\lambda_{bb} \le \xi$$

From these equations, once P_A, P_B, $P(Disease)$, and ξ are known, we can solve for the interaction effects (θ and α) using iterative numerical methods such that the bounded marginal effects are maximized at each locus.

2.4 Effects of Linkage Disequilibrium

In large scale disease association studies, often markers that are in LD with the disease loci are genotyped instead of the causative loci. To evaluate the performance of the information theoretic metrics under such situations, in a manner similar to [2] we consider specifying different values of LD between an observed marker $\tilde{L}_1$ and the corresponding unobserved causative locus L_1 and, similarly the observed marker $\tilde{L}_2$ and the corresponding unobserved causative locus L_2. By doing so, we evaluate the metrics only on the two observed markers that are in correlation individually with the unobserved disease associated markers.

2.5 Logistic Regression

Logistic Regression based association analysis is commonly employed to search for both single and multi-locus disease associations [9, 10]. The full single locus model under logistic regression modeling is [7, 8],

$$\log\left(\frac{r}{1-r}\right) = \mu + ax + dz$$

where r is the probability of each individual being a case, μ corresponds to the mean effect, the terms a and d correspond to the additive and dominance coefficient effects of the tested SNP variable, x and z are dummy variables with $x = 1$, $z = -0.5$ for one homozygote genotype (AA), $x = 0$, $z = 0.5$ for the heterozygote genotypes (Aa), and $x = -1$, $z = -0.5$ for the other homozygote type (aa). The log-likelihood ratio test is used to compare the full single locus model with the null model given by 0 values for both a and d and Bonferroni correction is commonly used to adjust the overall significance level.

Logistic Regression is also used to model the effect of genotypes and SNP × SNP interactions on the disease risk. We construct a fully saturated model by including terms that allow for the estimation of additive effects and dominance effects for each SNP locus, along with the inter-SNP additive and dominance interactions. The full interaction model, following Cordell's notation [8] is:

$$\log\left(\frac{r}{1-r}\right) = \mu + a_1x_1 + d_1z_1 + a_2x_2 + d_2z_2 + i_{aa}x_1x_2 + i_{ad}x_1z_2 + i_{da}z_1x_2 + i_{dd}z_1z_2$$

where r is the probability of each individual being a case, μ corresponds to the mean effect, the terms a_1, d_1, a_2, d_2 are the dominance and additive effect coefficients of the two SNPs, i_{aa}, i_{ad}, i_{da}, i_{dd} represent their interaction coefficients and x_i and z_i are dummy variables with $x_i = 1$, $z_i = -0.5$ for one homozygote

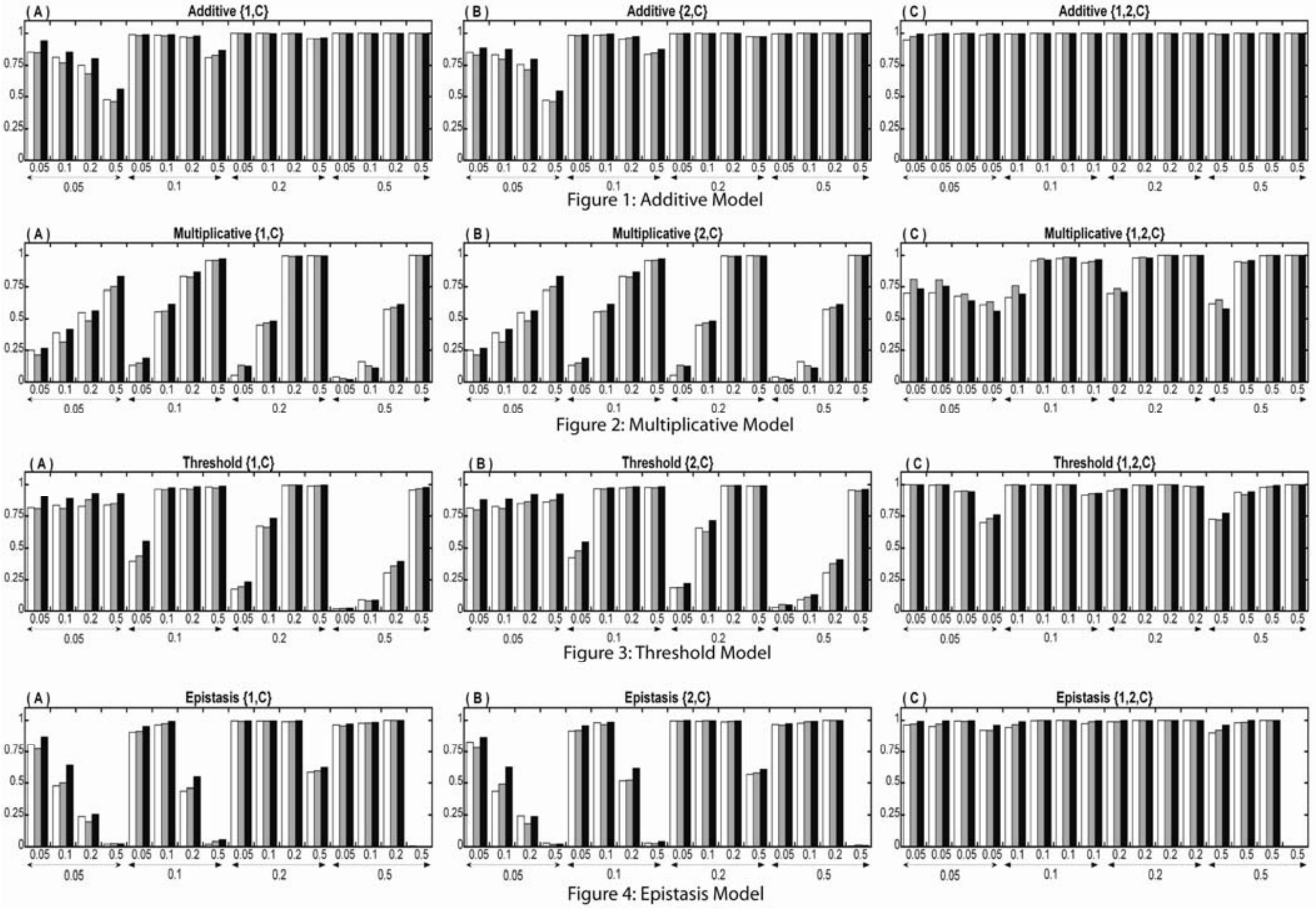


Figure 1: Additive Model

Figure 2: Multiplicative Model

Figure 3: Threshold Model

Figure 4: Epistasis Model

Figure 1-4: Power comparisons of the SNP combinations {1,*C*}, {2,*C*} and {1,2,*C*} for *PAI* (grey bars), *PAI*$_{pooled}$(black bars) and Logistic Regression (white bars) methods assuming that the typed loci are the causative loci for each of the four models. The powers are plotted against the allele frequencies at both the loci.

genotype (*AA* or *BB*), x_i = 0, z_i = 0.5 for the heterozygote genotypes (*Aa* or *Bb*), and x_i = -1, z_i = -0.5 for the other homozygote (*aa* or *bb*).

3. EXPERIMENTAL RESULTS

We have done detailed simulation of the four models with 100,000 loci, the two loci associated with the disease were SNPs 1 and 2 while the disease phenotype is denoted by *C*. For each gene-gene interaction model, a population of 100,000 individuals with genotypes in Hardy-Weinberg equilibrium and given allele frequencies was generated and a case-control study design was assumed. From the population, 1000 cases and 1000 controls were randomly selected. For convenience, the values 1, 2, and 3 were used to represent the homozygous for the major allele, the heterozygous genotypes and the homozygous for the minor allele, respectively. The value 1 was used to represent cases and 0 was used for controls. The disease prevalence was fixed at 0.01, the maximum marginal effect size (ξ) was varied as 1.5, 1.8 and 2.0, and the disease allele frequencies at each locus are varied as 0.05, 0.1, 0.2 and 0.5. We compare the power achieved by the information theoretic metrics *PAI* and *PAI*$_{pooled}$ with that obtained using logistic regression based analysis. Specifically, we are interested in the power achieved by the *PAI* (and *PAI*$_{pooled}$) values for the SNP combinations {1,*C*}, {2,*C*} and {1,2,*C*} since these are the combinations containing the implicated SNPs and the disease phenotype and their values are expected to be significantly higher than that of the combinations containing SNPs not involved in the disease process. High values for the combinations {1,*C*} and {2,*C*} shall enable the detection of the marginal effects at either loci while high values for the combination {1,2,*C*} denotes the presence of marginal effects at either loci or an interaction involving the two loci. Power calculations are performed using strategies similar to that described in [17]. We conducted 1000 independent simulations for each of the maximum marginal effect sizes and different allele frequencies at the two loci. The magnitudes of the maximum marginal effect sizes are chosen based on known results about complex diseases and previous works [2]. For each experiment with given allele frequencies the null distribution of the *PAI* and *PAI*$_{pooled}$ for each combination were obtained by calculating them on genotypes simulated with a marginal effect size of unity at each locus (θ=0) and their 95th percentile values were computed.

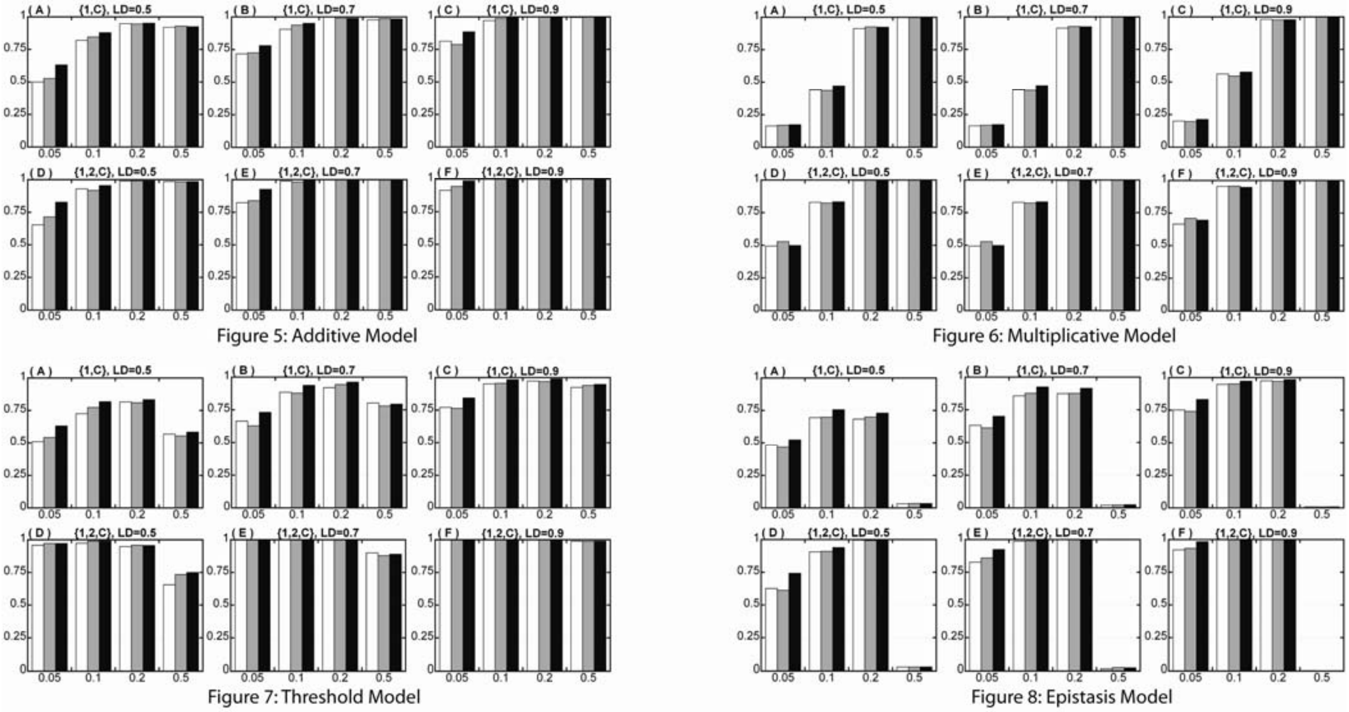


Figure 5: Additive Model

Figure 6: Multiplicative Model

Figure 7: Threshold Model

Figure 8: Epistasis Model

Figure 5-8: Power comparisons of the SNP combinations {1,*C*}, {2,*C*} and {1,2,*C*} for *PAI* (grey bars), PAI_{pooled}(black bars) and Logistic Regression (white bars) methods for LD of 0.5,0.7 and 0.9 between the each unobserved disease loci and the observed marker for each of the four models and assuming same allele frequency at either loci. The powers are plotted against the allele frequencies at both the loci.

A one-sided analysis was assumed since *PAI* and PAI_{pooled} are both positive and non-zero values indicate the presence of an interaction. The power was defined as the fraction of *PAI* (or PAI_{pooled}) values in the test distribution that were ≥ 95th percentile of the values in the corresponding null distribution. Similarly, a significance level of 0.05 was used for logistic regression.

A subset of the results for a low marginal effect size of 1.5 is presented in Figures 1 through 8 from our analysis such that not all of the three methods (*PAI*, PAI_{pooled} and logistic regression) compared head to head achieve high power for different values of the parameters considered. In each figure, the white, grey and black bars represent the powers of logistic regression, *PAI* and PAI_{pooled} metrics respectively for the combinations mentioned above. The x-axis enumerates the allele frequencies at both loci while the y-axis shows the powers achieved by each method. Figures 1-4 compares the powers for the information theoretic metrics and logistic regression for LD r^2 = 1.0 between each pair of the unobserved disease loci and the observed marker. We observed that the information theoretic methods achieve power comparable with the complex logistic regression based analysis (for both one and two loci models) and successfully catch both the marginal effects at individual loci and the interactions effects between the two loci. In particular, for many of the parameter settings, PAI_{pooled} achieves improved power than the other two methods in detecting the marginal effects in the Additive, Threshold and Epistasis models and interaction effects in the Additive and Epistasis models at lower allele frequencies. For all three methods, the power of the combination {1,2,*C*} is higher than that of the individual loci {1,*C*} or {2,*C*} since it detects the marginal effects at either loci or an interaction involving the two loci. Note that the power is near zero for all three methods for Epistasis model at allele frequencies of 0.5. This is because the marginal effect sizes remain very close to unity at each disease locus for this setting of the parameter values.

The effect of LD on the power was demonstrated in Figures 5-8 for r^2 = 0.5, 0.7 and 0.9. Since both the loci are simulated to have the same allele frequency, we show only one of the two loci (combination {1,*C*}) and the interaction combination {1,2,*C*}. As expected, the power of each of the combinations for all three methods increases with increase in LD between the observed marker and the unobserved disease loci. The effect of the allele frequencies is also pronounced for Additive and Multiplicative methods (Figures 5 and 6): increase in allele frequency increases power for both these methods, whereas for the other two models, power drops at the highest allele frequencies (Figures 7 and 8). Even at very low allele frequencies, information theoretic methods achieve better power than logistic regression for different values of LD, particularly in Additive, Threshold and Epistasis models. These results demonstrate the effectiveness of information theoretic methods in detecting various patterns of gene-gene interaction across a diverse range of simulation parameter settings.

We further evaluate performance of the information theoretic metrics using the data corresponding to problem 3 of the Genetic Analysis Workshop 15 (GAW15) . The data consist of 100 replicates simulated after the epidemiology and familial pattern of

Rheumatoid Arthritis (RA), a complex genetic disease in which it is hypothesized that several loci contribute to disease susceptibility. These data consist of 100 replicates of simulated data that are modeled after the rheumatoid arthritis data and contains: i) 730 microsatellite markers with an average spacing of 5 cM; ii) 9,187 SNPs distributed on the genome to mimic a 10K SNP chip set, and iii) 17,820 SNPs on chromosome 6. In addition RA affectation status, sex, age, smoking status, AntiCCP (anti-cyclic citrullinated peptide antibody)measure, IgM (immunoglobulin M) measure, severity, DR allele from father, DR allele from mother, age at onset, age at death are included as covariates (environmental variables). The AntiCCP and IgM measures are defined for the RA cases only.

We have used the 9187 SNPs distributed on all the chromosomes from the first of the replicates to evaluate the *PAI* and *PAI*$_{pooled}$ metrics using the proposed forward selection algorithm, and the remaining replicates were used to obtain the 95% confidence intervals for metrics for each combination of variables found by the algorithm. Three separate analyses was done:(i) with 9187 SNPs and Sex, Age, Smoking status as covariates and RA status as the phenotype variable, (ii) with 9187 SNPs and Sex, Age, Smoking status as covariates and AntiCCP measure as the phenotype variable, and (ii) with 9187 SNPs and Sex, Age, Smoking status as covariates and IgM measure as the phenotype variable. Although phase information was provided, we chose to not include it and treated the data as unphased genotype data. All the analyses were performed with algorithm input parameters ω= 50 and τ= 2. The Age, AntiCCP and IgM variables, which are continuous measures, were discretized by simple binning into five intervals of equal width.

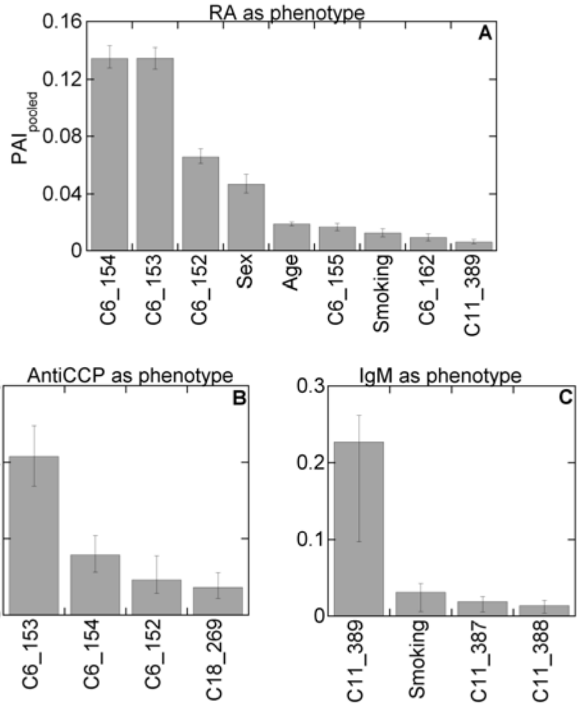


Figure 9: Results of the *Forward Selection Algorithm* using the *PAI*$_{pooled}$ metric as the search criterion using the three phenotypes. The x-axis shows the combinations obtained and the phenotypes are implicit in each combination. The confidence intervals are shown on each the metric values for each combination.

Figure 9-10 present the results for three analyses using the information theoretic metrics. The combinations in the figures were deemed significant since their confidence intervals did not span zero (zero indicates absence of an interaction). We find that both *PAI* and *PAI*$_{pooled}$ detects the SNPs and covariates that were simulated to have associations with the RA disease. In the figures, C{chromosome no.}_{SNP no.} is used as the naming convention for the markers. In figures 9A and 10A, the combinations consist of Locus DR and C (both SNPs C6_152-C6_155), Locus D (C6_162), Locus F (C11_387-C11_389) and the environmental variables Age, Sex and Smoking that had associations with the RA affection status in the simulated data set [22]. The simulation contained pronounced effects of DR on RA affection status and this was confirmed by the high values of *PAI* and *PAI*$_{pooled}$ which correspond to the DR locus. Locus D also has a direct effect on RA risk. Although it has a very low disease allele frequency (only

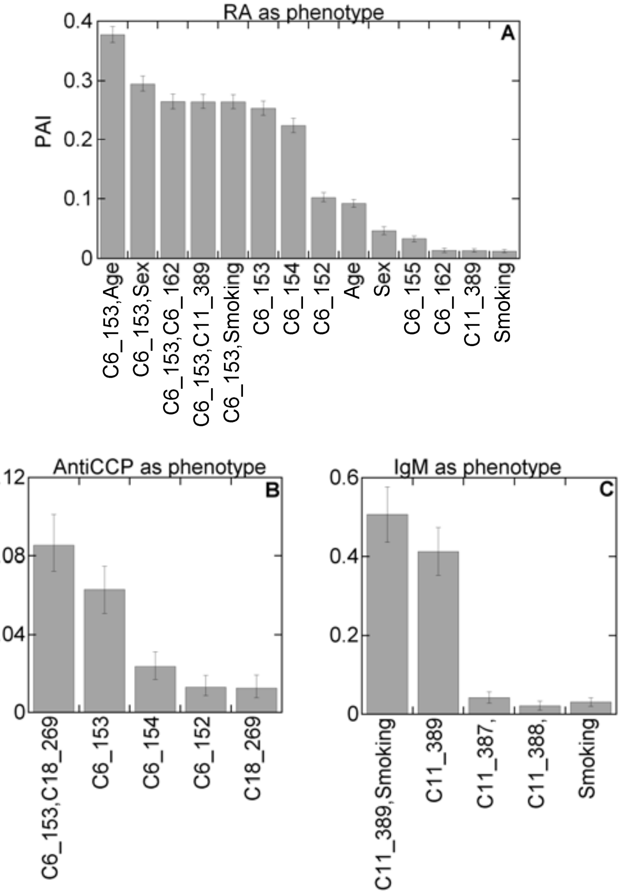


Figure 10: Results of the *Forward Selection Algorithm* using the *PAI* metric as the search criterion using the three phenotypes. The x-axis shows the combinations obtained and the phenotypes are implicit in each combination. The confidence intervals are shown on each the metric values for each combination.

0.0083, making minor allele homozygotes very rare), both the information theoretic metrics detected it successfully. Figure 9-10 B and C shows the combinations obtained with AntiCCP and IgM as phenotype variables, respectively. We successfully detect Locus DR and C (SNPs C6_152-C6_155) and Locus E (C18_269) with AntiCCP in figures 9B and 10B and the effects of Locus F (C11_387-C11_389) and Smoking on IgM in figures 9C and 10C using both the metrics.

These results demonstrate that our metrics are capable of analyzing a diverse range of epidemiological data sets containing evidences for gene-gene as well as gene-environment interactions.

4. DISCUSSION

We have presented two information theoretic measures and critically evaluated their performances using extensive simulation strategies that uses four different models of gene-gene statistical interaction. Detecting genes and environmental; factors interacting to increase the susceptibility to disease risk is a very challenging task due to many reasons, particularly due to the large size of the data and presence of confounding factors like linkage disequilibrium, presence of phenocopies and locus heterogeneity. Although regression based analyses can detect disease associated loci, they are computationally very intensive that grows exponentially with the number of loci considered in the model. Information theoretic methods have high power in detecting gene-gene interactions and have the advantage of being simpler and computationally faster. We do not intend to claim that our proposed metrics are the best since the properties of each method depends greatly on the factors like sample size, type of the interactions involved, density of the genetic maps, availability of phase information etc. However the metrics are appealing not only because they performed well in the experiments and the GAW15 data, but also because they are flexible and can be used when the genetic and environmental variables have different numbers of classes or when the phenotype has more than two classes. This means that SNP and microsatellite markers can be analyzed together if necessary. Also they are naturally extensible to study models with more than two loci and environmental variables.

We have used simulated data modeled after real disease data. The GAW15 data set was sufficiently rich and complex because it was modeled based on a real rheumatoid arthritis data set and the simulation details were available. The major advantage of using such data is that the ground truth is established during the simulation. For future work, we would like to test our metrics on several publicly available SNP data sets and also using more interaction models, particularly with models containing complex gene-gene and gene-environment interactions involving 3 or more loci in a manner similar to our simulations in [17]. Also given a large number of markers in large scale studies, some filtering approaches can be used as a preprocessing step to remove confounders caused by effects such as linkage disequilibrium. Additional biological knowledge e.g. gene expression and biological pathway information can also be incorporated along with the proposed metrics to make the search in the forward selection algorithm more biologically oriented.

5. References

[1] Hirschhcorn, J.N. and M.J. Daly, *Genome-wide association studies for common diseases and complex traits.* Nat Rev Genet, 2005. 6: p. 95-108.

[2] Marchini, J., P. Donnelly, and L.R. Cardon, *Genome-wide strategies for detecting multiple loci that influence complex diseases.* Nat Genet, 2005. 37: p. 413-417.

[3] Culverhouse, R., et al., *A perspective on epistasis: limits of models displaying no main effects.* Am J Hum Genet, 2002. 70(461-471).

[4] Hoh, J. and J. Ott, *Mathematical multi-locus approaches to localising complex human trait genes.* Nat Rev Genet, 2003. 4: p. 701-709.

[5] Hallgrímsdóttir, I.B. and D.S. S Yuster, *A complete classification of epistatic two-locus models.* BMC Genetics, 2008. 9(17).

[6] Li, W. and J. Reich, *A complete enumeration and classification of two-locus disease models.* Hum Hered, 2000. 50: p. 334-339.

[7] Barhdadi, A. and M.-P. Dubé, *Two-stage strategies to detect gene × gene interactions in case-control data.* BMC Proceedings, 2007. 1 (Suppl 1):S135.

[8] Cordell, H.J., *Epistasis: what it means, what it doesn't mean, and statistical methods to detect it in humans.* Human Molecular Genetics, 2002. 11(20): p. 2463-2468.

[9] Musani, S.K., D. Shriner, and N. Liu, *Detection of Gene x Gene Interactions in Genome-Wide Association Studies of Human Population Data.* Hum Hered, 2007. 63: p. 67-84.

[10] North, B.V., D. Curtis, and P.C. Sham, *Application of Logistic Regression to Case-Control Association Studies Involving Two Causative Loci.* Hum Hered, 2005. 59: p. 79-87.

[11] Liu, Z. and L. T., *Multilocus LD measure and tagging SNP selection with generalized mutual information.* Genetic Epidemiology 2005. 29: p. 353-364.

[12] Moore, J.H., et al., *A flexible computational framework for detecting, characterizing, and interpreting statistical patterns of epistasis in genetic studies of human disease susceptibility.* Journal of Theoretical Biology, 2006. 241: p. 252-261.

[13] Anderson, E.C. and E.A. Thompson, *A model-based method for identifying species hybrids using multilocus genetic data.* Genetics, 2002. 160: p. 1217-1229.

[14] Rosenberg, N.A., et al., *Informativeness of genetic markers for inference of ancestry.* Am J Hum Genet, 2003. 73: p. 1402-1422.

[15] Bhasi, K., et al., *VizStruct for visualization of genome-wide SNP analyses.* Bioinformatics, 2006. 22(1569-1576).

[16] Bhasi, K., et al., *Information-theoretic identification of predictive SNPs and supervised visualization of genome-wide association studies.* Nucleic Acids Res, 2006. 34(e101).

[17] Chanda, P., et al., *Information-theoretic metrics for visualizing gene-environment interactions.* Am J Hum Genet, 2007. 81: p. 939-963.

[18] Zhao, J., E. Boerwinkle, and M. Xiong, *An entropy-based statistic for genomewide association studies.* Am J Hum Genet, 2005. 77: p. 27-40.

[19] *Genetic Analysis Workshop 15.* (http://www.gaworkshop.org/gaw15data.htm) 2006.

[20] Shannon, C.E., *A mathematical theory of communication.* Bell System Technical Journal, 1948. 27: p. 379-423,623-656.

[21] Han, T.S., *Multiple mutual informations and multiple interactions in frequency data.* Information and Control, 1980. 46(1): p. 26-45.

[22] Miller, M.B., et al., *Genetic Analysis Workshop 15: Simulation of a complex genetic model for rheumatoid arthritis in nuclear families including a dense SNP map with linkage disequilibrium between marker loci and trait loci.* BMC Genetics, 2007.

A fast, large-scale learning method for protein sequence classification

Pavel Kuksa, Pai-Hsi Huang, Vladimir Pavlovic*
Department of Computer Science
Rutgers University
Piscataway, NJ 08854
{pkuksa;paihuang;vladimir}@cs.rutgers.edu

ABSTRACT

Motivation: Establishing structural and functional relationships between sequences in the presence of only the primary sequence information is a key task in biological sequence analysis. This ability can be critical for tasks such as making inferences of the structural class of unannotated proteins when no secondary or tertiary structure is available. Recent computational methods based on profile and mismatch neighborhood kernels have significantly improved one's ability to elucidate such relationships. However, the need for additional reduction in computational complexity and improvement in predictive accuracy hinders the widespread use of these powerful computational tools.
Results: We present a new general approach for sequence analysis based on a class of efficient string-based kernels, sparse spatial sample kernels (SSSK). The approach offers state-of-the-art accuracy for sequence classification, low computational cost, and scales well with the size of sequence databases, in both supervised and semi-supervised learning settings. Application of the proposed methods to a remote homology detection and a fold recognition problems yields performance equal to or better than existing state-of-the-art algorithms. We also demonstrate the benefit of the spatial information and multi-resolution sampling for achieving this accuracy and for discriminative sequence motif discovery. The proposed methods can be applied to very large partially-labeled databases of protein sequences because of low computational complexity and show substantial improvements in computing time over the existing methods.
Availability: Supplementary data and Matlab/C codes are available at http://seqam.rutgers.edu/spatial-kernels/
Contact: vladimir@cs.rutgers.edu

Categories and Subject Descriptors

I.2 [**Artificial Intelligence**]: Learning; I.5 [**Pattern Recognition**]: Applications; I.5.2. [**Pattern Recognition**]: Design Methodology

General Terms

Algorithms, Design, Measurement, Performance, Experimentation

Keywords

sequence classification, large-scale semi-supervised learning, string kernels

1. INTRODUCTION

Classification of protein sequences into structural or functional classes is a fundamental problem in computational biology. With the advent of large-scale sequencing techniques, experimental elucidation of an unknown protein sequence function becomes an expensive and tedious task. Currently, there are more than 61 million DNA sequences in GenBank [3], and approximately 349,480 annotated and 5.3 million unannotated sequences in UNIPROT [2], making development of computational aids for sequence annotation a critical and timely task. In this work we focus on protein sequence classification problems using only the primary sequence information. While additional sources of information, such as the secondary or tertiary structure, may lessen the burden of establishing the homology, they may often be unavailable or difficult to acquire for new putative proteins.

Early approaches to computationally-aided homology detection, such as BLAST [1] and FASTA [22], rely on aligning the query sequence to a database of known sequences (pairwise alignment). Later methods, such as profiles [7] and profile hidden Markov models (profile HMM) [6], collect aggregate statistics from a group of sequences known to belong to the same family. Such generative approaches only make use of positive training examples, while the discriminative approaches attempt to capture the distinction between different classes by considering both positive and negative examples. In many sequence analysis tasks, the discriminative methods such as kernel-based [25] machine learning methods provide the most accurate results [4, 13, 17, 24]. Several types of kernels for protein homology detection have been proposed over the last decade. In [11], Jaakkola *et al.* proposed *SVMFisher*, derived from probabilistic models. Leslie *et al.* in [17] proposed a class of kernels that operate directly on strings and derive features from the sequence content. Both classes of kernels demonstrated improved discriminative power over methods that operate under generative settings.

*corresponding author

BIOKDD '08 Las Vegas, Nevada, USA

Remote homology detection and fold recognition problems are typically characterized by few *positive training* sequences accompanied by a large number of negative training examples. Lack of positive training examples may lead to suboptimal classifier performance, therefore making training set expansion necessary. However, enlarging the training set by experimentally labeling the sequences is costly leading to the need for leveraging *unlabeled data* to refine the decision boundary. The profile kernel [14] and the mismatch neighborhood kernel [26] both use large unlabeled datasets and show significant improvements over the sequence classifiers trained under the supervised setting. Nevertheless, the promising results can be offset by a significant increase in computational complexity, thus hindering use of such powerful computational tools on very large sequence data sets.

In this study, we present a general approach for efficient classification of biological sequences, based on a class of string kernels, the *sparse spatial sampling kernels (SSSK)*. The proposed method effectively models sequences under complex biological transformations such as multiple mutations, insertions, and deletions by multi-resolutional sampling. Under the SSSK, feature matching is independent of the size of the alphabet set, which ensures low computational cost. Such characteristics open the possibility of analyzing very large unlabeled datasets under the semi-supervised setting with modest computational resources. Compared to the existing string kernels, the SSSK provide a richer representation for sequences by explicitly encoding the information on spatial configuration of features within the sequences, leading to discovery of sequence motifs. The proposed methods perform better and run substantially faster than existing state-of-the-art kernel-based algorithms [14, 13, 26].

2. BACKGROUND

In this section, we briefly review previously published state-of-the-art methods for protein homology detection. We denote the alphabet set as Σ in the whole study. Given a sequence X the *spectrum-k* kernel [16] and the *mismatch(k,m)* kernel [17] induce the following $|\Sigma|^k$-dimensional representation for the sequence:

$$\Phi(X) \quad = \quad \left(\sum_{\alpha \in X} I(\alpha, \gamma)\right)_{\gamma \in \Sigma^k}, \qquad (1)$$

where under the spectrum-k kernel, $I(\alpha, \gamma) = 1$ if $\alpha = \gamma$ and under the mismatch(k,m) kernel, $I(\alpha, \gamma) = 1$ if $\alpha \in N(\gamma, m)$, where $N(\gamma, m)$ denotes the *mutational neighborhood* induced by the k-mer γ for up to m mismatches.

Both the spectrum-k and the mismatch(k,m) kernel directly extract string features based on the observed sequence, X. On the other hand, the profile kernel, proposed by Kuang *et al.* in [13], builds a profile [7] P_X and uses a similar $|\Sigma|^k$-dimensional representation, derived from the profile:

$$\Phi^{profile(k,\sigma)}(X) = \left(\sum_{i=1\cdots(T_{P_X}-k+1)} I(P_X(i,\gamma) < \sigma)\right)_{\gamma \in \Sigma^k} \qquad (2)$$

where $P_X(i, \gamma)$ denotes the cost of *locally* aligning the k-mer γ to the k-length segment starting at the i^{th} position of P_X, σ a pre-defined threshold and T_{P_X} the length of the profile. Explicit inclusion of the amino acid substitution process allows both the mismatch and the profile kernels to significantly outperform the spectrum kernel and demonstrate state-of-the-art performance under both supervised and semi-supervised settings [26, 13] for the protein sequence classification tasks. However, such method of modeling substitution process induces a k-mer mutational neighborhood that is exponential in the size of the alphabet set during the matching step for kernel evaluation; for the mismatch(k,m) kernel, the size of the induced k-mer neighborhood is $k^m|\Sigma|^m$ and for the profile(k,σ) kernel, the maximum size of the mutational neighborhood is dependent on the threshold parameter σ and the shape of the profile. Increasing m or σ to model multiple mutations will incur high complexity for computing the kernel matrix hence hindering the use of such powerful tools.

The promising results of the profile kernel shown in [13] rely on the usage of a large unlabeled sequence database, such as the *non-redundant (NR)* data set, for estimation of profiles. On the other hand, for the mismatch string kernel, Weston *et al.* propose to use the *sequence neighborhood kernel* to leverage the unlabeled sequences in [26].

2.1 The sequence neighborhood kernel

The sequence neighborhood kernels take advantage of the unlabeled data using the process of neighborhood induced regularization. Let $\Phi^{orig}(X)$ be the original representation of sequence X. Also, let $N(X)$[1] denote the *sequence neighborhood* of X (a set of sequences neighboring X). Weston *et al.* proposed in [26] to re-represent X using:

$$\Phi^{new}(X) \quad = \quad \frac{1}{|N(X)|} \sum_{X' \in N(X)} \Phi^{orig}(X'). \qquad (3)$$

Under the new representation, the kernel value between the two sequences X and Y becomes:

$$K^{nbhd}(X, Y) \quad = \quad \sum_{X' \in N(X), Y' \in N(Y)} \frac{K(X', Y')}{|N(X)||N(Y)|}. \qquad (4)$$

Note that under such settings, all *training* and *testing* sequences will assume a new representation, whereas in a traditional semi-supervised setting, unlabeled data are used during the *training phase only.* The authors choose the mismatch representation for the sequences and show that the discriminative power of the classifiers improves significantly once information regarding the neighborhood of each sequence is available. Both the profile kernel and the mismatch neighborhood kernel show very promising results and demonstrate state-of-the-art performance in various protein sequence classification tasks. However, the exponential size of the incurred k-mer mutational neighborhood makes large-scale semi-supervised learning under the mismatch representation very computationally demanding.

3. THE SPARSE SPATIAL SAMPLE KERNELS

In this section, we present a new class of string kernels, the *sparse spatial sample kernels (SSSK)*, that effectively model complex biological transformations (such as highly diverse mutation, insertion and deletion processes) and can be efficiently computed. The SSSK family of kernels, parametrized

[1] We will discuss how to construct $N(X)$ in Section 4.1.

by three positive integers, assumes the following form:

$$K^{(t,k,d)}(X,Y) = \sum_{\substack{(a_1,d_1,\ldots,d_{t-1},a_t)\\ a_i\in\Sigma^k, 0\le d_i<d}} \begin{array}{r} C(a_1,d_1,\cdots,a_{t-1},d_{t-1},a_t|X)\cdot \\ C(a_1,d_1,\cdots,a_{t-1},d_{t-1},a_t|Y) \end{array}, \quad (5)$$

where $C(a_1, d_1, \cdots, a_{t-1}, d_{t-1}, a_t|X)$ denotes the number of times we observe substring $a_1 \overset{d_1}{\leftrightarrow} a_2, \overset{d_2}{\leftrightarrow}, \cdots, \overset{d_{t-1}}{\longleftrightarrow} a_t$ (a_1 separated by d_1 characters from a_2, a_2 separated by d_2 characters from a_3, etc.) in the sequence X. This is illustrated in Figure 1.

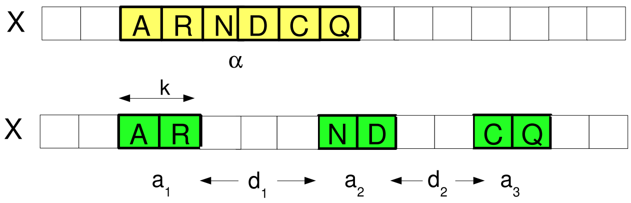


Figure 1: Contiguous k-mer feature α of a traditional spectrum/mismatch kernel (top) contrasted with the sparse spatial samples of the proposed kernel (bottom).

The new kernel implements the idea of sampling the sequences at different resolutions and comparing the resulting spectra; similar sequences will have similar spectrum at one or more resolutions. This takes into account possible mutations, as well as insertions/deletions[2]. Each sample consists of t spatially-constrained probes of size k, each of which lie no more than d positions away from its neighboring probes. In the proposed kernels, the parameter k controls the individual probe size, d controls the locality of the sample, and t controls the cardinality of the sampling neighborhood. In this work, we use short samples of size 1 (i.e., $k = 1$), and set t to 2 (i.e. features are pairs of monomers) or 3 (i.e. features are triples.)

The proposed sample string kernels not only take into account the feature counts (as in the family of spectrum kernels [16, 17] and gapped/subsequence kernels [15]), but also include spatial configuration information, *i.e.* how the features are positioned in the sequence. This is in contrast to the gapped or subsequence kernels where such information is not present. The spatial information can be critical in establishing similarity of sequences under complex transformations such as the evolutionary processes in protein sequences. The addition of the spatial information experimentally demonstrates very good performance, even with very short sequence features (*i.e. k=1*), as we will show in Section 4.

The use of short features can also lead to significantly lower computational complexity of the kernel evaluations. The dimensionality of the features induced by the proposed kernel is $|\Sigma|^t d^{t-1}$ for our choice of $k = 1$. As a result, for triple-(1,3) ($k = 1$, $t = 3$, $d = 3$) and double-(1,5) ($k = 1$,$t = 2$,$d = 5$) feature sets, the dimensionalities are $72,000$ and $2,000$, respectively, compared to $3,200,000$ for the spectrum-(k) [16], mismatch-(k,m) [17], and profile(k,σ) kernels with the common choice of $k = 5$. The low dimensionality of the feature sets ensures efficient computation. The proposed kernels can be efficiently computed using sorting and counting. To compute the kernel values, we first extract the features from the sequences and sort the extracted features in linear time using counting sort. Finally we count the number of distinct features and update the kernel matrix. For N sequences with the longest length n and u distinct features, computing the NxN kernel matrix takes linear $O(dnN + min(u, dn)N^2)$ time. Similar to the gapped kernels [15], the complexity for kernel evaluation is also independent of the size of the alphabet set. We provide a comprehensive comparison of the computational complexity and running times with other kernel methods in Section 5.

3.1 SSSK under Semi-supervised learning setting

The SSSK can also be extended to accommodate unlabeled data, similar to the approach presented by Weston *et al.* in [26]. Under the semi-supervised setting with unlabeled sequences, direct use of Equation 4 for computation of the refined kernel values between sequences X and Y requires $|N(X)| \times |N(Y)|$ kernel evaluations, i.e. quadratic running time in the size of the neighborhood. On the other hand, use of Equation 3 requires explicit representation of the sequences which can be problematic when the dimensionality of the feature space is high. As a result, performing such *smoothing* operation over the *mismatch kernel representation* is computationally intensive, as noted in [26, 14].

Equation 3 lends a useful insight into the complexity of the smoothing operation. For any explicit representation $\Phi(X)$, its smoothed version can be computed in time linear in the size of the neighborhood $|N(X)|$, therefore the smoothed kernel can also be evaluated in time linear in the neighborhood size. However, the smoothed representation in case of the mismatch kernel cannot be computed explicitly due to its exponential length. On the other hand, for the proposed kernels (doubles and triples) the smoothed representations can be computed explicitly, if desired.

In our experiments, we do not compute the explicit representation and instead use implicit computations over induced representations. For each neighborhood $N(X)$, a set of sequences neighboring X, we first sort the features (e.g. doubles of characters) and then obtain counts for distinct features to evaluate the kernel. This leads to a low space and time complexity for the kernel computations. The presence of mismatches, however, prevents one from applying the same approach under the mismatch representation.

4. EXPERIMENTAL RESULTS

We present experimental results for the remote homology detection under the supervised setting on the SCOP dataset in Section 4.2 and the results for large-scale semi-supervised homology detection in Section 4.3. In Section 4.4, we compare iterative (PSI-BLAST) and non-iterative (BLAST) methods for neighborhood construction. Finally, we present experimental results for remote fold recognition in Section 4.5.

4.1 Settings, parameters and performance measures

We evaluate all methods using the *Receiver Operating Characteristic* (ROC) and ROC-50 [8] scores. The ROC-50

[2] We discuss how insertions and deletions are modeled in Section 5.

score is the (normalized) area under the ROC curve computed for up to 50 false positives. With a small number of positive testing sequences and a large number of negative testing sequences, the ROC-50 score is typically more indicative of the prediction accuracy of a homology detection method than the ROC score.

In all experiments, we normalize kernel values $K(X,Y)$ using

$$K'(X,Y) \quad = \quad \frac{K(X,Y)}{\sqrt{K(X,X)K(Y,Y)}} \qquad (6)$$

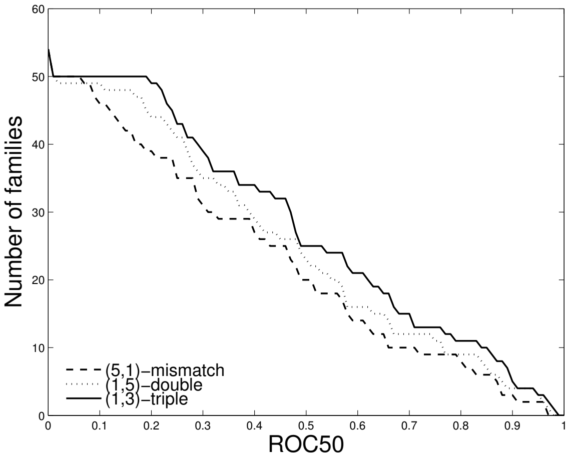
to remove the dependency between the kernel value and the sequence length. To perform our experiments, we use an existing SVM implementation from a standard machine learning package SPIDER[3] with the default parameters. In the semi-supervised experiments, we use kernel smoothing (Equation 4) as in [26]. For each sequence X, to construct the sequence neighborhood $N(X)$ we query the unlabeled dataset using 2 iterations of PSI-BLAST and recruit the sequences with e-values ≤ 0.05 as neighbors of X (*i.e.* $N(X) = \{X' : eValue(X,X') \leq 0.05\}$). To adhere to the true semi-supervised setting, we remove *all sequences in the unlabeled datasets that are identical to any test sequence.*

For all experiments, we compare with the state-of-the-art classifiers using the triple(1,3) ($k = 1$,$t = 3$,$d = 3$) and the double(1,5) ($k = 1$,$t = 2$,$d = 5$) feature sets.

4.2 SCOP Dataset

We use the dataset published in [26] to perform our experiments. The dataset contains 54 target families from SCOP 1.59 [19] with 7,329 isolated domains. Our experimental setup is the same as that of Jaakkola [11, 13]. In each of the 54 experiments, to simulate the remote homology problem one of the families is completely held out for testing (i.e. the classifiers are tested on the the sequences from unseen families). Different instances of this dataset have been used as a gold standard for protein remote homology detection in various studies [10, 18, 16, 17, 13].

We compare the performance of our proposed methods with previously published state-of-the-art methods [18, 17] under the supervised learning setting in Table 1. We also show the dimensionality of the induced features and the observed experimental running times, measured on a 2.8GHz CPU, for constructing the 7329x7329 kernel matrix[4]. It is clear from the table that the proposed kernels (doubles and triples) not only show significantly better performance than existing methods, but also require substantially less computational time. Also, as can be seen from the comparison with the gapped kernels, the addition of the spatial information substantially improves the classification performance. We also show the ROC-50 plot in Figure 2. In the plot, the horizontal axis corresponds to the ROC-50 scores and the vertical axis denotes the number of experiments, out of 54, with an equivalent or higher ROC-50 score. For clarity, we do not display the plot for every method. Our results clearly indicate that both double and triple kernels outperform all other methods.



th

Figure 2: Comparison of the performance (ROC50) in the supervised setting. Spatial kernels (triples and doubles) outperform other supervised methods.

Table 1: Comparison of the performance on the SCOP 1.59 dataset under the supervised setting.

Method	ROC	ROC50	# dim.	Time (s)
(5, 1)-mismatch	0.8749	0.4167	3200000	938
SVM-pairwise†	0.8930	0.4340	-	-
gapped(6,2)[15]	0.8296	0.3316	400	55
gapped(7,3)	0.8540	0.3953	8000	297
(1,5) double	0.8901	0.4629	2000	54
(1,3) triple	**0.9148**	**0.5118**	72000	112

†: directly quoted from [18]

4.3 Large-Scale semi-supervised experiments

In this section, we perform the semi-supervised experiments on three unlabeled datasets:the *non-redundant* (NR) dataset, Swiss-Prot[5], and PDB[6]. Table 2 summarizes the main characteristics of the unlabeled datasets used in this study. The second column shows the size of the unlabeled datasets and the third column shows the mean, median and maximum number of neighbors per sequence recruited using PSI-BLAST with the corresponding unlabeled dataset.

Table 2: Number of neighboring sequence recruited using PSI-BLAST with various unlabeled datasets(mean/median/max).

Dataset	# Seq	# Neighbors
Swiss-Prot	101602	56/28.5/385
PDB	116697	16/5/334
NR	534936	114/86/490

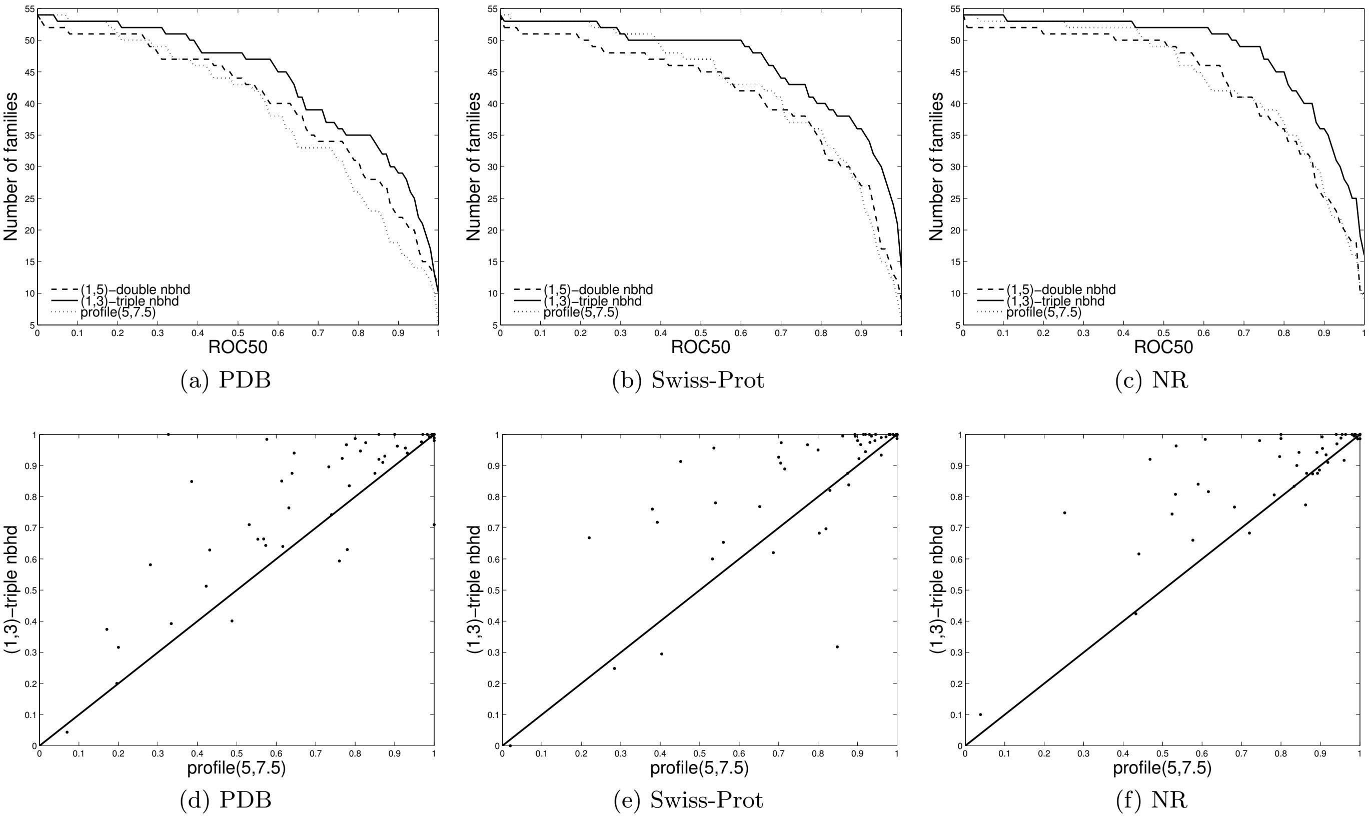
We perform all semi-supervised experiments on a 2.8GHz processor with 2GB of memory. Computation of the mismatch neighborhood kernels is computationally demanding and typically cannot be accomplished on a single machine for anything but relatively small unlabeled datasets. Therefore, the results for the mismatch neighborhood kernel can only

[3] http://www.kyb.tuebingen.mpg.de/bs/people/spider

[4] The code used for evaluation of the competing methods has been highly optimized to perform on par or better than the published spectrum/mismatch code. We also used the code provided by the authors of the competing methods.

[5] We use the same version as the one employed in [26] for comparative analysis of performance.

[6] As of Dec. 2007.



(a) PDB (b) Swiss-Prot (c) NR

(d) PDB (e) Swiss-Prot (f) NR

Figure 3: In the upper panel, we show the ROC-50 plots of three different features using PDB, Swiss-Prot and NR databases as unlabeled datasets, respectively. In the lower panel, we show the scatter-plot of ROC-50 scores of the triple-(1,3) kernel (vertical) and the profile(5,7.5) kernel (horizontal). Any point above the diagonal line in the figures (d),(e),(f) indicates better performance for the triple-(1,3) kernel.

Table 3: Statistical significance (p-values of the Wilcoxon signed rank test) of the observed differences between pairs of methods (ROC-50 scores) on unlabeled datasets. Triple denotes the triple-(1,3) neighborhood kernel, double denotes the double-(1,5) neighborhood kernel, mismatch denotes the mismatch(5,1) neighborhood kernel, and profile denotes the profile(5,7.5) kernel.

PDB			
	double	triple	profile
double	-	1.017e-01	4.762e-02
triple	1.017e-01	-	7.666e-06
profile	4.762e-02	7.666e-06	-
Swiss-Prot			
	double	triple	profile
double	-	9.242e-05	4.992e-01
triple	9.242e-05	-	2.419e-04
profile	4.992e-01	2.419e-04	-
NR			
	double	triple	profile
double	-	8.782e-06	9.762e-01
triple	8.782e-06	-	7.017e-06
profile	9.762e-01	7.017e-06	-

be shown using the previously published summary statistics [26] on Swiss-Prot, a moderately populated sequence database. In the upper panel of Figure 3, we show the ROC-50 plots of the double-(1,5) neighborhood, triple-(1,3) neighborhood, and profile(5,7.5) kernels using PDB (first column), Swiss-Prot (second column), and NR (third column) sequence databases as the unlabeled datasets. The ROC-50 curves of the triple-(1,3) neighborhood kernel on all unlabeled datasets consistently outperform the other two kernels. Furthermore, the performance of the double-(1,5) neighborhood kernel is on par with that of the profile(5,7.5) kernel. In the lower panel, we show the scatterplots of the ROC-50 scores of the triple-(1,3) kernel and the profile(5,7.5) kernel. Any point falling above the diagonal line in the figures indicates better performance of the triple-(1,3) kernel over the profile(5,7.5) kernel. As can be seen from these plots, the triple kernel outperforms the profile kernel on all three datasets (43/37/34 wins and 4/5/10 ties on PDB, Swiss-Prot, and NR datasets, respectively).

We also show the statistical significance of the observed differences between pairs of methods on various unlabeled datasets in Table 3. All the entries in the table are the p-values of the Wilcoxon signed rank test using the ROC-50 scores. For each unlabeled dataset, we highlight the method that has the best overall performance. The triple-(1,3) kernel consistently outperforms all other kernels, with high sta-

Table 4: The overall prediction performance of all compared methods over various unlabeled datasets.

PDB	ROC	ROC50
double-(1,5) neighborhood	.9599	.7466
triple-(1,3) neighborhood	**.9717**	**.8240**
profile(5,7.5)	.9511	.7205
Swiss-Prot		
double-(1,5) neighborhood	.9582	.7701
triple-(1,3) neighborhood	**.9732**	**.8605**
profile(5,7.5)	.9709	.7914
mismatch nbhd†	.955	.810
NR		
double-(1,5) neighborhood	.9720	.8076
triple-(1,3) neighborhood	**.9861**	**.8944**
profile(5,7.5)-2 iterations	.9734	.8151
profile(5,7.5)-5 iterations‡	.984	.874
profile(5,7.5)-5 iter. with secondary structure‡	.989	.883

†:directly quoted from [26]

‡:directly quoted from [14]

tistical significance.

Finally, we show the overall prediction performance of all compared methods over various unlabeled datasets in Table 4. For each unlabeled dataset, we highlight the best ROC and ROC-50 scores; on all datasets, the triple-(1,3) neighborhood kernel achieves the best performance. Furthermore, we achieve such performance by only 2 PSI-BLAST iterations. For example, the triple-(1,3) neighborhood kernel with 2 PSI-BLAST iterations outperforms the profile(5,7.5) kernel with 5 PSI-BLAST iterations. We also note that the performance of our kernels is achieved using primary sequence information only. However, as shown in the table, the triple-(1,3) kernel still outperforms the profile(5,7.5) kernel with the added secondary structure information. Such higher order information (e.g. secondary structure), if available and desirable, can be easily included in our feature set.

4.4 Non-iterative neighborhood construction

Performing the iterative search using PSI-BLAST for neighborhood construction is computationally demanding and consumes large portion of the overall running time of the methods. In this section, we present the results obtained using BLAST search only. The use of BLAST only requires a single pass over the unlabeled sequence database and therefore requires substantially less computational time and resources compared to the iterative multi-pass PSI-BLAST search. We use the same threshold on e-value ($\leq .05$) to recruit the neighboring sequences to form the neighborhood sets $N(X)$, for each query sequence X. We compare the performance of the classifiers estimated using *iterative* (PSI-BLAST) and *non-iterative* (BLAST) sequence neighborhood construction in Table 5. First, we observe that, as the size of the unlabeled sequence database increases, the margins between the performance of the iterative and non-iterative sequence neighborhood construction procedures narrows. Second, we observe that, using only BLAST, the triple(1,3) neighborhood kernel already outperforms the profile(5,7.5) kernel, constructed with 2 PSI-BLAST iterations, and also shows comparable performance with the profile(5,7.5) kernel, constructed with 5 PSI-BLAST iterations on the non-redundant data set (Table 4). Finally, compared with the number of neighbors recruited using PSI-BLAST in Table 2, we observe a three-fold reduction when using non-iterative (BLAST) neighborhood construction procedure. Such reduction in neighborhood size enables faster training and classification as well as reduces storage requirements for the support vectors.

Table 5: Comparison of performance using iterative (PSI-BLAST) and non-iterative (BLAST) sequence neighborhood construction procedures. The performance is measured using the triple(1,3) feature set.

Data set	PSI-BLAST		BLAST		
	ROC	ROC50	ROC	ROC50	#neighbors with BLAST
PDB	0.9691	0.8240	0.9557	0.7535	6/3/95
Swiss-Prot	0.9732	0.8605	0.9640	0.8144	16/9/177
NR	0.9861	0.8944	0.9787	0.8647	40/23/232

4.5 Preliminary results for fold prediction

For the fold recognition task, we use a challenging dataset designed by Ding *et al.* [7] in [5], used as a benchmark in many studies. The data set contains sequences from 27 folds divided into two *independent* sets, such that the training and test sequences share less than 35% sequence identities and within the training set, no sequences share more than 40% sequence identities.

We compare the performance of our methods under supervised and semi-supervised settings with previously published methods on Ding and Dubchak benchmark data set in Table 6. As can be seen from the table, our spatial kernels achieve higher overall performance compared to the state-of-the-art classifiers.

5. DISCUSSION

We next compare our family of kernels with other kernel methods and discuss computational aspects of the methods. We also demonstrate how our method discovers discriminative short sequence motifs.

5.1 Complexity Comparison

We first compare computational complexity of the methods in Table 7 and show the observed running times. Running time measurements for our methods are done on a 2.8GHz CPU. For supervised experiments, we compute the full 7329x7329 kernel matrix for all methods. For the semi-supervised setting (neighborhood kernels), we report average running time on the datasets used (i.e. PDB, Swiss-Prot, and non-redundant (NR) databases.) Both the mismatch neighborhood and the profile kernels have higher complexity compared to the sample kernels due to the exponential neighborhood size. The cardinalities of the mismatch and profile neighborhoods are $O(k^m|\Sigma|^m)$, where $k \geq 5$, and $|\Sigma| = 20$, compared to a much smaller feature space size of $d^{t-1}|\Sigma|^t$ for the sample kernels, where t is 2 or 3, and d is 3 or 5, respectively. This complexity difference leads to order-of-magnitude improvements in the running times of the sample kernels over the mismatch and profile kernels. The difference is even more pronounced when kernel smoothing is used under a semi-supervised setting. The neighborhood mismatch

[7] http://ranger.uta.edu/∼chqding/bioinfo.html

Table 6: Comparison on Ding and Dubchak benchmark data set

Method	Error	Top 5 Error	Balanced Error	Top 5 Balanced Error	Recall	Top 5 Recall	Precision	Top 5 Precision	F1	Top5 F1
Supervised										
SVM(D&D)†	-	-	56.5	-	-	-	-	-	-	-
Mismatch(5,1)	51.17	22.72	53.22	28.86	46.78	71.14	**90.52**	**95.25**	61.68	81.45
Double(1,5)	44.13	23.50	46.19	23.92	53.81	76.18	61.90	79.85	57.57	77.97
Triple (1,3)	**41.51**	**18.54**	**44.99**	**21.09**	**55.01**	**78.91**	80.42	89.19	**65.33**	**83.74**
Semi-supervised (Non-redundant data set)										
Profile(5,7.5)	31.85	15.14	32.17	16.73	67.83	83.27	**89.49**	**94.9**	77.16	88.71
Double(1,5)	28.72	14.99	24.74	**11.6**	75.26	**88.4**	76.02	86.86	75.63	87.62
Triple(1,3)	**24.28**	**12.79**	**22.38**	11.79	**77.62**	88.21	84.02	91.45	**80.69**	**89.8**
Profile NR(Perceptron)‡	-	-	26.5	-	-	-	-	-	-	-

All measures are presented as percentages.

†: quoted from [5]; ‡: quoted from [21]

kernel becomes substantially more expensive to compute for large datasets as indicated in [14, 26] by Weston *et al.* .

Table 7: Complexity of computations.

Method	Time complexity	Running time (s)
Supervised setting		
Triple kernel	$O(d^2nN + d^2\|\Sigma\|^3N^2)$	112
Double kernel	$O(dnN + d\|\Sigma\|^2N^2)$	54
Mismatch	$O(k^{m+1}\|\Sigma\|^m nN + \|\Sigma\|^k N^2)$	948
Gapped kernel	$O(\binom{g}{k}knN + \|\Sigma\|^k N^2)$	176
Semi-supervised setting		
Triple kernel	$O(d^2HnN + d^2\|\Sigma\|^3N^2)$	327
Double kernel	$O(dHnN + d\|\Sigma\|^2N^2)$	67
Mismatch	$O(k^{m+1}\|\Sigma\|^m HnN + \|\Sigma^k\|N^2)$	-
Profile kernel	$O(kM_\sigma nN + \|\Sigma\|^k N^2)$	10 hours†

† the running time is quoted from [14]

Notations used in the table: N-number of sequences, n-sequence length,
H is the sequence neighborhood size,
$|\Sigma|$ is the alphabet size
k, m are mismatch kernel parameters ($k = 5, 6$ and $m = 1, 2$ in most cases)
M_σ is the profile neighborhood size, $M_\sigma \leq |\Sigma^k|$

In previous studies [14, 26], to achieve good accuracy the number of the PSI-BLAST iterations needs to be at least 5, while our performance is achieved with only 2 iterations. We also note that the results reported in [23] are not directly comparable since an older SCOP 1.53 benchmark is used and the results are optimized on testing sequences; also, the obtained similarity measures in the corresponding study do not satisfy positive semi-definiteness condition (are not Mercer kernels).

5.2 Biological motivation

The feature sets induced by our kernels cover segments of variable length (e.g., $2-6$ residues in the case of the double-$(1,5)$ kernel). On the other hand, the mismatch and profile kernels cover segments of fixed length (e.g., 5 or 6 residues long) as illustrated in Figure 1. Sampling at different resolutions also allows one to capture similarity in the presence of more complex substitution, insertion, and deletion processes, whereas sampling at a fixed resolution, the approach used in mismatch and spectrum kernels, limits the sensitivity in the case of multiple insertions/deletions or substitutions. Increasing the parameter m (number of mismatches allowed) to accommodate the multiple substitutions, in the case of mismatch/spectrum kernels, leads to an exponential growth in the neighborhood size, and results in high computational complexity.

The proposed features also capture short-term dependencies and interactions between local sequence features by explicitly encoding the spatial information. In contrast, such information is not present in the gapped/subsequence kernels [15, 20]. In a weighted version of the subsequence kernel, where each instance (subsequence) of a particular k-mer is weighted inversely proportional to the length of the subsequence, the count for a particular k-mer is the sum of such weights. When sequences are matched under the weighted subsequence kernel, the final counts (the sum of weights) are compared and no distinction is made as to how the features were positioned in the sequences, i.e. the information on the spatial configuration of the features within the sequence is not retained.

We further illustrate differences between the proposed kernels and gapped/subsequence kernels for the case when the basic features (individual samples) of the spatial sample kernels are single characters in Equations 7 (spatial kernels) and 8 (gapped/subsequence kernels) below:

$$K(X,Y) = \sum_{\substack{(a_1,\ldots,a_t)\\ a_i\in\Sigma}} \sum_{\substack{(d_1,\ldots,d_{t-1})\\ 0\leq d_i<d}} \begin{array}{c} c((a_1,d_1,\ldots,d_{t-1},a_t)|X)\cdot \\ c((a_1,d_1,\ldots,d_{t-1},a_t)|Y) \end{array} \tag{7}$$

$$K_g(X,Y) = \sum_{(a_1,a_2,\ldots,a_t)} \begin{array}{c} \left(\sum_{d_1,d_2,\ldots,d_{t-1}} c((a_1,d_1,\ldots,d_{t-1},a_t)|X)\right)\cdot \\ \left(\sum_{d_1,d_2,\ldots,d_{t-1}} c((a_1,d_1,\ldots,d_{t-1},a_t)|Y)\right) \end{array} \tag{8}$$

where $c(\cdot|X)$ is the (weighted) count and $\sum_{i=1}^{t-1} d_i = g-t$ for the gapped (g,t) kernels. Note that the spatial configuration information is integrated out in the gapped/subsequence kernels, but still maintained in SSSK.

5.3 Discovering short sequence motifs with spatial information

Previous biological studies (e.g. [12]) suggested that the spatial information such as distances between some conserved key positions can play a key role in capturing inherent characteristics of superfamilies. Our method indirectly identifies meaningful features in the protein data using the *Scorpion toxin-like* superfamily as an example. Several families in this superfamily are characterized by a number of disulphide bridges formed by conserved cysteine (C) residues. The relative positions (distances) of some neighboring key residues are also conserved as shown in Figure 4 (obtained from PROSITE [9]) for the *short-chain scorpion toxin* family. In the experiment, this family is held out for testing and all other families under the superfamily are used for training (16 positive training sequences). Among the positive sequences, 16 are selected as positive support vectors and 88 out of 1067 negative sequences are selected as negative support vectors. Under the double(1,5) representation, the pattern 'C___C' (3 residues between the two conserved cysteines residues) has the highest weight, consistent with the schematic representation shown in Figure 4. This feature is present in all positive support vectors, with the average count of 1.81 and in the negative support vectors with the average count of 0.43. The corresponding feature 'CC' under the gapped(2,4) representation has been suppressed (ranked 38 out of 400 features) due to over-representation of such feature in the negative support vectors: 39 out of 43 negative support vectors contain the feature, compared to 25 out of 88 negative support vectors with the similar feature using the double kernel. Integrating out the spatial information suppresses such feature due to its presence in the negative sequences (the average counts in the positive and negative support vectors are very close: 8.33 and 7.61). Lack of spatial information also leads to lower performance for the gapped kernel: the ROC50 score for the gapped kernel is 28.35, compared to 76.61 for the double kernel.

```
       +---------------------+
       |                     |
       |                     |
xxxxxxxCxxxxxCxxxCxxxxxxxxxxxCxxxxCxCxxx
             |   |                |  |
             |   +----------------+  |
             +-----------------------+
0         1         2         3
1         1         1         1
```

Figure 4: The schematic representation of the *short-chain scorpion toxins* family (obtained from PROSITE)

6. CONCLUSION

We present a computationally efficient approach for protein sequence analysis that scales well with very large sequence databases and shows state-of-the-art performance on two difficult tasks in protein sequence classification: remote homology detection and remote fold recognition. The key component of the method is the spatially-constrained sample kernel for efficient sequence comparison, which, when combined with kernel smoothing using unlabeled sequence databases, leads to rapid and accurate semi-supervised remote homology detection and fold recognition. The proposed methodology can be readily applied to other challenging problems in biological sequence analysis such as motif elucidation, ranking and clustering.

7. REFERENCES

[1] S. Altschul, W. Gish, W. Miller, E. Myers, and D. Lipman. Basic Local Alignment Search Tool. *Journal of Molecular Biology*, pages 403–410, 1990.

[2] A. Bairoch, R. Apweiler, 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. The Universal Protein Resource (UniProt). *Nucl. Acids Res.*, 33(suppl-1):D154–159, 2005.

[3] D. A. Benson, I. Karsch-Mizrachi, D. J. Lipman, J. Ostell, and D. L. Wheeler. Genbank. *Nucl. Acids Res.*, 33(suppl-1):D34–38, 2005.

[4] J. Cheng and P. Baldi. A machine learning information retrieval approach to protein fold recognition. *Bioinformatics*, 22(12):1456–1463, June 2006.

[5] C. H. Ding and I. Dubchak. Multi-class protein fold recognition using support vector machines an d neural networks . *Bioinformatics*, 17(4):349–358, 2001.

[6] S. Eddy. Profile hidden Markov models. *Bioinformatics*, 14(9):755–763, 1998.

[7] M. Gribskov, A. McLachlan, and D. Eisenberg. Profile analysis: detection of distantly related proteins. *PNAS*, 84:4355–4358, 1987.

[8] M. Gribskov and N. L. Robinson. Use of receiver operating characteristic (roc) analysis to evaluate sequence matching. *Computers & Chemistry*, 20(1):25–33, 1996.

[9] N. Hulo, A. Bairoch, V. Bulliard, L. Cerutti, E. De Castro, P. S. Langendijk-Genevaux, M. Pagni, and C. J. A. Sigrist. The PROSITE database. *Nucl. Acids Res.*, 34:D227–230, 2006.

[10] T. Jaakkola, M. Diekhans, and D. Haussler. Using the Fisher kernel method to detect remote protein homologies. In *Proceedings of the Seventh International Conference on Intelligent Systems for Molecular Biology*, pages 149–158. AAAI Press, 1999.

[11] T. Jaakkola, M. Diekhans, and D. Haussler. A discriminative framework for detecting remote protein homologies. In *Journal of Computational Biology*, volume 7, pages 95–114, 2000.

[12] A. E. Kister, A. V. Finkelstein, and I. M. Gelfand. Common features in structures and sequences of sandwich-like proteins. *PNAS*, 99(22):14137–14141, 2002.

[13] R. Kuang, E. Ie, K. Wang, K. Wang, M. Siddiqi, Y. Freund, and C. Leslie. Profile-based string kernels for remote homology detection and motif extraction. In *CSB '04: Proceedings of the 2004 IEEE Computational Systems Bioinformatics Conference (CSB'04)*, pages 152–160, August 2004.

[14] R. Kuang, E. Ie, K. Wang, K. Wang, M. Siddiqi, Y. Freund, and C. Leslie. Profile-based string kernels for remote homology detection and motif extraction. *J Bioinform Comput Biol*, 3(3):527–550, June 2005.

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

[16] C. S. Leslie, E. Eskin, and W. S. Noble. The spectrum kernel: A string kernel for svm protein classification. In *Pacific Symposium on Biocomputing*, pages 566–575, 2002.

[17] C. S. Leslie, E. Eskin, J. Weston, and W. S. Noble. Mismatch string kernels for svm protein classification. In *NIPS*, pages 1417–1424, 2002.

[18] L. Liao and W. S. Noble. Combining pairwise sequence similarity and support vector machines for remote protein homology detection. In *RECOMB*, pages 225–232, 2002.

[19] L. Lo Conte, B. Ailey, T. Hubbard, S. Brenner, A. Murzin, and C. Chothia. SCOP: a structural classification of proteins database. *Nucleic Acids Res.*, 28:257–259, 2000.

[20] H. Lodhi, C. Saunders, J. Shawe-Taylor, N. Cristianini, and C. Watkins. Text classification using string kernels. *J. Mach. Learn. Res.*, 2:419–444, 2002.

[21] I. Melvin, E. Ie, J. Weston, W. S. Noble, and C. Leslie. Multi-class protein classification using adaptive codes. *J. Mach. Learn. Res.*, 8:1557–1581, 2007.

[22] W. Pearson and D. Lipman. Improved tools for biological sequence comparison. *PNAS*, 85:2444–2448, 1988.

[23] H. Rangwala and G. Karypis. Profile-based direct kernels for remote homology detection and fold recognition. *Bioinformatics*, 21(23):4239–4247, 2005.

[24] S. Sonnenburg, G. Rätsch, and B. Schölkopf. Large scale genomic sequence svm classifiers. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 848–855, New York, NY, USA, 2005.

[25] V. N. Vapnik. *Statistical Learning Theory.* Wiley-Interscience, 1998.

[26] J. Weston, C. Leslie, E. Ie, D. Zhou, A. Elisseeff, and W. S. Noble. Semi-supervised protein classification using cluster kernels. *Bioinformatics*, 21(15):3241–3247, 2005.

Catching Old Influenza Virus with A New Markov Model

HamChing Lam
Dept. of Computer Science and Engineering
University of Minnesota
Minneapolis, Minnesota 55455, USA
hamching@cs.umn.edu

Daniel Boley
Dept. of Computer Science and Engineering
University of Minnesota
Minneapolis, Minnesota 55455, USA
boley@cs.umn.edu

ABSTRACT

We have developed a novel Markov model which models the genetic distance between viruses based on the Hemagglutinin (HA) gene, a major surface antigen of the avian influenza virus. Using this model we estimate the probability of finding highly similar virus sequences separated by long time gaps. Our biological assumption is based on neutral evolutionary theory, which has been applied previously to study this virus [Gojobori, Moriyama, and Kimura. PNAS Vol 87. 1990]. Our working hypothesis is that after a long enough time gap and with the high mutation rate usually found in RNA viruses, many site mutations should accumulate, leading to distinct modern variants. We obtained 3439 HA protein sequences isolated through years 1918 to 2006 from around the globe, aligned them to a consensus sequence using the NCBI alignment tool, and used a Hamming distance metric on the aligned sequences. We tested our hypothesis by combining a standard Poisson process with a Markov model. The Poisson process models the occurrences of mutations in a given time interval, and the Markov model estimates the probabilities of changes to the genetic distances due to mutations. By coalescing all sequences at a given genetic distance to a single state, we obtain a tractable Markov chain with a number of states equal to the length of the base peptide sequence. The model predicts that the probability of finding highly similar virus after several decades is extremely small. The existence of recent viruses which are very similar to older viruses suggests that potentially there exists some reservoir which preserves viruses over long periods.

Keywords

Influenza virus, Poisson process, Markov Model

1. INTRODUCTION

For the past century researchers have been studying influenza viruses (IV). Belonging to the viral family *Orthomyxoviridae*, influenza viruses have eight unique RNA segments [20] that encode 10 different gene products (PB1 polymerase, PB2 polymerase, PA polymerases, Hemagglutinin (HA), Nucleoprotein (NP), Neuraminidase (NA), Matrix M1 and M2 proteins, and Nonstructural NS1 and NS2 proteins). The target of our study is the Hemagglutinin HA gene product. We have developed a novel Markov model which models the genetic distance between viruses based on the Hemagglutinin (HA) gene, a major surface antigen of the avian influenza virus. Our working hypothesis is that after a long enough time gap, many site mutations should accumulate in the virus due to a lack of a proofreading function [6], leading to distinct modern variants. We based our biological assumption on neutral theory of evolution [7, 12, 17, 8] and that each amino acid site is under a neutral mutation pressure. Previous studies have shown that subtypes of influenza virus are subjected to higher silent substitution rate [7, 24], which is consistent with the neutral theory of molecular evolution. Although their studies were conducted using nucleotide sequences, we believe that the same general concept and framework can be applied to study protein sequences of this virus under this evolutionary assumption. We test our hypothesis by combining a standard Poisson process with the Markov model. The Poisson process models the occurrences of mutations in a given time interval, and the Markov model estimates the probabilities of changes to the genetic distances due to mutations. We show that it is highly unlikely that very similar sequences would arise long after the original sequence. Given the observations of several pairs of very similar sequences separated by several decades, we conclude that there must be some reservoir or evolutionary mechanism that is capable of preserving old virus strains, allowing them to reappear after extended time intervals.

2. MATERIALS AND METHODS

2.1 Protein Sequence Data and Processing

The HA protein is the major surface antigen of the influenza virus. Its role is to bind to host cell receptors promoting fusion between the viron envelope and the host cell [20]. Influenza A virus HA genes have been classified into 16 subtypes (H1-H16) according to their antigenic properties. This HA protein is cleaved into two peptide chains HA1 and HA2 respectively when matured [19]. The HA2 chain has been found to vary less and is more conserved compared to HA1 chain [10]. The HA1 chain is 329 residues long and is the immunogenic part of HA protein. Past studies have shown that HA1 is undergoing continual diversify-

BIOKDD '08 Las Vegas, Nevada USA
Copyright 2008 ACM ISBN 978-1-60558-302-0 ...$5.00.

ing change [5, 14] and is the most variable portion of the influenza genome[16].

Using the NCBI Influenza database available online [23], we have collected 3439 influenza virus type A protein sequences deposited before December, 2007 (excluding identical sequences and lab strains/NIAID FLU project). This collection of protein sequences contains isolates from around the globe and from a diverse range of hosts. We used protein sequences because they were known to give more reliable results than nucleotide sequences when constructing evolutionary history [19]. Each of the 3439 sequences has a unique annotation which contains the host organism, the strain number, the year of isolation, subtype, and protein name. We aligned all sequences to a consensus sequence using the NCBI alignment tool. According to the study presented by [16], a uniform consensus strain tends to circulate for some time, since the mutations that occur during replication do not become fixed in the early stages of circulating. The aligned sequence data were then used with a genetic distance function to determine the pairwise genetic distance (including gaps) of the sequences.

The genetic distance between two sequences can be thought of as the "edit" distance, which is the number of single letter changes needed to transform one sequence to the other. This yields a simple scoring function assigning a zero to a matching amino acid base and a one to a mismatch. The sum of all mismatches is usually called the Hamming distance (k) or Hamming score for the pairwise sequence comparison. For comparison of very similar biological sequences, this Hamming distance can be used under the assumption that the observed difference between a pair of sites represents one mutation [3]. The present study could also be carried out using BLAST or any alignment algorithm, but as considerably greater expense. In [15], Hamming distance was successfully used to find interesting clusters of IV HA sequences and to predict vaccine strains with good results. Hamming distance as genetic distance between viruses has also been used effectively in modeling influenza viruses [18]. In our study, we compute the Hamming distance based on a consensus alignment to account for the small number of insertions and deletions. We then store the pairwise Hamming distance scores of HA gene in a pairwise affinity matrix and identify virus sequence pairs sharing high sequence similarity (at least 90 percent) but separated by a long time gap.

2.2 Markov model

We model all mutations as the combination of several single point mutations and use a Poisson process to model the mutation rate. The Poisson process naturally admits more complex mutations, treating them as several single point mutations occurring in rapid succession. Then we build a compact Markov model to model the mutations themselves. Markov models have proven to be a powerful tool for phylogenetic inference and hypothesis testing when modeling transitions between amino acid states. Modeling amino acid transitions is complex since proteins are made of twenty amino acids. Because of this, we take a very different approach in building our Markov model. We are trying to avoid a Markov chain where each sequence is a state because this would give rise to an exponentially large number of states (20^n where n is the number of sites). In our Markov model, we collect into a single state H_k all the protein sequences at given Hamming distance k from the starting sequence $s_0 \in H_0$. The starting sequence s_0 can be chosen either as the earliest isolated sequence or the most recent one as long as a large time gap is observed when comparing to other sequences. Our Markov model assigns the probability of an arbitrary HA sequence $s_1 \in H_k$ mutating into a different HA sequence $s_2 \in H_l$ through a single point mutation, where l must be one of $k-1, k, k+1$.

$$\begin{pmatrix} 0 & 1 & & & \\ x_1 & y_1 & z_1 & & 0 \\ & \ddots & \ddots & \ddots & \\ 0 & & x_{n-1} & y_{n-1} & z_{n-1} \\ & & & x_n & y_n \end{pmatrix}$$

Figure 1: Markov transition matrix

Previous studies [4, 13] have shown that to better fit the model, active sites should be excluded in the analysis under the neutral theory framework. Here we have taken the same approach where we have limited the mutations captured by our Markov chain to the HA1 domain consisting of $n = 329$ sites, since this region is less conserved than the HA2 region [15, 14]. Therefore, our Markov model has only $n + 1 = 330$ states instead of the 20^n states it would have if we kept each state and each possible transition separate.

Formally, consider a finite set of states labeled $\{H_0, ..., H_n\}$. In order to keep the Markov chain to a manageable size, we group all the sequences within Hamming distance of k from a start sequence into a single "super state" H_k. At each transition, we assume a single point mutation occurs, and that this mutation of amino acid replacement exhibits uniform rate of evolution throughout long periods of evolutionary time [25]. This assumption is particularly consistent with the concept of "molecular evolutionary clock" and is central to the neutral theory [1, 7, 22]. Because of the high rate at which RNA viruses evolve, it has been observed that these sequences show the typical pattern of neutral evolution [7].

We denote by a the size of the alphabet of amino acids, in our case 20. For a sequence $s_1 \in H_k$, there is a probability k/n that the mutation occurs in one of the k positions where s_1 differs from s_0, and if this change occurs, there is a $1/(a-1)$ chance that the new amino acid in this position will match that in the same position of s_0. Hence the probability x_k of a transition from H_k to H_{k-1} is $x_k = \frac{k}{n} \cdot \frac{1}{a-1}$. Similar reasoning yields the probability y_k that a transition will remain at the same Hamming distance: $y_k = \frac{k}{n} \cdot \frac{a-2}{a-1}$. The probability that mutation will be in one of the $n - k$ sites that still match s_0 is $z_k = 1 - \frac{k}{n}$, corresponding to a transition from H_k to H_{k+1}. The resulting probabilities x_k, y_k, z_k are assembled into a Markov transition matrix M shown in Figure 1. The entries in each row of M add up to 1.

Using this model, we can compute the probability q_t that a virus will have a Hamming distance at most k from the initial source sequence after t mutations. We give the general form of how to compute the above probability. We let $v_t = (v_{t0}, v_{t1}, \ldots, v_{tn})$ be the row vector of probabilities of being in state $H_0, H_1, \ldots, H_n$, respectively, after t mutations. At $t = 0$ we are in state H_0 consisting of just the initial sequence. This is represented by the row vector

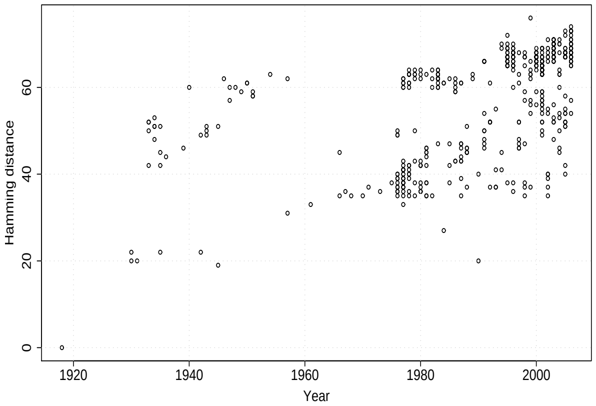


Figure 2: H1 subtype pairwise Hamming distance plot

Table 1: H1N1 subtype long time gap strains (Rate: 2×10^{-3} per site per year). H = Hamming distance, Y = Year, EG = Expected number of mutations.

Strain	H	Y	EG	$\mathcal{P}$-value
AAD17229: A/South Carolina/1/1918	0	0	0	source sequence
AAA91616: A/swine/St-Hyacinthe/148/1990(H1N1)	20	72	47.3	6.3499e-06

$v_0 = (1, 0, 0, ..., 0)$. Then the vector of probabilities after $t + 1$ mutations is related to the probabilities after t mutations by $v_{t+1} = v_t * M$. The probability of being at most distance k from s_0 after t mutations is the sum of the first $k + 1$ components of v_t: $q_t(k) = \sum_{i=0}^{k} v_{ti}$.

The above analysis counts events consisting of a single mutation. The mutation rate is modeled by a Poisson process [4, 11]. This includes the possibility that no mutation or several mutations take place in a given time interval, assuming all sites undergo the same substitution rate. This assumes that the probability of a mutation in a given time interval depends only on the length of the interval but is independent of the behavior outside the time interval. If λ is the average number of mutations in a time interval of 1 year, then the probability that t mutations occur in any time interval of length Y is given by $p_t(Y) = \frac{(Y\lambda)^t}{t!}e^{-Y\lambda}$. The Poisson process models when mutations occur, and the Markov model models the nature of the mutations. Combining these two models yields the probability $P_\kappa(Y)$ that after Y years a sequence would appear with a genetic distance from s_0 of κ, namely $P_\kappa(Y) = \sum_{t=0}^{\infty} p_t(Y) \cdot q_t(\kappa)$.

3. RESULTS AND DISCUSSION

We first identified viruses having very close genetic distance but with large time gap. Figure 2 shows the H1 subtype HA1 domain pairwise sequence genetic distance plotted against time of isolation in year. The genetic distance corresponds to the Hamming distance including gaps. Tables 1 and 2 show viruses sharing very high sequence similarity but with large time gap. We used the amino acid substitution rate of $r = 2\times10^{-3}$ per site per year for H1 and H2 subtype viruses, estimated using the entire region of the HA gene and assuming that the molecular clock is followed [19] throughout evolutionary history. This yields an annual mutation rate of $\lambda = nr = 329\cdot2\times10^{-3} = 0.658$. We give two examples of unlikely similarities over long time gaps in table 1 and 2. Each table includes the accession number "Accession", strain name "Strain", the Hamming distance "H" (calculated from the first strain), expected number of mutations "EG", the year difference "Y", and the $\mathcal{P}$-value, the probability that this Hamming distance (or less) would be observed after the given time interval as predicted by our model. Using the pandemic strain A/South Carolina/1/1918 and A/swine/St-Hyacinthe/148/1990(H1N1) from Table 1, the interpretation of the result is that after 72 years, the expected number of mutations is 47.3 and the probability of being within a Hamming distance of 20 of the original source sequence is 6.35×10^{-6}. A very recent published research study [22] employing the state-of-the-art Bayesian Markov chain Monte Carlo [2] which allows for substitution rate variation and maximum likelihood phylogenetic methods indicates that this A/swine/St-Hyacinthe/148/1990(H1N1) virus is a contaminant from the A/swine/1930 strain. The genetic distance of the pandemic strain to the A/swine/1930 strain is 22. The genetic distance of A/swine/1930 to A/swine/St-Hyacinthe/148/1990(H1N1) is only 3 indicating that these two strains are virtually identical. From table 2, we see that A/Human/Canada/720/2005(H2N2) strain isolated in 2005 is exceptionally similar to the two asian pandemic strains A/Singapore/1/1957(H2N2) and A/Human/Japan/305/1957(H2N2) in terms of the genetic distance. These two pandemic strains were human transmissible and currently no influenza vaccines contained the H2N2 virus [21]. This reappearance of the highly pathogenic H2N2 virus could cause a potential pandemic as current population is not immunized against this strain of virus. The origin of the A/Human/Canada/720/ 2005(H2N2) strain was traced back to human error at a laboratory distributing virus samples for training purposes and the distributed strains were quickly destroyed at all receiving laboratories [21].

Table 2: H2 subtype long time gap strains

Strain	H	Y	EG	$\mathcal{P}$-value
AAY28987: A/Human/Canada/720/2005(H2N2)	0	0	0	source sequence
AAA64365: A/RI/5+/1957(H2N2)	6	48	31.5	7.807e-09
AAA64363: A/RI/5-/1957(H2N2)	3	48	31.5	1.206e-11
AAA64366: A/Singapore/1/1957(H2N2)	5	48	31.5	1.155e-09
AAA43185:A/Human/Japan/305/1957(H2N2)	5	48	31.5	1.155e-09

To check how our model matches the data, we show the predicted distribution of Hamming distances in Figure 3 based on a time interval of $Y = 49$ and annual mutation rate of $nr = 0.658$ for the H2 subtype. The peak of the curve indicates that with high probability, roughly 30-40 mutation events would have taken place. This tells us that we should expect to see the majority of H2 sequence pairs with Hamming distances in the vicinity of 40 given the length of time interval equals 49 years base on Poisson process assumption. We compare this to the actual distribution of Hamming distances found in the H2 subtype data shown in Figure 4 over the range of data available (from 1957 through 2006 or a span of 49 years). Figure 4 shows that the majority

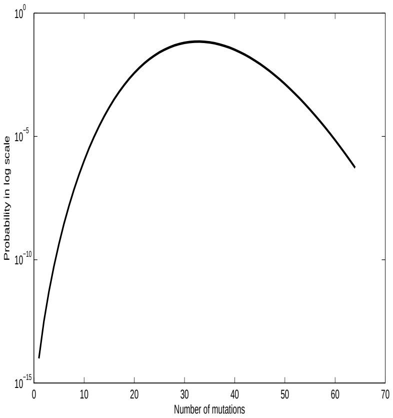


Figure 3: Poisson process distribution plot

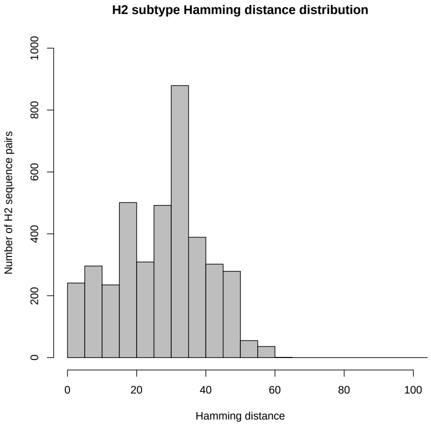


Figure 4: H2 subtype histogram plot

of the H2 sequence pairs have Hamming distances around 35, which matches the Poisson process prediction. Figure 6 illustrates how the probability values of 3 H2 strains in Table 2 are rapidly dropping against the expected number of mutations from the Markov model calculation. Figure 5 shows the predicted distribution within the time interval of 70-85 years from the combined Poisson process and Markov chain model using H1 subtype HA1 sequences. The curve shows that with high probability most sequences should be in states H_{60} to H_{70}. This reflects what is observed in figure 2 and figure 7 where most sequences have Hamming distance around 60-70. This suggests that our model is able to capture the overall evolutionary behavior of the influenza virus according to a molecular clock, leading to a natural increase in the genetic distance as time passes, consistent with [1].

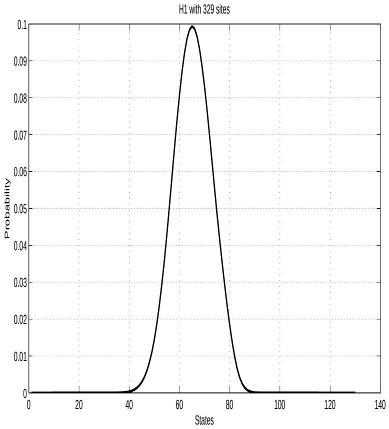


Figure 5: Model prediction plot

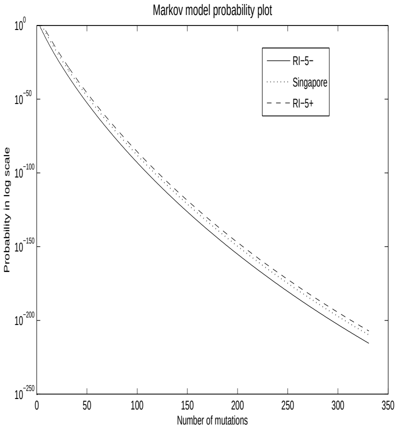


Figure 6: H2 strains probability plot

4. CONCLUSIONS

The extensive genetic diversity of influenza A viruses through genetic drift and reassortment in the past century has resulted in many new strains being produced. However, H1, H2, and H3 subtypes strains have displayed cyclic behavior resulting in influenza pandemics [6]. In the present study, we applied neutral evolution theory to influenza virus HA protein sequences to investigate the evolutionary dynamics of the virus. Using the combination of a Poisson model with a novel Markov model, we were able to calculate the probability values of finding a very similar sequence composition separated by a large time gap. We have so far been able to identify several anomalies due to laboratory artifacts or human error. This finding is promising since we have yet to apply it in a full scale comprehensive analysis of all 16 subtypes of the virus. However, judging by the extremely low probability values obtained for some observed sample strains, we conclude that there may be one or more sources of various strains of the virus in which they are preserved over long time periods. The existence of reservoirs preserving viruses for decades cannot be completely eliminated.

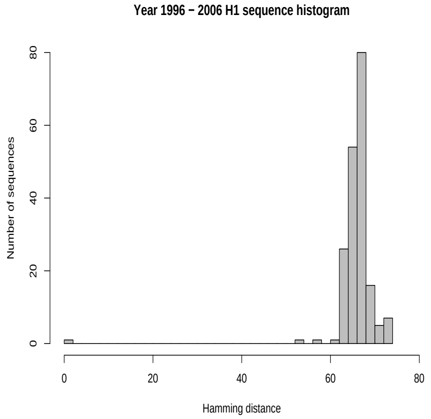


Figure 7: Histogram of H1 from 1996-2006

5. FUTURE WORK

For future work, our immediate next steps are: (1) apply our model to nucleotide sequences which allows us to compare our model with other existing models that study nucleotide sequences of the virus, and (2) use a more robust distance function in which we can incorporate antigenic distance information to the model. The evolutionary modeling of influenza virus has primarily been based on models using nucleotides substitution models and phylogenetic analysis. Our approach is different in that we demonstrated that by applying the same theoretical concept, we can instead model the differences between viral protein sequences. A key advantage of modeling the differences between sequences is that the distance function can be further refined so that additional genetic information can be incorporated into the model. However, it is imperative that we compare our model to existing models where nucleotide sequences are used and to provide a rigorous statistical framework in support of our new Markov model.

Incorporating antigenic distance information is vital due to the fact that vaccine strain selection is largely based on the antigenic differences between circulating strains and influenza viruses are antigenically variable in each influenza season. The antigenic distance map, originally proposed by Lapedes and Farber [9], is a geometric interpretation of Hemagglutination Inhibition (HI) binding assay data wheres a point is assigned in a two dimensional grid between each antigen and antiserum and this distance reflects the direct HI measurement. The antigenic distance measurement can be included in the genetic distance function to find a total distance value. Further, HI binding assay data is generated through the binding of individual viral protein to red blood cells[6], this implies that a pairwise alignment scheme for sequence comparison can be used to capture each sequence's compositional characteristic.

6. ACKNOWLEDGEMENT

This work was partially supported by NSF grant 0534286.

7. REFERENCES

[1] R. Chen and E. C. Holmes. Avian influenza virus exhibits rapid evolutionary dynamics. *Mol. Biol. Evol.*, 23:2336–2341, 2006.

[2] A. Drummond, G. Nicholls, A. Rodrigo, and W. Solomon. Estimating mutation parameters, population history and genealogy simultaneously from temporally spaced sequence data. *Genetics*, 161:1307–1320, 2002.

[3] I. Eidhammer, I. Jonassen, and W. Taylor. *Protein Bioinformatics: An algorithmic approach to sequence and structure analysis*. John Wiley and Sons, 2004.

[4] W. Fitch and E. Margoliash. A method for estimating the number of invariant amino acid coding positions in a gene using cytochrome c as model case. *Biochemical Genetics*, 1(1):65–71, June 1967.

[5] W. M. Fitch, J. M. E. Leiter, X. Li, and R. Palese. Positive darwinian evolution in human influenza a viruses. *Proc. Natl. Acad. Sci. USA*, 88:4270–4274, 1991.

[6] S. J. Flint, L. Enquist, V. Racaniello, and A. Skalka. *Principles of Virology.* ASM press, 2004.

[7] T. Gojobori, E. Moriyama, and M. Kimura. Molecular clock of viral evolution, and the neutral theory. *Proc Natl Acad. Sci. USA*, 87:10015–10018, 1990.

[8] F. P. Kelly. *Reversibility and Stochastic Networks.* John Wiley and Sons, 1979.

[9] A. Lapedes and R. Farber. The geometry of shape space: Application to influenza. *Journal of Theor. Biol.*, 212(1):57–69, September 2001.

[10] W. Laver, G. Air, R. Webster, W. Gerhard, C. Ward, and T. Dopheide. The antigenic sites on influenza virus hemagglutinin. studies on their structure and variation in influenza virus. *Dev. Cell Biol*, 5:295–307, 1980.

[11] M. Nei and S. Kumar. *Molecular evolution and phylogenetics*. Oxford University Press, Oxford, New York, 2000.

[12] T. Ohta and M. Kimura. On the constancy of the evolutionary rate of cistrons. *J Mol Evol*, 1:18–25, 1971.

[13] M. Plass and E. Eyras. Differentiated evolutionary rates in alternative exons and the implications for splicing regulation. *BMC Evol. Biol*, 6(50), June 2006.

[14] J. B. Plotkin and J. Dushoff. Codon bias and frequency-dependent selection on the hemagglutinin epitopes of influenza a virus. *Proc Natl Acad Sci USA*, 100(12):7152–7157, June 2003.

[15] J. B. Plotkin, J. Dushoff, and S. A. Levin. Hemagglutinin sequence clusters and the antigenic evolution of influenza a virus. *Proc Natl Acad Sci USA*, 99(9):6263–6268, April 2002.

[16] A. H. Reid, T. A. Janczewski, R. Lourens, A. J. Elliot, R. Daniels, C. L. Berry, J. S. Oxford, and J. K. Taubenberger. 1918 influenza pandemic caused by highly conserved viruses with two receptor-binding variants. *Emerging Infectious Diseases*, 9(10), 2003.

[17] S. A. Sawyer. On the past history of an allele now known to have frequency p. *J Appl Probab*, 14:439–450, 1977.

[18] D. J. Smith, F. Forrest, D. H. Ackley, and A. S. Perelson. Variable efficacy of repeated annual influenza vaccination. *Proc Natl Acad Sci USA*,

96(24):14001–14006, November 1999.

[19] Y. Suzuki and M. Nei. Origin and evolution of influenza virus hemagglutinin genes. *Mol. Biol. Evol.*, 19(4):501–509, 2002.

[20] R. Webster, W. Bean, O. Gorman, T. Chambers, and Y. Kawaoka. Evolution and ecology of influenza a viruses. *Microbiological Reviews*, pages 152–179, March 1992.

[21] WHO. Epidemic and pandemic alert and response (epr):international response to the distribution of a h2n2 influenza virus for laboratory testing: Risk considered low for laboratory workers and the public. April 2005.

[22] M. Worobey. Phylogenetic evidence against evolutionary stasis and natural abiotic reservoirs of influenza a virus. *J of Virology*, 82(7):3769–3774, April 2008.

[23] P. B. Y. Bao, D. Dernovoy, B. Kiryutin, L. Zaslavsky, T. Tatusova, J. Ostell, and D. Lipman. The influenza virus resource at the national center for biotechnology information. *Journal of Virology*, 82(2):596–601, January 2008.

[24] S. Yoshiyuki. Natural selection on the influenza virus genome. *Mol. Biol. Evol*, 23, June 2006.

[25] E. Zuckerkandl and L. Pauling. *Molecular disease, evolution, and genetic heterogeneity*. Academic Press, New York, 1962.

GPD: A Graph Pattern Diffusion Kernel for Accurate Graph Classification with Applications in Cheminformatics

Aaron Smalter, Jun Huan, Jia Yi
Department of Electrical Engineering and Computer Science
University of Kansas
Lawrence, KS, 66047-7621
{asmalter, jhuan, yjia}@eecs.ku.edu

Gerald H. Lushington
Molecular Graphics and Modeling Laboratory
University of Kansas
Lawrence, KS, USA
glushington@ku.edu

ABSTRACT

Graph data mining is an active research area. Graphs are general modeling tools to organize information from heterogenous sources and have been applied in many scientific, engineering, and business fields. With the fast accumulation of graph data, building highly accurate predictive models for graph data emerges as a new challenge that has not been fully explored in the data mining community.

In this paper, we demonstrate a novel technique called graph pattern diffusion kernel (GPD). Our idea is to leverage existing frequent pattern discovery methods and to explore the application of kernel classifier (e.g. support vector machine) in building highly accurate graph classification. In our method, we first identify all frequent patterns from a graph database. We then map subgraphs to graphs in the graph database and use a process we call “pattern diffusion” to label nodes in the graphs. Finally we designed a novel graph alignment algorithm to compute the inner product of two graphs. We have tested our algorithm using a number of chemical structure data. The experimental results demonstrate that our method is significantly better than competing methods such as those kernel functions based on paths, cycles, and subgraphs.

Keywords

Graph Classification, Graph alignment, Frequent Subgraph Mining

1. INTRODUCTION

Graphs are ubiquitous models that have been applied in many scientific, engineering, and business fields. For example, in finance data analysis, graphs are used to model dynamic stock price changes [17]. To analyze biological data, graphs have been utilized in modeling chemical structures [27], protein sequences [34], protein structures [13], and gene regulation networks [14]. In web page classification, graphs are used to model the referencing relationship in HTML documents [40].

Due to the wide range of applications, development of computational and statistical frameworks for analyzing graph data has attracted significant research attention in the data mining community. In the past a few years, various graph pattern mining algorithms have been designed [10, 11, 28, 30, 36, 39]. There are also many research efforts dedicated to efficiently searching graph databases [20, 26, 35, 37]. Most of the existing work concentrates on analyzing graph data in an unsupervised way, and making predictions about graphs is usually not the goal. The research focus is well justified, since in order to make predictions of graph data we must have a large number of labeled training samples. Activities such as sample collection and sample labeling are time consuming and expensive.

With the rapid development of powerful and sophisticated data collection methods, there is a fast accumulation of labeled graph data. For example, many XML documents are modeled as trees or graphs and it is important to build classifiers for XML data [38]. As another example, natural language processing of sentences usually produces a tree (parsing tree) representation of a sentence. In many social science studies, building automated systems to classify sentences into several groups [21] is an important task.

What is especially interesting to us is the chemical classification problem in cheminformatics. Chemical structures have been studied using graph modeling for a long time [29]. With recently developed high throughput screening methods, the National Institute of Health has started an ambitious project called the Molecular Library Initiative aiming to determine and publicize the biological activity of at least a million chemical compounds each year in the next 5 to 10 years[2].

With the fast accumulation of graph data including class labels, *graph classification*, which we focus on in this paper, is an emergent research topic in the data mining community. Though classification has been studied for many years in data mining, graph classification is undeveloped and brings many new challenges. Below, we highlight a few of the new challenges.

In many existing classification algorithms [4], samples and their target values are organized into an object-feature matrix $X = (x_{i,j})$ where each row in the matrix represents a sample and each column represents a measurement (or a *feature*) of the sample. Graphs are among a group of objects called semi-structured data that cannot easily conform to a

SIGKDD ’08 Las Vegas, Nevada USA

matrix representation. Other examples in the group include sequences, cycles, and trees. Though many different features have been proposed for graph data (e.g. paths, cycles, and subgraphs), there is no universally accepted way to define features for graph data.

Besides choosing the right feature representation, computational efficiency is also a serious concern in analyzing graph data. Many graph related operations, such as subgraph matching, clique identification, and hamiltonian cycle discovery are NP-hard problems. For those that are not NP-hard problems, e.g. all-by-all shortest distance, the computational cost could be prohibitive for large graphs.

In this paper, we aim to leverage existing frequent pattern mining algorithms and explore the application of kernel classifiers in building highly accurate graph classification algorithms. Towards that end, we demonstrate a novel technique called graph pattern diffusion kernel (GPD). In our method, we first identify all frequent patterns from a graph database. We then map subgraphs to graphs in the graph database and project nodes of graphs to a high dimensional space with a specially designed function. Finally we designed a novel graph alignment algorithm to compute the inner product of two graphs. We have tested our algorithm using a number of chemical structure data sets. The experimental results demonstrate that our method is significantly better than competing methods such as those based on paths, cycles, and other subgraphs.

In summary we present the following contributions in this paper:

- We have designed a novel way to measure graph similarity using graph kernel functions
- We prove that the exact computation of the kernel function is an NP-hard problem and we have designed an efficient algorithm to approximately compute the graph kernel function.
- We have implemented our kernel function and tested it with a series of cheminformatics data sets. Our experimental study demonstrates that our algorithm performs much better than existing state-of-the-art graph classification algorithms.

The rest of the paper is organized as follows. In section 1.1, we discuss the research efforts that are closely related to our current effort. In section 2, we define important concepts such as labeled graphs and graph kernel function, and clearly layout the graph classification problem. In section 3, we present the details of our way of measuring graph similarity with kernel functions. In section 4 we use real-world data sets to evaluate our proposed methods and perform a comparison of ours to the current state-of-the-art. Finally we conclude and present our future plan in section 5.

1.1 Related Work

We survey the work related to graph classification methods by dividing them into two categories. The first category of methods explicitly collect a set of *features* from the graphs. Possible choices are paths, cycles, trees, and general subgraphs [38]. Once a set of features is determined, a graph is described by a feature vector, and any existing classification methods such as CBA [4] and decision tree [24] that work in an n-dimensional Euclidian space, may be applied for graph classification.

The second approach is to implicitly collect a (possibly infinite) set of features from graphs. Rather than computing the features, this approach computes the similarity of graphs, using the framework of "kernel functions" [31]. The advantage of a kernel method is that it has low chance of over fitting, which is a serious concern in high dimensional space with low sample size.

In what follows we first give a brief review of pattern discovery algorithms from graphs. Those algorithms provide features for graph classification. We then review the first category algorithms, which explicitly utilize identified features. We delay the discussion of graph kernel functions to section 2 where we discuss kernel function in general and graph kernel functions specifically.

1.1.1 *Pattern Discovery*

Algorithms that search for frequent patterns (e.g. trees, paths, cyclic graphs) in graphs can be roughly divided into three groups.

The first group uses a level-wise search strategy, including AGM [15] and FSG [22]. The second category takes a depth-first search strategy, including gSpan[36] and FFSM [16]. Different from level-wise search algorithms AGM and FSG, the depth-first search strategy utilizes a back-track algorithm to mine frequent subgraphs. The advantage of a depth-first search is a better memory utilization since depth-first search keeps one frequent subgraph in memory and enumerates its supergraphs, in contrast to keeping all k-edge frequent subgraph in memory.

The third category of frequent subgraph mining algorithms does not work directly on a graph space to identify frequent patterns. Instead, algorithms in this category first project a graph space to another space such as that of trees, then identify frequent patterns in the projected space, and finally reconstruct all frequent patterns in the graph space. We call this strategy *progressive mining*. Algorithms in this category includes SPIN [12] and GASTON [23].

1.1.2 *Graph Classification*

Below we review two algorithms that use rule based methods for classifying graph data.

XRules [38] utilizes frequent tree-patterns to build a rule based classifier for XML data. Specifically, XRules first identifies a set of frequent tree-patterns. An association rule: $G \rightarrow c_i$ is then formed where G is a tree pattern and c_i is a class label. The *confidence* of the rule is the conditional probability $p(c_i|G)$ estimated from the training data. XRules carefully selects a subset of rules with high confidence values and uses those rules for classification.

Graph boosting [21] also utilizes substructures toward graph classification. Similar to XRules, graph boosting uses rules with the format of $G \rightarrow c_i$. Different from XRules, it uses the boosting technique to assign weights to different rules. The final classification result is computed as the weighted majority.

In the following discussion, we present the necessary background for a formal introduction to the graph classification problem, and introduce a suite of graph kernel functions for graph classification.

2. BACKGROUND

In this section we discuss a few important definitions for graph database mining: labeled graphs, subgraph isomor-

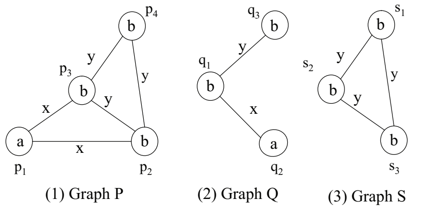


Figure 1: A Database of three labeled graphs.

phic relation, graph kernel function, and graph classification.

Definition 2.1. *A **labeled graph** G is a quadruple $G = (V, E, \Sigma, \lambda)$ where V is a set of vertices or nodes and $E \subseteq V \times V$ is a set of undirected edges. Σ is a set of (disjoint) vertex and edge labels, and λ: $V \cup E \to \Sigma$ is a function that assigns labels to vertices and edges. We assume that a total ordering is defined on the labels in Σ.*

A *graph database* is a set of labeled graphs.

Definition 2.2. *A graph $G' = (V', E', \Sigma', \lambda')$ is **subgraph isomorphic** to $G = (V, E, \Sigma, \lambda)$, denoted by $G' \subseteq G$, if there exists a 1-1 mapping $f : V' \to V$ such that*

- $\forall v \in V', \lambda'(v) = \lambda(f(v))$
- $\forall (u, v) \in E', (f(u), f(v)) \in E$, *and*
- $\forall (u, v) \in E', \lambda'(u, v) = \lambda(f(u), f(v))$

.

The function f is a *subgraph isomorphism* from graph G' to graph G. We say G' *occurs* in G if $G' \subseteq G$. Given a subgraph isomorphism f, the image of the domain V' $(f(V'))$ is an *embedding* of G' in G.

Example 2.1. *Figure 1 shows a graph database of three labeled graphs. The mapping (isomorphism) $q_1 \to p_3$, $q_2 \to p_1$, and $q_3 \to p_2$ demonstrates that graph Q is subgraph isomorphic to P and hence Q* occurs *in P. Set $\{p_1, p_2, p_3\}$ is an embedding of Q in P. Similarly, graph S occurs in graph P but not Q.*

Problem Statement: Given a graph space G^*, a set of n graphs sampled from G^* and the related target values of these graphs $D = \{(G_i, T_i,)\}_{i=1}^n$, the **graph classification problem** is to estimate a function $F : G^* \to T$ that accurately map graphs to their target value.

By *classification* we assume all target values are discrete values, otherwise it is a *regression* problem. Below, we review several algorithms for graph classification that work within a common framework called a kernel function. The term *kernel function* refers to an operation of computing the inner product between two points in a Hilbert space. Kernel functions are widely used in classification of data in a high dimensional feature space.

2.1 Kernel Functions for Graphs

Graph kernel functions are simply kernel functions that have been defined to compute the inner product between two graphs. In recent years a variety of graph kernel functions have been developed, with promising application results as described by Ralaviola *et al.* [25]. Among these methods, some kernel functions draw on graph features such as walks [19] or cycles [9], while others may use different approaches such as genetic algorithms [3], frequent subgraphs [6], or graph alignment [7]. We review two graph kernel functions in details due to their close relationship to our algorithm.

Kashima *et al.* [19] proposed a kernel function called the marginalized graph kernel. This kernel function is based on the use of shared label sequences in the comparison of graphs. Their marginalized graph kernel uses a Markov model to randomly generate walks of a labeled graph, based on a transition probability matrix combined with a walk termination probability. These collections of random walks are then compared and the number of shared sequences is used to determine the overall similarity between two graphs.

The optimal assignment kernel, proposed by Fröhlich *et al.* [7], differs significantly from the marginalized graph kernel in that it attempts to align two graphs, rather than compare sets of linear substructures. This kernel function first computes the similarity between all vertices in one graph and those in another. The similarity between the two graphs is then computed by finding the maximal weighted bipartite graph between the two sets of vertices, called the *optimal assignment*. The authors investigate an extension of this method whereby certain structure patterns defined *a priori* by expert knowledge, are collapsed into single vertices, and this reduced graph is used as input to the optimal assignment kernel.

3. GRAPH ALIGNMENT KERNELS

Here we present our design of a pattern diffusion kernel. We start the section by first presenting a general framework. We prove, through a reduction to the subgraph isomorphism problem, that the computational cost of the general framework can be prohibitive for large graphs. We then present our pattern based graph alignment kernel. Finally we show a technique we call "pattern diffusion" that can significantly improve graph classification accuracy in practice.

3.1 Graph Similarity Measurement with Alignment

An *alignment* of two graphs G and G' (assuming $|V[G]| \leq |V[G']|$) is a 1-1 mapping $\pi : V[G] \to V[G']$. Given an alignment π, we define the similarity between two graphs, as measured by a kernel function k_A, below:

$$k_A(G, G') = \max_{\pi} \sum_{v} k_n(v, \pi(v)) + \sum_{u,v} k_e((u, v), (\pi(u), \pi(v))) \tag{1}$$

The function k_n is a kernel function to measure the similarity of node labels and the function k_e is a kernel function to measure the similarity of edge labels. Equation 1 uses an additive model to compute the similarity between two graphs. The maximal similarity among all possible mappings is defined as the similarity between two graphs.

3.2 NP-hardness of Graph Alignment Kernel Function

It is no surprise that computing the graph alignment kernel is an NP-hard problem. We prove this with a reduction from the graph alignment kernel to the subgraph isomorphism problem. In the following paragraphs, we assume we have an efficient solver of the graph alignment kernel problem, we show that the same solver can be used to solve the subgraph isomorphism problem efficiently. Since the subgraph isomorphism problem is an NP-hard problem, with the reduction we mentioned before we prove that the graph alignment kernel problem is therefore an NP-hard problem as well. Note: this subsection is a stand-alone component of our paper, and readers who choose to skip this section should encounter no difficulty in reading the rest of the paper.

Given two graphs G and G' (for simplicity, assume nodes and edges in G and G' are not labeled as usually studied in the subgraph isomorphism problem), we use a node kernel function that returns a constant 0. We define an edge kernel function $k_e : V[G] \times V[G] \times V[G'] \times V[G'] \to \mathbb{R}$ as

$$k_e((u,v),(u',v')) = \begin{cases} 1 & \text{if } (u,v) \in E[G] \text{ and } (u',v') \in E[G'] \\ 0 & \text{otherwise} \end{cases}$$

With the constant node function and the specialized edge function, the kernel function of two graphs is simplified to the following format:

$$k_A(G,G') = \max_{\pi} \sum_{u,v} k_e((u,v),(\pi(u),\pi(v))) \qquad (2)$$

We establish the NP-hardness of the graph alignment kernel with the following theorem.

THEOREM 3.1. *Given two (unlabeled) graphs G and G' and the edge kernel function k_e defined previously, G is subgraph isomorphic to G' if and only if $K_a(G,G') = |E[G]|$*

PROOF. If: We notice from the definition of k_e that the maximal value of $K_a(G,G')$ is $|E[G]|$. Given $K_a(G,G') = |E[G]|$, we claim that there exists an alignment function $\pi : V[G] \to V[G']$ such that for all $(u,v) \in E[G]$ we have $(\pi(u),\pi(v)) \in E[G']$. The existence of such a function π guarantees that graph G is a subgraph of G'.

Only if: Given G is a subgraph of G', we have an alignment function $\pi : V[G] \to V[G']$ such that for all $(u,v) \in E[G]$ we have $(\pi(u),\pi(v)) \in E[G']$. According to Equation 2, $K_a(G,G') = |E[G]|$. □

Theorem 3.1 shows that the graph alignment kernel problem is no easier than the subgraph isomorphism problem and hence is at least NP-hard in complexity.

3.3 Graph Node Alignment Kernel

To derive an efficient algorithm scalable to large graphs, our idea is that we use a function f to map nodes in a graph to a high (possibly infinite) dimensional feature space that captures not only the node label information but also the neighborhood topological information around the node. If we have such function f, we may simplify the graph kernel function with the following formula:

$$k_M(G,G') = \max_{\pi} \sum_{v \in V[G]} k_n(f(v), f(\pi(v))) \qquad (3)$$

Where $\pi : V[G] \to V[G']$ denotes an alignment of graph G and G'. $f(v)$ is a set of "features" associated with a node.

With this modification, the optimization problem that searches for the best alignment can be solved in polynomial time. To derive a polynomial running time algorithm, we construct a weighted complete bipartite graph by making every node pair (u,v) $\in V[G] \times V[G']$ incident on an edge. The weight of the edge (u,v) is $k_n(f(v), f(u))$. In Figure 2, we show a weighted complete bipartite graph for $V[G] = \{v_1, v_2, v_3\}$ and $V[G'] = \{u_1, u_2, u_3\}$.

With the bipartite graph, a search for the best alignment becomes a search for the maximum weighted bipartite subgraph from the complete bipartite graph. Many network flow based algorithms (e.g. linear programming) can be used to obtain the maximum weighted bipartite subgraph. We use the Hungarian algorithm with complexity $O(|V[G]|^3)$. For details of the Hungarian algorithm see [1].

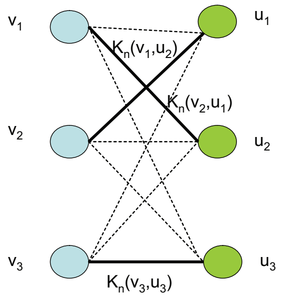


Figure 2: The maximum weighted bipartite graph for graph alignment. Highlighted edges $(v1,u2)$, $(v2,u1)$, $(v3,u3)$ **have larger weights than the rest of the edges (dashed).**

Applying the Hungarian algorithm to graph alignment was first explored by [7] for chemical compound classification. In contrast to their algorithm, which utilized domain knowledge of chemical compounds extensively and developed a complicated recursive function to compute the similarity between nodes, we develop a new framework that maps such nodes to a high dimensional space in order to measure the similarity between two nodes without assuming any domain knowledge. Even in cheminformatics, our experiments shows that our technique generate similar and sometimes better classification accuracies compared to the method reported in [7].

Unfortunately, using the Hungarian algorithm for assignment, as used by [7] is not a true Mercer kernel. Since our proposed kernel function uses this algorithm as well, it is also not a Mercer kernel. Like in [7], however, we have found that practically our kernel still performs competitively.

3.4 Pattern Diffusion

In this section, we introduce a novel function "pattern diffusion" to project nodes in a graph to a high dimensional space that captures both node labeling information and local topology information. Our design has the following advantages as a kernel function:

- Our design is generic and does not assume any domain knowledge from a specific application. The diffusion process may be applied to graphs with dramatically different characteristics.
- The diffusion process is straightforward to implement and can be computed efficiently.
- We prove that the diffusion process is related to the probability distribution of a graph random walk (in

Appendix). This explains why the simple process may be used to summarize local topological information.

Below, we outline the pattern diffusion kernel in three steps.

In the first step, we identify a seed as a starting point for the diffusion. In our design, a "seed" could be a single node, or a set of connected nodes in the original graph. In our experimental study, we use frequent subgraphs for seeds since we can easily compare a seed from one graph to a seed in another graph. However, there is no requirement that we must use frequent subgraphs.

In the second step given a set of nodes S as seed, we recursively define f_t in the following way.

The base f_0 is defined as:

$$f_0(u) = \begin{cases} 1/|S| & \text{if } u \in S \\ 0 & \text{otherwise} \end{cases}$$

Given some time t, we define f_{t+1} $(t \geq 0)$ with f_t in the following way:

$$f_{t+1}(v) = f_t(v) \times (1 - \frac{\lambda}{d(v)}) + \sum_{u \in N(v)} f_t(u) \times \frac{\lambda}{d(u)} \quad (4)$$

In the notation, $N(v)$ is the set of nodes that connects to v directly. $d(v)$ is the node degree of v, or $d(v) = |N(v)|$. λ is a parameter that controls the diffusion rate.

The formula 4 describes a process where each node distributes a λ fraction of its value to its neighbors evenly and in the same way receives some value from its neighbors. We call it "diffusion" because the process simulate the way a value is spreading in a network. Our intuition is that the distribution of such a value encodes information about the local topology of the network.

To constrain the diffusion process to a local region, we use one parameter called diffusion time, denoted by τ, to control the diffusion process. Specifically we limit the diffusion process to a local region of the original graph with nodes that are at most τ hops away from a node in the seed S. For this reason, the diffusion is referred to as "local diffusion".

Finally, for the seed S, we define the mapping function f_S as the limit function of f_t as t approaches to infinity, or

$$f_S = \lim_{t \to \infty} f_t \quad (5)$$

3.5 Pattern Diffusion Kernel and Graph Classification

In this section, we summarize the discussion of kernel function and show how the kernel function is utilized to construct an efficient graph classification algorithm at both the training and testing phases.

3.5.1 Training Phase

In the training phase, we divide graphs of the training data set $D = \{(G_i, T_i,)\}_{i=1}^{n}$ into groups according to their class labels. For example in binary classification, we have two groups of graphs: positive or negative. For multi-class classification, we have multiple groups of graphs where each group contains graphs with the same class label. The training phase is composed of four steps:

- Obtain frequent subgraphs for seeds. We identify frequent subgraphs from each graph group and union the subgraph sets together as our seed set $\mathcal{S}$.
- For each seed $S \in \mathcal{S}$ and for each graph G in the training data set, we use f_S to label nodes in G. Thus the feature vector of a node v is a vector $L_V = \{f_{S_i}(v)\}_{i=1}^{m}$ with length $m = |\mathcal{S}|$.
- For two graphs G, G', we construct the complete weighted bipartite graph as described in section 3.3 and compute the kernel $K_a(G, G')$ using Equation 3.
- Train a predictive model using a kernel classifier.

3.5.2 Testing Phase

In the testing phase, we compute the kernel function for graphs in the testing and training data sets. We use the trained model to make predictions about graph in the testing set.

- For each seed $S \in \mathcal{S}$ and for each graph G in the testing data set, we use f_S to label nodes in G and create feature vectors as we did in the training phase.
- We use Equation 3 to compute the kernel function $K_a(G, G')$ for each graph G in the testing data set and for each graph G' in the training data set.
- Use kernel classifier and trained models to obtain prediction accuracy of the testing data set

Below we present our empirical study of different kernel functions including our pattern diffusion kernel.

4. EXPERIMENTAL STUDY

We have conducted classification experiments using ten different biological activity data sets, and compared cross-validation accuracies for different kernel functions. In the following subsections, we describe the data sets and the classification methods in more detail along with the associated results.

We performed all of our experiments on a desktop computer with a 3Ghz Pertium 4 processor and 1 GB of RAM. Generating a set of frequent subgraphs is efficient, generally taking a few seconds. Computing alignment kernels somewhat takes more computation time, typically in the range of a few minutes.

In all kernel classification experiments, we used the LibSVM classifier [5] as our kernel classifier. We used nu-SVC with nu = 0.5, the LibSVM default. To perform a fair comparison, we did not perform model selection and tune the SVM parameters to favor any particular method, and used default parameters in all cases. We download the classifiers CBA and Xrule as instructed in the related papers, and used default parameters for both. Our classification accuracy is computed by averaging over ten trials of a 10-fold cross-validation experiment. Standard deviation is computed similarly.

4.1 Data Sets

We have selected ten data sets covering typical chemical benchmarks in drug design to evaluate our classification algorithm performance.

The first five data sets are from drug virtual screening experiments taken from [18]. In this data set, the target values are drugs' binding affinity to a particular protein. Five proteins are used to in the data set including: CDK2, COX2, FXa, PDE5, and A1A where each symbol represents a specific protein. For each protein, the data provider carefully selected 50 chemical structures that clearly bind to the protein ("active" ones). The data provider also deliberately listed chemical structures that are very similar to the active ones (judged with domain knowledge) but clearly do not bind to the target protein. This list is known as the "decoy" list. We randomly sampled 50 chemical structures from the decoy list. Since our goal is to evaluate classifiers, we will not further elaborate the nature of the data set. See [18] for details.

The next data set, from Wessel et al.[33] includes compounds classified by affinity for absorption through human intestinal lining. More over, we included the Predictive Toxicology Challenge[8] data sets, which contain a series of chemical compounds classified according to their toxicity in male rats, female rats, male mice, and female mice.

We use the same way as was done in [11] to transform chemical structure data set to graphs. In Table 1 for each data set, we list the total number of chemical compounds in the data set, as well as the number of positive and negative samples.

Table 1: Data set and class statistics. # G: number of samples (chemical compounds) in the data set. # P: positive samples. # N: negative samples

Dataset	# G	# P	# N
CDK2 inhibitors	100	50	50
COX2 inhibitors	100	50	50
Fxa inhibitors	100	50	50
PDE5 inhibitors	100	50	50
A1A inhibitors	100	50	50
intestinal absorption	310	148	162
toxicity (female mice)	344	152	192
toxicity (female rats)	336	129	207
toxicity (male mice)	351	121	230
toxicity (male rats)	349	143	206

4.2 Feature Sets

We used frequent patterns from graph represented chemicals exclusively in our study. We generate such frequent subgraphs from a data set using two different graph mining approaches: that with exact matching [11] and that of approximate matching. In our approximate frequent subgraph mining, we consider that a pattern *matches* with a graph as long as there are up to $k > 0$ node label mismatches. For chemical structures typical mismatch tolerance is small, that is k values are 1, 2, etc. In our experiments we used approximate graph mining with $k = 1$.

Once frequent subgraphs are mined, we generate three feature sets: (i) general subgraphs (all of mined subgraphs), (ii) tree subgraphs, and (iii) path subgraphs. We tried cycles as well, but did not include them in this study since typically less than two cyclic subgraphs were identified in a data set. These feature sets are used for constructing kernel functions as discussed below.

4.3 Classification Methods

We have evaluated the performance of the following classifiers.

- CBA. The first is a classifier that uses frequent itemset mining, known as Classification Based on Association (CBA) [4]. In CBA we treat mined frequent subgraphs as item sets.
- Graph Convolution Kernels. This type of kernel include the mismatch kernel (MIS) and the min-max (MNX) kernel. The former is based on the normalized Hamming distance of two binary vectors, and the latter is computed as the ratio between two sums: the numerator is the sum of the minimum between each feature pair in two binary vectors, and the denominator is the same except it sums the maximum. See [32] for details about the min-max kernel.
- SVM built-in Kernels. We used linear kernel (Linear) and radial basis function (RBF) kernel.
- GPD. We implemented the graph pattern diffusion kernel as discussed in Section 3. The default parameter for the GPD kernel is a diffusion rate of λ =20% and the diffusion time $\tau = 5$.

4.4 Experimental Results

Here we present the results of our graph classification experiments. We perform one round of experiments to evaluate the methods based on exact subgraph mining, and another round of experiments with approximate subgraph mining. For both of these two subgraph mining methods, we selected patterns that were general graphs, tree graphs, and cycles.

We perform a simple feature selection in order to identify the most discriminating frequent patterns. Using a simple statistical formula, Pearson correlation coefficient (PCC), we measure the correlation between a set of feature samples (in our case, the occurrences of a particular subgraph in each of the data samples) and the corresponding class labels. Frequent patterns are ranked according to correlation strength, and the top 10% patterns are selected to construct the feature set.

4.4.1 Comparison between classifiers

The results of the comparison of different graph kernel functions are shown in Table 3. For this results, we used frequent subgraph mined using exact matching. From the table using general subgraphs (the first 10 rows in Table 3), we observe that for exact mining of general subgraphs, in 4 of the 10 data sets, our GPD method provides mean accuracy that is significantly better (at least two standard deviations above the next best method). In another 4 data sets GPD gives the best performance, but the difference is less significant but is still more than 1 standard deviation). In the last two data sets other methods perform better, but not significantly better. The mismatch and min-max kernels all give roughly the same performance and hence we only show the results of the mismatch kernel. The GPD's superiority is also confirmed in classifications where tree and path patterns are used.

In Table 2 we compare the performance of our GPD kernel to the CBA method, or Classification Based on Association. In general it shows comparable performance to the other methods. In one data set it does show a noticeable increase over the other methods. This is expected since CBA is designed specifically for discrete data such as the binary feature occurrences used here. Despite the strengths of CBA, we can see that GDA method still gives the best performance for 6 of the seven data sets. We also tested these data sets using the recursive optimal-assignment kernel included in the JOELib2 computational chemistry library. Its results are comparable to those of the CBA method and hence were not included as separate results here.

Table 2: Comparison of GPD and CBA.

Data set	GPD	CBA
CDK2 inhibitors	88.6*	80.46
COX2 inhibitors	82.7*	77.86
Fxa inhibitors	89.3*	86.87
PDE5 inhibitors	81.9	87.14*
A1A inhibitors	91.4*	87.76
intestinal absorption	63.14*	54.36
toxicity (male rats)	56.66*	55.95

In addition we tested a classifier called XRules. XRules is designed for classification of tree data [38]. Chemical graphs, while not strictly trees, often are close to trees. To run the XRules executable, we transform a graph to a tree by randomly selecting a spanning tree of the original graph. Our experimental study shows the application of XRules on average delivers incompetent results among the group of classifiers (e.g. 50% accuracy on the CDK2 inhibitor data set), which may be due to the particular way we transform a graph to a tree. Since we compute tree patterns for rule based classifier such as CBA in our comparison, we did not explore further of XRules.

We also tested a method based on a recursive optimal-assignment [7] using biologically-relevant chemical descriptors labeling each node in a chemical graph. In order to perform a fair comparison with this method to the other methods we chose to ignore the chemical descriptors and focus on the structural alignment. In our experiments the performance of this method is very similar to CBA and hence we show results of CBA only.

4.4.2 *Comparison Between Descriptor Sets*

Various types of subgraphs such as trees, paths, and cycles have been used in kernel functions between chemical compounds. In addition to exact mining of general subgraphs, we also chose to use approximate subgraph mining to generate the features for our respective kernel methods. In both cases we filtered the general subgraphs mined into sets of trees and sets of paths as well. The results for these experiments are given in Tables 2 and 3 above.

From Table 2 we see that the results for all kernels using exact tree subgraphs are identical to those for exact general subgraphs. This is not surprising, given that most chemical fragments are structured as trees. The results using exact path subgraphs, however, do show some shifts in accuracy but the difference is not significant.

The results using approximate subgraph mining (shown in Table 4) are similar to those for exact subgraph mining (shown in Table 3). In contrast to our hypothesis that using approximate subgraph mining might improve the classification accuracy, the data show that there is no significant difference between the set of features. However, it is clearly that GPD is still better than the competing kernel functions.

4.4.3 *Effect Of Varying GPD Diffusion Rate And Time*

We want to evaluate the sensitivity of the GPD methods to its two parameters: diffusion rate λ and diffusion time. We tested different diffusion rate λ values and diffusion time values. Figure 3 shows that the GPD algorithm is not very sensitive to the two parameters at the range that we tested. Although we show only three data sets in Figure 3, the observation is true for other data sets in our experiments.

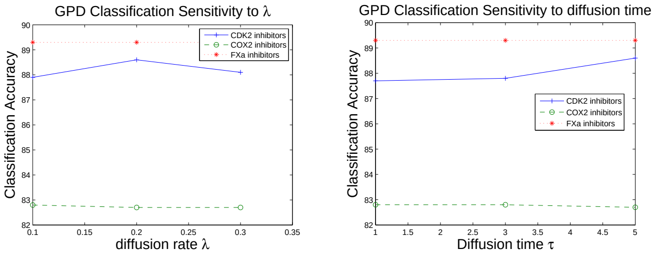


Figure 3: Left: GPD Classification Accuracy with different diffusion rate. Right: GPD Classification Accuracy with different diffusion time.

5. CONCLUSIONS AND FUTURE WORKS

With the rapid development of fast and sophisticated data collection methods, data has become complex, high-dimensional and noisy. Graphs have proven to be powerful tools for modeling complex, high-dimensional and noisy data; building highly accurate predictive models for graph data is a new challenge for the data mining community. In this paper we have demonstrated the utility of a novel graph kernel function, graph pattern diffusion kernel (GPD kernel). We showed that the GPD kernel can capture the intrinsic similarity between two graphs and has the lowest testing error in many of the data sets evaluated. Although we have developed a very efficient computational framework, computing a GPD kernel may be hard for large graphs. Our future work will concentrate on improving the computational efficiency of the GPD kernel for very large graphs, as well as performing additional comparisons between our method other 2D-descriptor and QSAR-based methods.

Acknowledgments

This work has been supported by the Kansas IDeA Network for Biomedical Research Excellence (NIH/NCRR award #P20 RR016475) and the KU Center of Excellence for Chemical Methodology and Library Development (NIH/NIGM award #P50 GM069663)

6. REFERENCES

[1] R. Ahuja, T. Magnanti, and J. Orlin. Network flows. *SIAM Review*, 37 No.1, 1995.

[2] C. Austin, L. Brady, T. Insel, and F. Collins. Nih molecular libraries initiative. *Science*, 306(5699):1138–9, 2004.

[3] E. Barbu, R. Raveaux, H. Locteau, S. Adam, and P. Heroux. Graph classification using genetic algorithm and graph probing application to symbol recognition. *Proc. of the 18th International Conference on Pattern Recognition (ICPR)*, 2006.

Table 3: Comparison of Different Graph Kernel Functions Using Different Subgraph Feature minded by FFSM. * denotes the best classification accuracy. Standard deviations are provided in the second column.

subgraph type	data set	MIS		GPD		Linear		RBF	
general	CDK2 inhibitors	76.3	2.06	87.2*	2.04	76.3	2.06	77.9	1.6
	COX2 inhibitors	85.1*	0.99	83.2	0.79	85.1*	0.99	84.5	1.08
	FXa inhibitors	87	0.94	87.6*	0.52	87	0.94	86.2	0.42
	PDE5 inhibitors	83.2*	0.63	82.8	1.4	83.2*	0.63	83	0.67
	A1A inhibitors	84.8	0.63	90.9*	0.74	85	0.94	88.7	1.06
	intestinal absorption	49.53	4.82	56.86*	3.12	50.7	4.56	47.56	3.44
	toxicity (female mice)	51.46	3.4	54.81*	1.16	51.95	3.26	50.95	2.75
	toxicity (female rats)	52.99	4.33	56.35*	1.13	49.57	4.71	51.94	3.34
	toxicity (male mice)	49.64	3.43	60.71*	1.16	49.38	1.96	51.16	2.28
	toxicity (male rats)	50.44	3.06	56.83*	1.17	49.91	3.09	54.3	2.59
trees	CDK2 inhibitors	76.3	2.06	87.2*	2.04	76.3	2.06	77.9	1.6
	COX2 inhibitors	85.1*	0.99	83.2	0.79	85.1*	0.99	84.5	1.08
	FXa inhibitors	87	0.94	87.6*	0.52	87	0.94	86.2	0.42
	PDE5 inhibitors	83.2*	0.63	82.8	1.4	83.2*	0.63	83	0.67
	A1A inhibitors	84.8	0.63	90.9*	0.74	85	0.94	88.7	1.06
	intestinal absorption	49.53	4.82	56.86*	3.12	50.7	4.56	47.56	3.44
	toxicity (female mice)	51.46	3.4	54.81*	1.16	51.95	3.26	50.95	2.75
	toxicity (female rats)	52.99	4.33	56.35*	1.13	49.57	4.71	51.94	3.34
	toxicity (male mice)	49.64	3.43	60.71*	1.16	49.38	1.96	51.16	2.28
	toxicity (male rats)	50.44	3.06	56.83*	1.17	49.91	3.09	54.3	2.59
paths	CDK2 inhibitors	76.3	0.82	86.2*	2.82	76.4	0.97	77.1	0.74
	COX2 inhibitors	85*	0	83.7	0.48	85*	0	85*	0
	FXa inhibitors	86.8	0.79	87.6*	0.52	86.8	0.79	86.6	0.84
	PDE5 inhibitors	82.6	0.84	83*	1.25	82.6	0.84	82.7	0.95
	A1A inhibitors	84.1	0.88	91.2*	1.14	84	0.67	85.7	0.67
	intestinal absorption	49.07	7.16	54.07*	3.52	50.58	4.32	50	4.72
	toxicity (female mice)	50.14	3.41	54.79*	2.13	50.37	2.59	50.14	4.38
	toxicity (female rats)	47.83	6.85	55.93*	2.44	48.32	7.83	50.09	4.37
	toxicity (male mice)	46.85	3.57	58.81*	1.07	48.6	4.78	50.33	2.29
	toxicity (male rats)	50.26	3.13	54.71*	1.38	48.69	3.93	54.27	3.04

Table 4: Comparison of Different Graph Kernel Functions Using Different Subgraph Feature minded by approximate matching. Standard deviations are provided in the second column.

subgraph type	data set	MIS		GPD		Linear		RBF	
general	CDK2 inhibitors	76.3	2.06	85.7*	1.49	76.3	2.06	77.9	1.6
	COX2 inhibitors	85*	0	83	0.67	85*	0	85*	0
	FXa inhibitors	86.4	0.52	87.5*	0.53	86.4	0.52	86.1	0.32
	PDE5 inhibitors	83.3*	0.67	83.3*	1.64	83.3*	0.67	82.9	0.74
	A1A inhibitors	86.2	1.81	88.7*	0.82	86.2	1.81	88.7	0.48
	intestinal absorption	51.28	4.3	60.81*	2.63	52.67	4.07	51.86	6.18
trees	CDK2 inhibitors	76.3	2.06	85.7*	1.49	76.3	2.06	77.9	1.6
	COX2 inhibitors	85*	0	83	0.67	85*	0	85*	0
	FXa inhibitors	86.4	0.52	87.5*	0.53	86.4	0.52	86.1	0.32
	PDE5 inhibitors	83.3*	0.67	83.3*	1.64	83.3*	0.67	82.9	0.74
	A1A inhibitors	86.2	1.81	88.7*	0.82	86.2	1.81	88.7*	0.48
	intestinal absorption	51.28	4.3	60.81*	2.63	52.67	4.07	51.86	6.18
paths	CDK2 inhibitors	76.3	0.82	86.1*	2.13	76.4	0.97	77.1	0.74
	COX2 inhibitors	85*	0	83.4	0.7	85*	0	85*	0
	FXa inhibitors	86	0	88*	0.82	86	0	86	0
	PDE5 inhibitors	83.1	0.57	83.8*	2.53	83.1	0.57	82.9	0.57
	A1A inhibitors	83.6	0.7	88.6*	0.7	83.6	0.7	85.7	0.67
	intestinal absorption	49.88	4.3	60.23*	4.34	51.05	3.82	49.65	3.76

[4] Y. M. Bing Liu, Wynne Hsu. Integrating classification and association rule mining. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, 1998.

[5] C. Chang and C. Lin. Libsvm: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[6] M. Deshpande, M. Kuramochi, and G. Karypis. Frequent sub-structure-based approaches for classifying chemical compounds. *IEEE Transactions on Knowledge and Data Engineering*, 2005.

[7] Fröohlich, J. Wegner, F. Sieker, and A. Zell. Kernel functions for attributed molecular graphs - a new similarity-based approach to adme prediction in classification. QSAR & Combinatorial Science, 2006.

[8] C. Helma, R. King, and S. Kramer. The predictive toxicology challenge 2000-2001. *Bioinformatics*, 17(1):107–108, 2001.

[9] T. Horvath, T. Gartner, and S. Wrobel. Cyclic pattern kernels for predictive graph mining. *SIGKDD*, 2004.

[10] T. Horvath, J. Ramon, and S. Wrobel. Frequent subgraph mining in outerplanar graphs. In *SIGKDD*, 2006.

[11] J. Huan, W. Wang, and J. Prins. Efficient mining of frequent subgraph in the presence of isomorphism. In *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM)*, pages 549–552, 2003.

[12] J. Huan, W. Wang, J. Prins, and J. Yang. SPIN: Mining maximal frequent subgraphs from graph databases. pages 581–586, 2004.

[13] J. Huan, W. Wang, A. Washington, J. Prins, R. Shah, and A. Tropsha. Accurate classification of protein structural families based on coherent subgraph analysis. In *Proceedings of the Pacific Symposium on Biocomputing (PSB)*, pages 411–422, 2004.

[14] Y. Huang, H. Li, H. Hu, X. Yan, M. S. Waterman, H. Huang, and X. J. Zhou. Systematic discovery of functional modules and context-specific functional annotation of human genome. *Bioinformatics*, pages ISMB/ECCB Supplement, 222–229, 2007.

[15] A. Inokuchi, T. Washio, and H. Motoda. An apriori-based algorithm for mining frequent substructures from graph data. *In PKDD'00*, pages 13–23, 2000.

[16] W. W. J. Huan and J. Prins. Efficient mining of frequent subgraphs in the presence of isomorphism. *In Proc. of ICDM*, 2003.

[17] R. Jin, S. Mccalle, , and E. Almaas. Trend motif: A graph mining approach for analysis of dynamic complex networks. *ICDM*, 2007.

[18] R. Jorissen and M. Gilson. Virtual screening of molecular databases using a support vector machine. *J. Chem. Inf. Model.*, 45(3):549–561, 2005.

[19] H. Kashima, K. Tsuda, and A. Inokuchi. Marginalized kernels between labeled graphs. In *Proc. of the Twentieth Int. Conf. on Machine Learning (ICML)*, 2003.

[20] Y. Ke, J. Cheng, , and W. Ng. Correlation search in graph databases. In *SIGKDD*, 2007.

[21] T. Kudo, E. Maeda, and Y. Matsumoto. An application of boosting to graph classification. In *NIPS*, 2004.

[22] M. Kuramochi and G. Karypis. Frequent subgraph discovery. In *Proc. International Conference on Data Mining'01*, pages 313–320, 2001.

[23] S. Nijssen and J. Kok. A quickstart in frequent structure mining can make a difference. In *Proceedings of the 10th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 647–652, 2004.

[24] J. R. Quinlan. *C4.5 : Programs for Machine Learning.* Morgan Kaufmann, 1993.

[25] L. Ravaliola, S. J. Swamidass, and H. Saigo. Graph kernels for chemical informatics. *Neural Networks*, 2005.

[26] D. Shasha, J. T. L. Wang, and R. Giugno. Algorithmics and applications of tree and graph searching. In *Proceeding of the ACM Symposium on Principles of Database Systems (PODS)*, 2002.

[27] A. Smalter, J. Huan, and G. Lushington. Structure-based pattern mining for chemical compound classification. In *Proceedings of the 6th Asia Pacific Bioinformatics Conference*, 2008.

[28] J. Sun, S. Papadimitriou, P. S. Yu, and C. Faloutsos. Parameter-free mining of large time-evolving graphs. In *SIGKDD*, 2007.

[29] N. Tolliday, P. A. Clemons, P. Ferraiolo, A. N. Koehler, T. A. Lewis, X. Li, S. L. Schreiber, D. S. Gerhard, and S. Eliasof. Small molecules, big players: the national cancer institute's initiative for chemical genetics. *Cancer Research*, 66:8935–42, 2006.

[30] H. Tong, Y. Koren, , and C. Faloutsos. Fast direction-aware proximity for graph mining. In *SIGKDD*, 2007.

[31] V. Vapnik. *Statistical Learning Theory*. John Wiley, 1998.

[32] N. Wale, I. Watson, , and G. Karypis. Comparison of descriptor spaces for chemical compound retrieval and classification. *Knowledge and Information Systems*, 2007.

[33] M. Wessel, P. Jurs, J. Tolan, and S. Muskal. Prediction of human intestinal absorption of drug compounds from molecular structure. *J. Chem. Inf. Comput. Sci.*, 38(4):726–735, 1998.

[34] J. Weston, R. Kuang, C. Leslie, and W. S. Noble. Protein ranking by semi-supervised network propagation. *BMC Bioinformatics*, 2006.

[35] D. Williams, J. Huan, and W. Wang. Graph database indexing using structured graph decomposition. In *in Proceedings of the 23rd IEEE International Conference on Data Engineering (ICDE)*, 2007.

[36] X. Yan and J. Han. gspan: Graph-based substructure pattern mining. In *Proc. International Conference on Data Mining'02*, pages 721–724, 2002.

[37] X. Yan, P. S. Yu, and J. Han. Graph indexing based on discriminative frequent structure analysis. In *ACM Transactions on Database Systems (TODS)*, 2005.

[38] M. J. Zaki and C. C. Aggarwal. Xrules: An effective structural classifier for xml data. *Machine Learning Journal special issue on Statistical Relational Learning and Multi-Relational Data Mining*, 62, No. 1-2:137–170, 2006.

[39] Z. Zeng, J. Wang, L. Zhou, and G. Karypis. Coherent closed quasi-clique discovery from large dense graph databases. In *SIGKDD*, 2006.

[40] D. Zhou, J. Huang, and B. Schöolkopf. Learning from labeled and unlabeled data on a directed graph. *Proceedings of the 22nd International Conference on Machine Learning*, 2005.

APPENDIX

A. CONNECTION OF PATTERN DIFFUSION TO MARGINALIZED GRAPH KERNEL

Here we show the connection of pattern diffusion kernel function to the marginalized graph kernel [19], which uses a Markov model to randomly generate walks of a labeled graph.

Given a graph G with nodes set $V[G] = \{v_1, v_2, \ldots, v_n\}$, and a seed $S \subseteq V[G]$, for each diffusion function f_t, we construct a vector $U_t = (f_t(v_1), f_t(v_2), \ldots, f_t(v_n))$. According to the definition of f_t, we have $U_{t+1} = \Gamma \times U_t$ where the matrix Γ is defined as:

$$\Gamma(i,j) = \begin{cases} \frac{\lambda}{d(v_j)} & \text{if } i \neq j \text{ and } i \in N(j) \\ 1 - \frac{\lambda}{d(v_i)} & i = j \\ 0 & \text{otherwise} \end{cases}$$

In this representation, we compute the stationary distribution ($f_S = \lim_{t\to\infty} f_t$) by computing $\Gamma^\infty \times U_0$.

We notice that the matrix Γ corresponds to a probability matrix corresponding to a Markov Chain since

- all entries are non-negative
- column sum is 1 for each column

Therefore the vector $\Gamma^\infty \times U_0$ corresponds to the stationary distribution of the local random walk as specified by Γ. In other words, rather than using random walk to retrieve information about the local topology of a graph, we use the stationary distribution to retrieve information about the local topology. Our experimental study shows that this in fact is an efficient way for graph classification.

Reinforcing Mutual Information-based Strategy for Feature Selection for Microarray Data

Jian Tang
Department of Computer Science
Memorial University of Newfoundland
St. John's, NL, Canada
jian@cs.mun.ca

Shuigeng Zhou Feng Li Kai Jiang
School of Computer Science, and
Shanghai Key Lab of Intelligent Information Processing
Fudan University, Shanghai 200433, China
{sgzhou,fengli2006,jiangkai}@fudan.edu.cn

ABSTRACT

Mutual information (MI) is a powerful concept for correlation-centric applications. Recently, it has been used for feature selection for microarray gene expression data. One of the merits of MI is, unlike many other heuristic methods, it is based on a mature theoretic foundation. When applied to microarray data, however, it faces a number of challenges. Firstly, due to large numbers of features (i.e., genes) present in microarrays, the true distributions for the expression values of some genes may be masked by noises. Secondly, evaluating inter-group mutual information requires estimating multi-variate distributions, which is difficult. To address the first problem, we use a scheme called Substantial relevance boosting, which requires a non-noisy feature to show *substantially additional relevance* with class labeling beyond the already selected features. To address the second problem, we use *Increasing likelihood of feature interaction*, which probabilistically compensates for feature interaction missing from simple aggregation-based simulation. We justify our strategies from both a theoretical perspective, and the experimental results on real life data sets, which show the improved effectiveness of our method over the existing schemes.

Categories and Subject Descriptors

I.5.2 [**Computing Methodologies**]: Design Methodology—*feature evaluation and selection*

General Terms

Algorithms

Keywords

Feature selection, gene expression profiling, mutual information, classification, feature interaction

1. INTRODUCTION

Due to large numbers of features (i.e., genes) in microarray gene expression data, some genes may seem to be relevant to the class labeling, but in fact are biologically irrelevant. These 'noisy' features can mislead the learning process, rendering traditional classification methods not being effective. Feature selection is a process to select truly relevant features with the class labeling. When the purpose of learning is to classify unlabeled samples, the selected feature subset with small size is desirable. There are some good reasons for this. Small feature sizes imply less redundancies, which can improve inferences and classifications. Small feature sizes are less likely to over-fit, and therefore can increase generalization capability for classifiers. This is beneficial in clinical settings. In disease profiling, a small sized gene set that enables accurate classification may be potential diagnostic or prognostic markers. Because of this, minimizing the sizes of feature sets that enable accurate classifications has been the goal of a large number of existing works on feature selection for gene expression data.

Mutual information (MI) is a powerful concept for correlation-centric applications. Recently, it has been used for feature selection for microarray gene expression data [4, 9, 10, 17]. One of the merits of MI is that, unlike many other heuristic methods, it has a mature theoretic foundation [8]. For a data set where many noisy features are present, however, its power may diminish. In these data sets, the mutual information evaluated for noisy features may not reflect the true facts. Another problem with MI in this context arises from evaluation of inter-group mutual information. This normally requires estimating multivariate distributions, which in the general case is difficult. Existing MI-based feature selection methods rank features according to their individual mutual information with class labeling, with the hope that noisy features will be filtered out automatically. Redundancies are evaluated based on mutual information between a candidate feature and the group of already-selected features. Estimation of multi-variate distribution is avoided by using simple aggregation (e.g., averaging) over individual mutual information as an approximation for group mutual information. Since the simple aggregation is based on mutual information individually evaluated, it is incapable of simulating mutual information arising from *feature interactions.* (Refer to Section 2.2.4 for detail.)

MI-based feature selection belongs to the general category of *individual-oriented* scoring methods, which score genes individually based on their ability to discriminate different classes, and measure redundancies using feature-feature correlation [4, 6, 9, 11, 16, 17, 19, 21]. These schemes all shared the two problems mentioned previously. A different category of *group-oriented* methods associates scores

BIOKDD '08, August 24, 2008, Las Vegas, Nevada, USA.
Copyright 2008 ACM 978-1-60558-302-0 ...$5.00.

with groups, rather than individual features [1, 5, 12, 15, 20]. Higher scores are assigned to groups with higher relevance with class labeling and smaller redundancies among the group members. The actual scoring mechanisms used are quite diverse. Focus [1] systematically searches for a feature set that has consistent discriminative power with the full feature set. Its search method is essentially exhaustive, and costly in high dimensional space. In [15], the authors introduce Markov-blanket to detect effectively redundant features at the conceptual level. It is however difficult to implement in practice since it requires estimating multivariate densities. The same problem is shared by the work in [5]. Some work relaxes this requirement. The correlation-based feature selection [12] uses a metric that requires only estimating bi-variant densities, at the price of reduced capability of describing relevance and redundancy. The fast-correlation-based feature selection proposed in [23] approximates Markov blanket. It is a powerful model. However, it requires estimating tri-variate densities. The scheme in [24] uses as the group criteria the leave-one-out cross validation errors based on least-square SVM (i.e., LS-SVM) classifiers. The error is actually generated by solving a linear system of equations for the training set, and therefore avoids repeated LOOCV applications. However, solving the linear system of equations is still sometime time consuming. Also, it is unclear how this technique can be applied to data sets with more than two classes. The work in [22] uses principle component technique to select features. However, there is no discussion of how effective this approach is in dealing with redundancies. Recently, some work studies margin-based schemes, which score features based on the maximum margins they can exert between different classes [18]. They are essentially individual-oriented methods mentioned previously, but can be conveniently extended to kernel spaces [3, 7, 20]. However, these schemes normally are not concerned with redundancies.

In this paper, we propose a group-oriented, MI-based feature selection scheme. Our scheme uses explicit mechanisms to relieve the two problems mentioned previously. To address the first problem, we use a mechanism called *Substantial relevance boosting*, which requires a non-noisy feature to show substantially additional relevance with class labeling beyond the already selected features. To address the second problem, we use *Increasing likelihood of feature interaction*, which probabilistically compensates for feature interaction missing from simple aggregations. In our scheme, it is only required to estimate bi-variant densities. We justify our strategies from both a theoretical perspective, and the experimental results on real life data sets, which show the improved effectiveness of our method over the existing schemes.

The rest of paper is organized as follows. In Section 2, we provide basic concepts that will be used in the subsequent sections, and then introduce our method. In section 3, we present the experimental results. We conclude the paper by summarizing the main results.

2. SELECTING RELEVANT FEATURES

2.1 A metric for relevancy and redundancy

Our definitions relating to mutual information are based on the concepts from information theory [8]. Let X and Y be two variables (features) and C be a set of class labels. The *entropy of* X is defined as: $H(X) = -\sum_{Dom(X)} p(x)log(p(x))$. The *conditional entropy of* X *given* Y is $H(X|Y) = -\sum_{Dom(X)}\sum_{Dom(Y)} p(x,y)log(p(x|y))$. The entropy of a variable measures the amount of uncertainty of the variable. (If X and/or Y are sets of variables, the above definitions still hold.) The *mutual information of* X *and* Y is: $I(X;Y) = H(X) - H(X|Y)$. Note that $I(X;Y) = I(Y;X) = H(X) + H(Y) - H(X,Y)$. The *mutual information of* X *and* Y *given* Z is: $I(Y;X|Z) = H(Y|Z) - H(Y|Z,X)$. Also note that we have $I(X;Y|Z) = I(Y;X|Z) = H(X|Z) + H(Y|Z) - H(X,Y|Z)$. Let C be the target class. We say $I(X;C)$ is the *relevance of* X *to* C. Let X and Y be features, we also use $I(X;Y)$ to measure the *amount of redundancy of* X *and* Y. It is easy to verify that if X and Y are independent, then their redundancy is 0. On the other hand, if they are fully dependent of each other, i.e., they are (essentially) duplicates, then their redundancy is $H(X)$ or $H(Y)$.

2.2 Feature selections based on mutual information

2.2.1 General criteria

Let C be the target class, and F be the entire set of features. We can view C and each feature in F as a random variable, and F as an n-variate. In theory, our goal is to search for a subset $D = arg\max_{S\subseteq F} I(S;C)$. That is, D is the subset most relevant to the target class. In the general case, the desired subset is not unique. We would also like it to be of a smallest size, implying that it contains least redundancy. In practice, however, that expression is difficult to evaluate since the size of the search space is $2^{|F|}$. To reduce the search space, we may use a local optimization criterion. Let S be the current feature set, which is initialized to ϕ. We search for the next feature g to be included into S if:

$$g = arg \max_{e\in F-S}\{I(e;C|S)\} \quad (1)$$

$$I(g;C|S) > 0 \quad (2)$$

The conditional mutual information of e and C given S is the information that e has about C given S. This is the information that is not subsumed by the information given in S about C. Therefore, both relevance and redundancy have been taken care of. We select g if it results in this information being positive and maximized. With the above criterion, we can use a forward hill climbing method to search for features one at a time in an incremental fashion, and therefore drastically cut the search space, at the price of returning local optimum only. However, if we use that criterion as is for microarray data, a problem may arise, as described in the next section.

2.2.2 Substantial relevance boosting

From the information theory, $I(e;C|S) = 0$ if and only if e is conditionally independent of C given S. If e is indeed conditionally independent of C given S, and the distribution demonstrated by the training set is a good approximation of the true distribution for the variables, then we can expect that e will be close to being conditionally independent of C given S reflected from the training set. However, as mentioned previously, due to the fact that genes in typical microarray datasets greatly outnumber sample examples, the distributions described by the training set may not accurately reveal the true distributions of the variables. Thus a biologically irrelevant or redundant gene may turn out not to be very close to being conditionally independent of the class labeling, resulting in $I(s;C|S)$ not being close to zero when estimated based on the values in the training set. We now make the following assumption:

> If e is biologically irrelevant to C given S, then $I(e;C|S)/H(C|S)$ based on the microarray dataset will not be *substantially larger than 0.*

In other words, we allow a true conditional irrelevant gene to deviate in the distribution reflected in the training set from its true distribution for certain amount, which should not be too large. We believe this assumption is reasonable for most of the microarray datasets which have a priori gone through pre-processing stages, in which many artifacts and variations have been eliminated. The amount of deviation is measured by the ratio $I(e;C|S)/H(C|S)$, which is proportional to $I(e;C|S)$, but inversely proportional to $H(C|S)$. The rationale for the former is easy to see: should e's true distribution be preserved in the training set, then we would have $I(e;C|S) = 0$. Thus, a larger value for $I(e;C|S)$ signifies a larger departure from e's true distribution. For the latter, recall that $H(C|S)$ is the amount of uncertainty that remains in C when S is present, and $I(e;C|S) = H(C|S) - H(C|S,e)$ is the amount of uncertainty further deducted from C when e is present. Such a further deduction requires e to have the information that S does not have. A low value of $H(C|S)$ implies S contains a lot of information about C, and therefore requires e to deviate more to possess additional information to what S already has.

Based on the above assumption, we replace condition (2) by a stronger condition:

$$I(g;C|S)/H(C|S) > \alpha \tag{3}$$

where α is a threshold set by users. Note that it is easy to verify $I(g;C|S) = I(S,g;C) - I(S;C)$. Let S be a set of features already selected. According to our terminologies, we can view $I(g;C|S)$ as a measure of the relevance with C by in addition to $I(S;C)$ due to g's presence. Thus we call condition (3) *substantial relevance boosting*, where α implements substantiality. A critical issue, of course, is how to set a proper value for α. Too large a value would filter out not only irrelevant features, but also some relevant ones, while too small a value would weaken the filtering power. We have done extensive testing, and found that the best value for α is between 0.05 and 0.15, and is almost invariant for all the data sets in our experiment. (Refer to Section 3.)

2.2.3 Computing relevance

A practical issue is how to estimate a conditional mutual information. Our goal is to develop approximations that preserve its basic characteristics, and at the same time possess reasonable applicability. We first introduce an assertion.

Assertion 1. *Let $S \subset F$ and $e \in F - S$. Then*

$$I(e;C|S) = I(S,C;e) - I(S;e) \tag{4}$$

Proof. The chain rule of entropy states that, for any variable A, and set of variables R, it is true that $H(R,A) = H(R) + H(A|R)$. Based on this, we have

$$\begin{aligned}
&I(e;C|S)\\
&= H(C|S) - H(C|S,e)\\
&= H(C,S) - H(S) - (H(C,S,e) - H(S,e))\\
&= H(S,C) + H(e) - H(S,e,C) - (H(S) + H(e) - H(S,e))\\
&= I(S,C;e) - I(S;e) \quad \square
\end{aligned}$$

Thus, given feature set S and labeling C, maximizing $I(e;C|S)$ is equivalent to maximizing the difference on the right side of (4). Since we are unable to actually maximize that difference, we will use approximations for the two terms on the right hand side of (4). In the following, for any X and Y, we use $\widehat{I}(X;Y)$ for the approximation of $I(X;Y)$ calculated using a different formula from its definition.

$$\widehat{I}(S,C;e) = \frac{|S|}{|S|+1}I(S;e) + \frac{1}{|S|+1}I(C;e) \tag{5}$$

$$\widehat{I}(S;e) = \frac{1}{|S|}\sum_{x\in S} I(x;e) \tag{6}$$

Substituting the right side of formulas (5) into (4), and then replace $I(S;e)$ by $\widehat{I}(S;e)$, we have the following

$$\widehat{I}(e,S;C) - \widehat{I}(S;C) = \frac{1}{|S|+1}I(e;C) - \frac{1}{|S|+1}\widehat{I}(e;S) \tag{7}$$

Formula (7) will be used to simulate formula (4). Observe that larger value for $\widehat{I}(e;S)$, or smaller value for $\widehat{I}(e;C)$, results in smaller amount to be added to the (approximated) relevance of S to C. If we use $\widehat{I}(e,S;C) - \widehat{I}(S;C)$ to simulate $I(e;C|S)$, we must maximize the former. This is equivalent to maximizing $I(e;C) - \widehat{I}(e;S)$. Note that the last maximization operation requires estimating bi-variant densities only. Now the question is: how do we justify maximizing $\widehat{I}(e,S;C) - \widehat{I}(S;C)$ in terms of the effectiveness of feature selection, not just in terms of the effectiveness of multi-variate density estimations. After all, our ultimate purpose is to maximize $I(e;C|S)$. From formula (5) and (6), it is not immediately clear how the maximum value for $\widehat{I}(e,S;C) - \widehat{I}(S;C)$ can boost the value of $I(e;C|S)$.

Assertion 2. *The following inequality is true:*

$$I(e;C|S) \le I(e;C) - \widehat{I}(e;S) + H(S|C) \tag{8}$$

Proof. First, by chain rule for mutual information, $I(S,C;e) = I(C;e) + I(S;e|C)$, we have the following:

$$\begin{aligned}
&I(S,C;e) - I(S;e)\\
&= I(C;e) - I(S;e) + I(S;e|C)\\
&= I(C;e) - I(S;e) + H(S|C) - H(S|C,e)\\
&\le I(C;e) - I(S;e) + H(S|C)\\
&= I(e;C) - I(e;S) + H(S|C)
\end{aligned}$$

On the other hand, by the information theory, for any $x \in S$, $I(e;S) \ge I(e;x)$. Thus $I(e;S) \ge \frac{1}{|S|}\sum_{x\in S} I(e;x) = \widehat{I}(e;S)$. This implies $I(S,C;e) - I(S;e) \le I(e;C) - \widehat{I}(e;S) + H(S|C)$. Inequality (8) then follows from Assertion 1 $\square$

Assertion 2 establishes an upper bound for $I(e;C|S)$. When S and C are given, the term $H(S|C)$ is a constant. Thus, maximizing $I(e;C) - \widehat{I}(e;S)$ actually maximizes this upper bound, which approximates the criterion of maximizing $I(e;C|S)$. The expected effect of this approximation is the high likelihood that a large value for $I(e;C|S)$ will be obtained. In the following section, we will use a strategy that additionally promotes such a likelihood.

2.2.4 Increasing Likelihood for Features Interacting

Let us analyze formula $I(e;C|S) = I(S,e;C) - I(S;C)$ more closely. It is the additional information to what S already has about C, that e can generate *in cooperation with S*. This additional information may contain a portion called interacting information [13, 14].(Its precise definition is complex, and not important in our discussion.) It is only necessary to note that the interacting information of S and e about C can exist only in $I(S,e;C)$(or $I(e;C|S)$), but never in either $I(S;C)$ or $I(e;C)$. Since it is difficult to evaluate directly $I(S,e;C)$, all the existing approximation methods have the effects only of increasing directly the non-interacting information, and heuristically the interacting information, of $I(e;C|S)$. (There are a few work, however, which attempt to increase directly the interacting information as well for $|S| = 1$ [16].)

Our method basically follows this line also. Consider formula (8) again. Maximizing $I(e;C) - \widehat{I}(e;S)$ implies $I(e;C)$ tends to be large. This has the effect of boosting the non-interacting information in $I(S,e;C)$. On the other hand, maximizing the above difference maximizes the upper bound of $I(e;C|S)$, and hence it has the effect of increasing the probability of boosting interacting information in $I(e;C|S)$. Is it possible to increase this probability further in addition to the maximization of the above mentioned difference? We look into this issue in the following.

Recall that, presumably, the data sets we are dealing with contain a large number of features. Thus it is quite likely that multiple features can reach the maximum value for $I(e;C) - \widehat{I}(e;S)$. A question is, among all these features, which one should we choose? It may seem that we should choose the one that maximizes $I(e;C)$. In the following, we will give plausible argument to show that this is not the case. We first look at an example to get some motivation.

Exmple: Consider the dataset in Table 1.

Table 1: An example data set

C	0	0	1	1
s	0	1	0	1
e_1	0	1	1	0
e_2	0	0	0	1
e_3	1	0	1	0

Suppose the current feature set is $\{s\}$. To select the next feature, note that $I(e_1;C) - I(e_1;s) = I(e_2;C) - I(e_2;s) = 0$ and $I(e_3;C) - I(e_3;s) = -1$. Thus e_1 and e_2 are the candidates. First, we have $I(e_1;C) = 0 < I(e_2;C) = 0.31$, which means e_2 is more relevant to C than e_1 is. This however only illustrates that e_2 offers more non-interacting information about C than e_1 does. On the other hand, we have $I(e_1;C|s) = 1 > I(e_2;C|s) = 0.5$. This means actually e_1 is a better choice. The implication is, e_1 offers more interacting information with s about C than e_2 does, and this interacting information plays a pre-dominant role in the above conditional mutual information. Now, let us see how s is correlated with e_1 and e_2. We have $H(e_1|s) = 1$ and $H(e_2|s) = 0.5$. This means that given s, e_2 is more certain than e_1. Intuitively, we can think of e_2 as having less freedom than e_1 given s. This point of 'conditional freedom' can be further illustrated, in part, by considering e_3. Here, we have $I(e_3;C|s) = 0$, implying that e_3 provides no additional information about C when s is given. Note that $I(e_1;C) = I(e_3;C)$, thus the non-interacting information by e_3 about C is not a factor for the discrepancy between $I(e_1;C|s)$ and $I(e_3;C|s)$. The only explanation for this discrepancy is that e_3 offers no interacting information with s about C. The cause for no interacting information is $H(e_3|s) = 0$, i.e., when s is given, e_3 is completely certain, and hence has no freedom at all.

In the following, we look at this issue from some theoretical perspective.

Assertion 3. *Let F be the full feature set and $S \subseteq F$. Let e_i and e_j be two features randomly chosen from $F - S$. Let $H_{i,j} = H(e_i|S) - H(e_j|S)$ and $HC_{i,j} = H(e_i|S,C) - H(e_j|S,C)$. Assume the following conditions hold true:*

1. *$Pr(H_{i,j} > 0) = Pr(H_{i,j} < 0) = 0.5$*[1]
2. *$Pr(HC_{i,j} > 0) = Pr(HC_{i,j} < 0) = 0.5$*
3. *for any $I \subseteq [-H(e_j), H(e_i)], Pr(HC_{i,j} \in I) > 0$*
4. *$H_{i,j}$ and $HC_{i,j}$ are independent.*

Then $Pr\big(I(e_i;C|S) > I(e_j;C|S)|H_{i,j} > 0\big)$
$> Pr\big(I(e_j;C|S) > I(e_i;C|S)|H_{i,j} > 0\big)$

Proof. We have:

$Pr(I(e_i;C|S) > I(e_j;C|S)|H_{i,j} > 0)$
$= Pr(H_{i,j} > HC_{i,j}|H_{i,j} > 0)$
$= Pr(HC_{i,j} < 0|H_{i,j} > 0) + Pr(H_{i,j} > HC_{i,j} > 0|H_{i,j} > 0)$
$= Pr(HC_{i,j} < 0) + Pr(H_{i,j} > HC_{i,j} > 0) > 0.5$

Since $Pr\big(I(e_i;C|S) > I(e_j;C|S)|H_{i,j} > 0\big)$ $+ Pr\big(I(e_j;C|S) > I(e_i;C|S)|H_{i,j} > 0\big) = 1$, the claim follows. □

Assertion 3 states that, under the specified conditions, features with higher remaining entropies when S is given are more likely to provide more additional information about C. Conditions 1 and 2 are reasonable since, without any prior knowledge about the distributions of the variables, and their correlations with S and C, we should not be biased toward the relative values of their entropies. Condition 3 states that $HC_{i,j}$ can possibly fall into any interval within its domain. Again, this is because we do not know the distributions of e_i and e_j, we cannot claim in a definitive term that $HC_{i,j}$ will not fall into certain interval in its domain. Condition 4, however, may seem a bit strong, since one may argue that knowing $H_{i,j} > 0$ increases the likelihood that $HC_{i,j} > 0$. We note, however, that $H_{i,j}$ involves only non-interacting information of S about e_i or e_j, while $HC_{i,j}$ involves interacting information of S and C about them, and the latter cannot be derived from the former.

The above discussion suggests that among the features e with the maximum difference of $I(e;C) - \widehat{I}(e;S)$, we should select the one that maximizes its conditional entropy given S. We will stress conditional entropy further, however, due to its intrinsic role in boosting the interacting information. In our algorithm we will expand the candidate set in which to apply conditional entropy, by requiring a feature to be only 'close' to, rather than attain the maximum difference of $I(e;C) - \widehat{I}(e;S)$, to be eligible for participating in the next

[1] We omit the probability that $H_{i,j} = 0$, since $H_{i,j}$ and $HC_{i,j}$ can be viewed as continuous variables.

round of conditional entropy filtering. We call the process of maximizing the conditional entropy *increasing likelihood for feature interacting.* Note that since $H(e|S)$ involves estimating multi-variate densities, we will approximate it as

$$\widehat{H}(e|S) = \frac{1}{|S|} \sum_{x \in S} H(e|x) \tag{9}$$

2.2.5 Algorithm

The algorithm is presented below, which is called Reinforced Mutual Information based Feature Selection (RMIFS).

Algorithm 1 RMIFS

```
1: $S \leftarrow \phi$
2: $optimal\_size \leftarrow |F|$
3: $current\_size \leftarrow 0$
   $\{min(*,*) = |F|$ if optimal unknown$\}$
4: while $|S| < |F|$ & $current\_size \le min(optimal\_size + W, |F|)$ do
5:     $max\_diff \leftarrow max(I(e;C) - \widehat{I}(e;S)|e \in F - S)$
6:     if $optimal\_size = |F|$ then
7:         if $max\_diff/(|S|+1) < \alpha H(C|S)$ or $H(C|S) = 0$ then
8:             $optimal\_size \leftarrow current\_size$
9:         end if
10:    end if
11:    if $max\_diff < 0$ then
12:        $max\_diff \leftarrow ((2-\beta)/\beta) \times max\_diff$
13:    end if
14:    $g \leftarrow argmax\{\widehat{H}(e|S)|e \in \{h|I(h;C) - \widehat{I}(h;S) \ge \beta \times max\_diff \& h \in F - S\}\}$
15:    append $g$ to the list
16:    $S \leftarrow S \bigcup \{g\}$
17:    $current\_size$++
18: end while
```

In the above algorithm, F is the full feature set, and S is the current feature set. The variable *optimal_size* is the size of the selected feature set deemed to be optimal. A full feature set size for *optimal_size* indicates the optimal feature set is unknown (i.e., yet to be determined). Variable *current_size* is the size of the feature set being examined in the current iteration, i.e., S. The test in the while loop implies that even when the optimal feature set has been selected, the algorithm will continue outputting up to W features, where W is a small value given by a user. This is because the optimal feature set serves only as a reference point, thus we output a few additional features to give a user some flexibilities. (See the experiments.) In line 7, if the first test evaluates to true, no remaining feature can substantially boost the relevance to C. Then the current feature set is identified as the optimal feature set. When the second test evaluates to true in line 7, all the remaining features meet the substantial boosting criterion. However, we have $I(S:C) = H(C)$, implying S has reached its full relevance to C. Thus, S is identified as the optimal feature set. Line 14 implements the process of increasing likelihood for feature interacting, where β controls the number of features which we allow not to reach the maximum difference of $I(e;C) - \widehat{I}(e;S)$ and to be scrutinized by the 'increasing likelihood for feature interacting' criterion. Our experiments show that the best value for it is between 0.9 and 0.95. The mutual information and the entropy are calculated in the manner described in the previous sections, and require only estimations of bi-variate densities. For the time complexity of the algorithm, let F be the full feature set, e_i be the feature selected in iteration i, and S_i be the set of features selected up to iteration i. Logically, selecting e_{i+1} in iteration $i+1$ requires evaluating $i+1$ mutual information for every feature $e \in F - S_i$, i.e., $I(e;C)$, and $I(e;x)$ for each $x \in S_i$. (Evaluating $\widehat{H}(e|S)$ does not take extra time, since $H(e|x)$ is already available in $I(e;x)$ for all $x \in S_i$.) However, observe that for every $x \in S_i - \{e_i\}$, $I(e;x)$ and $I(e;C)$ have already been generated in the previous iterations, and therefore can be stored for the later use. Thus, we need to evaluate $I(e;e_i)$ only. This results in an asymptotic time of $O((|S|+W)n)$ where S is the feature set identified as being optimal, W is the additional features output beyond S, and n is the total number of features. Since W is normally selected as a very small value, the time complexity is $O(|S|n)$.

3. EMPIRICAL STUDY

Our empirical analysis consists in comparison of our algorithm with four other algorithms, Correlation-based Feature Selection (CFS) [12], Fast Correlated-based Feature Selection (FCBF) [23], ReliefF [19] and Maximum Relevance Minimum Redundancy (MRMR) [9]. The former two are group-oriented, and the latter two are individual-oriented. The following is a brief description of them.

CFS associates each feature set with a metric proportional to the correlation between the feature set and the class, and inversely proportional to the inter-correlations among the features themselves. The correlation can be either Pearson correlation or symmetric uncertainty. CFS uses a best-first search strategy where the first feature set that attains the best metric is returned. FCBF uses approximate Markov-blanket (AMB) to filter out redundant features. A feature e_1 is an AMB for e_2 if the correlation between e_2 and C is lower than not only the correlation between e_1 and C, but also the correlation between e_2 and e_1. FCBF iteratively removes features that have an AMB. ReliefF assigns a discriminative score to each feature based on how well it discriminates a randomly selected instance from its nearest neighbors of different classes. MRMR assigns a score to each remaining feature that incorporates both redundancy and discriminative power. The redundancy and the discriminative power are measured by its mutual information with the features that are already selected, and with the class labeling, respectively. Since the CFS and FCBF are aimed at selecting the best feature sets, they are group-based, while the Relief and MRMR essentially rank all the features based on their scores, we consider them as individual-based[2].

We use three classifiers, NB (Naive Bayesian), ID3 (decision tree), and Logistic classifier, and five data sets, Leukemia, Colon Cancer, DLBCL (diffuse large b-cell lymphomas and follicular lymphomas), Prostate Tumor and Lung Cancer. The properties of the data sets are listed in Table 2.

We compare the classification accuracies and the sizes of the feature sets returned by the selection algorithms. The accuracies are generated by a LOOCV test. This is performed on the entire data set, i.e., training plus testing, for

[2] A more recent version of MRMR [19] uses an external classifier to identify the best feature set. This is similar to a wrapper approach. Therefore, we did not include it for comparison here.

Table 2: Descriptions of the Data Sets

name	#classes	#samples	#features	sources
Leukemia	2	72	7129	http://www.genome .wi.mit.edu/MPR
Colon Cancer	2	62	2000	http://www.molbio .princeton.edu/colondata
DLBCL	2	77	5470	http://www.tech .plym.ac.uk
Prostate Tumor	2	102	10510	http://www.tech .plym.ac.uk
Lung Cancer	5	203	12600	http://sdmc.i2r.a-star.edu.sg/rp/Main.html

each data set. (For datasets that do not provide testing sets, we perform LOOCV on the training sets.) Since all the algorithms are heuristics in nature, it makes practical sense to treat the feature sets returned by them as a reference, rather than a firm result to be followed. For this reason, we use a window with a small size, say 5, to identify a neighborhood of the feature set returned, and retrieve the highest accuracy attained by any feature set in that window. Since the CFS and FCBF do not output additional features once they have selected the feature set with the best metric value, we align the window's upper boundary to the last feature in the set, i.e, the neighborhood contains the preceding five subsets (inclusive) of the selected set formed by the algorithm. For RMIFS, since it can continue output features after it finds the one with the best metric value, we align the center of the window to the last feature in the set, i.e, the neighborhood contains the preceding three subsets (inclusive), and the following two supersets, of the selected set. For the two individual-based algorithms, since they do not return any single feature set, the above method is not applicable. We give them an advantage of a much larger window size of 32, i.e., the neighborhood contains the top 32 feature sets. As mentioned before, when $\alpha \in [0.05, 0.15]$ and $\beta \in [0.9, 0.95]$, the RMIFS has a stable performance for all the data sets in the experiments. As such, we set $\alpha = 0.1$ and $\beta = 0.93$. We run the other algorithms implemented in Weka V3.5. The results are shown in Table 3.

From the table, except for Colon Cancer dataset where FCBF has the best performance, RMIFS leads all the other algorithms in average accuracy in all the datasets. Although the improvement is not substantial, such a consistent trend is obvious. We attribute the small margin in the improvement over some of the other algorithms to their near saturate accuracies on the data sets. (For example, for Lung Cancer data set, which contains five classes, 95% of the accuracy should be very near to a global optimum for the classification accuracies.) In the case where their performances are relatively weak, RMIFS shows a clear advantage. For example, while most algorithms have relatively low accuracies for ID3 for a number of datasets, RMIFS still demonstrates a robust behavior for ID3 in all the data sets. The effectiveness of the mechanisms used in our algorithm can also be explained from the comparison with MRMR. Both RMIFS and MRMR are based on mutual information. RMIFS however enjoys consistently higher accuracies than MRMR in all data sets, despite the fact that the latter has been given a selection space five times larger. The largest improvement over MRMR occurs in Colon Cancer data set. This is not a coincidence. It is well known that Colon Cancer dataset presents some challenges in feature selection due to the fact that it contains some hard-to-recognize genes that may confuse a learning process. (One of these kinds of genes is responsible for cell composition, which may appear to be good indicators for the cancerous and normal samples, but in fact are not informative [2]. This is because the cancerous tissue generally contains many epithelial (skin) cells, while the normal tissue contains different kinds, including smooth muscle cells.) The above results suggest that the advantages of our mechanisms are more obvious in the cases where the existing algorithms are weak. We attribute this point mainly to the use of 'increasing likelihood for feature interacting' mechanism in our algorithm, which aims at increasing the probability that the selected genes have high interacting information about the class labeling.

Now, consider the sizes of the feature sets determined by the three group-based algorithms. Table 3 shows that RMIFS consistently selects smaller feature sets than CFS and FCBF on all the datasets by a big margin. The discrepancies are especially noticeable for Lung Cancer dataset, which contains five class labels. In this case, we checked output data by the three algorithms in detail. For CFS and FCBF, we found that the same accuracies were also attained at some subsets with far smaller sizes that were scrutinized at some earlier stages. On the other hand, for RMIFS, none of the nine proper subsets of the selected set processed at the earlier stages can reach the same accuracy. Recall that in our 'substantial relevance boosting' mechanism, a gene will be selected only if it contains a substantial amount of additional information about the class labeling given the current feature set. The above result shows that this mechanism indeed has a strong capability of filtering out redundant and/or noisy features, even in the case where multiple class labels are present.

Finally, it is worth mentioning that the feature sets by the individual-based algorithms listed in the table have smaller sizes than the CFS and FCBF in all cases, and than RMIFS in almost half of the cases. This is not entirely unexpected, given that these feature sets are manually selected to be the smallest ones with the highest accuracies from the top 32 feature sets, while the feature sets for the group-based algorithms are determined by the programs based only on the metric values.

4. CONCLUSION

Due to the insufficient representation for the true distribution of the data population by microarray data sets, which possess extremely high dimensions and only a small number of training cases, feature selection has proven to be a necessary step in pre-processing the data set for further data analysis tasks such as classifications. We study for this kind of data sets the issue of feature selections based on a popular framework of mutual information, and propose methods to enhance its effectiveness by reinforcing the existing criteria. Our main strategy consists of substantial relevance boosting and increasing likelihood for feature interacting. The former requires the common maximization strategy to be

Table 3: Experimental Results

		CFS	FCBF	RMIFS	ReliefF	MRMR
Leukemia	**Size**	81	51	4	3	2
	NB	100%	100%	100%	100%	100%
	ID3	90.27%	88.89%	100%	97.22%	100%
	Logistic	100%	100%	100%	100%	100%
	Avg	96.75%	96.30%	**100%**	99.07%	**100%**
Colon Cancer	**Size**	26	14	6	7	4
	NB	98.39%	100%	96.77%	88.71%	95.16%
	ID3	93.55%	95.16%	95.16%	90.32%	93.55%
	Logistic	96.77%	100%	100%	88.71%	95.16%
	Avg	96.23%	98.38%	97.31%	89.25%	94.62%
DLBCL	**Size**	87	65	6	14	17
	NB	100%	100%	100%	97.40%	100%
	ID3	96.10%	93.51%	98.70%	90.91%	94.81%
	Logistic	97.40%	98.70%	100%	98.70%	100%
	Avg	97.83%	97.40%	99.57%	95.67%	98.27%
Prostate Tumor	**Size**	79	67	10	5	11
	NB	98.04%	98.04%	98.04%	93.14%	98.04%
	ID3	88.24%	85.29%	96.08%	93.14%	97.06%
	Logistic	98.04%	99.02%	99.02%	93.14%	97.06%
	Avg	94.77%	94.12%	97.71%	93.14%	97.39%
Lung Cancer	**Size**	550	453	10	23	26
	NB	99.01%	99.01%	98.52%	95.07%	98.52%
	ID3	91.13%	88.67%	93.60%	88.18%	90.15%
	Logistic	96.06%	95.07%	95.57%	93.10%	97.54%
	Avg	95.40%	94.25%	95.89%	92.12%	95.40%

strengthened to cope with the distribution deviation, and the latter calls for additional feature scrutiny based on the concept of conditional entropy to increase the probability of increasing interacting information. Our experiments show the effectiveness of these strategies.

5. ACKNOWLEDGEMENTS

Shuigeng Zhou, Feng Li and Kai Jiang were supported by the National Basic Research Program of China under grant No. 2007CB310806, the National Natural Science Foundation of China under grant Nos. 60573183 and 60773123, and Shanghai Leading Academic Discipline Project No. B114.

6. REFERENCES

[1] H. Almuallim and T. Dietterich. Learning boolean concepts in the presence of many irrelevant features. *Artificial Intelligence*, 69(1-2):279–305, 1994.

[2] U. Alon. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. of the National Academy of Sciences of the United States of America*, 96(12):6745–6750, 1999.

[3] J. S. Q. Y. B. Cao, D. Shen and Z. Chen. Feature selection in a kernel space. *Proc. of the 24th ICML*, 227:121–128, 2007.

[4] R. Battiti. Using mutual information for selecting features in supervised neural net learning. *IEEE Trans. On Neural Networks*, 5(4):537–550, 1994.

[5] D. Bell and H. Wang. A formalism for relevance and its application in feature subset selection. *Machine Learning*, 41(2):175–195, 2000.

[6] A. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, pages 245–271, 1997.

[7] H. Cho. Nonlinear feature extraction and classification of multivariate process data in kernel feature space. *Expert Systems with Applications*, 32:534–542, 2007.

[8] T. Cover and J. Thomas. *Elements of information theory.* John Wiley, 2006.

[9] C. Ding and H. Peng. Minimum redundancy feature selection from microarray gene expression data. *Comp. Syts. Bioinfor. Conf.*, pages 523–52, 2003.

[10] F. Fleuret. Fast binary feature selection with conditional mutual information. *Journal of Machine Learning Research*, 5:1531–1555, 2004.

[11] F. L. H. Peng and C. Ding. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 27(8):1226–1238, 2005.

[12] M. Hall. Correlation-based feature selection for discrete and numeric class machine learning. *Proc. of the 17th ICML*, pages 359 – 366, 2000.

[13] A. Jakulin and I. Bratko. Analyzing attribute dependencies. *PAKDD*, pages 229–240, 2003.

[14] A. Jakulin and I. Bratko. Testing the significance of attribute interactions. *Proc. of the 21st ICML*, 69:52, 2004.

[15] D. Koller and M. Sahami. Toward optimal feature selection. *Proc. of the 13rd ICML*, pages 284–292, 1996.

[16] P. Meyer and G. Bontempi. On the use of variable complementarity for feature selection in cancer

classification. Technical report.

[17] H. Peng and F. Long. An efficient max-dependency algorithm for gene selection. *36th Symposium on the interface: Computational Biology and Bioinformatics*, 2004.

[18] A. N. R. Gilad-Bachrach and N. Tishby. Margin based feature selection - theory and algorithms. *Proc. of the 21st ICML*, 69:43, 2004.

[19] M. Sikonja and I. Kononenko. Theoretical and empirical analysis of relief and relieff. *Machine Learning*, 53:23–69, 2003.

[20] Y. Sun and J. Li. Iterative relief for feature weighting. *Proc. of the 23rd ICML*, 148:913–920, 2006.

[21] J. B. W. Duch, T. Winiarsi and A. Kachel. Feature selection and ranking filters. *Intl. Conf. on Artifical Neural Networks*, pages 251–254, 2003.

[22] A. Wang and E. Gehan. Gene selection for microarray data analysis using principle component analysis. *Statistics in Medicine*, 24(1313):2069–2087, 2005.

[23] L. Yu and H. Liu. Efficient feature selection via analysis of relevance and redundancy. *Journal of Machine Learning Research*, 5:1205–1224, 2004.

[24] X. Zhou and K. Mao. Ls bound based gene selection for dna microarray data. *Bioinformatics*, 21(8):1559–1564, 2005.

Graph-based Temporal Mining of Metabolic Pathways with Microarray Data

Chang hun You, Lawrence B. Holder, Diane J. Cook
School of Electrical Engineering & Computer Science
Washington State University
Box 642752, Pullman, WA 99164-2752
{changhun, holder, cook}@eecs.wsu.edu

ABSTRACT

We propose a dynamic graph-based relational learning approach using graph-rewriting rules to analyze how biological networks change over time. The analysis of dynamic biological networks is necessary to understand life at the system-level, because biological networks continuously change their structures and properties while an organism performs various biological activities to promote reproduction and sustain our lives. Most current graph-based data mining approaches overlook dynamic features of biological networks, because they are focused on only static graphs. First, we generate a dynamic graph, which is a sequence of graphs representing biological networks changing over time. Then, our approach discovers graph rewriting rules, which show how to replace subgraphs, between two sequential graphs. These rewriting rules describe the structural difference between two graphs, and describe how the graphs in the dynamic graph change over time. Temporal relational patterns discovered in dynamic graphs representing synthetic networks and metabolic pathways show that our approach enables the discovery of dynamic patterns in biological networks.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning; J.3 [**Life and Medical Science**]: Biology and genetics—*Biological Networks*

Keywords

Temporal Graph Mining, Graph Rewriting Rules, Biological Network

1. INTRODUCTION

To investigate bio-organisms and understand the theory of life, we should consider our bodies are dynamic. Our bodies are well-organized and vigorous systems, which promote reproduction and sustain our lives. Well-organized systems refer to structural properties of biological networks, which include various molecules and relationships between molecules. Vigorous systems refer to dynamic properties of biological networks, which continuously change their structures and properties, while an organism performs various biological activities, such as digestion, respiration and so on. We assume the structures of biological networks change over time as they interact with specific conditions, for instance, a disease.

We propose a novel approach to analyze structural features along with temporal features in a time series of biological networks to enhance our systems-level understanding of bio-organisms. The temporal patterns in the structural changes of biological networks can be significant information about a disease and help researchers develop new drugs. During the development period, the temporal patterns in the structural changes of biological networks after taking the medicine are also used for the development and evaluation of the new drug. Lactose intolerance is the inability to digest lactose because of a lack of the lactase enzyme, breaking down lactose into galactose and glucose [3]. Two major treatments are to minimize the intake of lactose products and take the lactase supplement. Our approach can help us discover the temporal patterns in the structural changes of galactose metabolism pathway after these treatments, and investigate another treatment (i.e., improving the production of the lactase enzyme in the pathway).

Temporal data mining can discover temporal features in the sequence of data. But it is hard for temporal data mining to discover structural features or relational patterns between two entities. Graph-based data mining is a process to learn novel knowledge in data represented as a graph and has been applied to identify relational patterns in biological networks [24]. However, the current graph-based data mining approaches overlook dynamic features of networks, because most of them are focused on only static graphs. Our dynamic graph-based relational learning approach uses graph-rewriting rules to analyze how biological networks change over time. Graph-rewriting rules define how one graph changes to another in its topology replacing vertices, edges or subgraphs according to the rewriting rules. Our discovery algorithm takes a dynamic graph as an input. The dynamic graph contains a sequence of graphs representing biological networks changing over time. Then, the algorithm discovers rewriting rules between two sequential graphs. After discovery of whole sets of graph rewriting rules from the dynamic graph, we discover temporal patterns in the discovered graph rewriting rules.

This paper, first, introduces several preceding approaches

BIOKDD'08, August 24, 2008, Las Vegas, Nevada, USA

related to dynamic analysis of biological networks. Then, we present our definition of graph rewriting rules and our Dynamic Graph Relational Learning (DynGRL) algorithm. In our experiments, we generate several dynamic graphs of the yeast metabolic pathways using the KEGG PATHWAY database and microarray data. Then, we apply our DynGRL approach to the dynamic graphs. The results show our discovered graph rewriting rules and temporal patterns in the rewriting rules. The temporal patterns show which graph rewriting rules are repeated periodically or temporal relations among several graph rewriting rules. Our results also help us to visualize what substructures change over time and how they change. This approach enables us to investigate dynamic patterns in biological networks in two aspects: structural and temporal explorations. The ultimate goal of this research is to discover the temporal patterns in the structural changes of biological networks for drug discovery and the systems-level understanding of complex biosystems.

2. RELATED WORK

To understand how biosystems change over time, we need to follow two aspects: structural and temporal analysis of dynamic biological networks. Here, we introduce microarray analysis and temporal data mining for the temporal exploration. Then, related research on biological networks is followed for the structural exploration.

The microarray is a tool for measuring gene expression levels for thousands of genes at the same time [4, 17], and have already produced terabytes of important functional genomics data that can provide clues about how genes and gene products interact and form their gene interaction networks. Most genes are co-expressed, as most proteins interact with other molecules. Co-expressed genes construct common processes or patterns in biological networks (gene regulatory networks or protein networks) in the specific condition or over time. Microarrays can also monitor patterns in gene expression levels for the period of time or at the different conditions. Patterns in gene expression levels can represent changes in the biological status or distinguish two different states, such as the normal and disease state.

Some microarray research [7, 22] describes patterns in gene expression values. One approach explores temporal patterns in gene expression promoting the regulation of a metabolic pathway [7]. Other research observes more than half of the yeast genes show periodic temporal patterns during metabolic cycles [22]. But the microarray analysis can overlook structural aspects, which show how the genes or expressed gene products are related to each other in biological networks.

Temporal data mining attempts to learn temporal patterns in sequential data, which is ordered with respect to some index like time stamps, rather than static data [20]. Temporal data mining is focused on discovery of relational aspects in data such as discovery of temporal relations or cause-effect association. In other words, we can understand how or why the object changes rather than merely static properties of the object. In this research, we are focused on discovery of temporal patterns and their visualization. Allen and et al. [2] formalized temporal logic for time intervals using 13 interval relations. This approach allows us to present temporal relations in sequential data.

There are several approaches to apply temporal data mining in biological data. Ho et al. [11] propose an approach to detect temporal patterns and relations between medical events of Hepatitis data. They represent medical information of patients as sequential events and classify temporal patterns and relations of medical testing results in the sequential events using the Naive Bayes classifier. Farach-Colton et al. [9] introduce an approach of mining temporal relations in protein-protein interactions. They model the assembly pathways of Ribosome using protein-protein interactions. This approach determines the order of molecular connections using the distance measure of each interaction between two proteins.

Temporal data mining approaches discover temporal patterns in data, but they disregard relational aspects among entities. For example, they can identify temporal patterns of appearance of genes such that a gene, YBR218C, appears before another gene, YGL062W, but cannot identify how these two genes interact with each other.

According to the central dogma in molecular biology, the genetic information in DNA is transcribed into RNA (transcription) and protein is synthesized from RNA (translation). These biomolecules (DNA, RNA and proteins) play central roles in the aspects of the function and structure of organisms. However, there are few molecules that can work alone. Each molecule has its own properties and relationships with other molecules to carry out its function. Biological networks have various molecules and relations between them including reactions and relations among genes and proteins. Biological networks including metabolic pathways, protein-protein interactions and gene regulatory networks, consist of various molecules and their relationships [13]. In addition to the structural aspect, we also consider the temporal aspect of biological networks, because the biosystems always change their properties and structures while interacting with other conditions.

Two approaches have been developed for the analysis of biological networks. One approach is graph-based data mining [14, 24]. This approach represents biological networks as graphs, where vertices represent molecules and edges represent relations between molecules, and discovers frequent patterns in graphs. Many approaches of graph-based data mining discover structural features of biological networks, but they overlook temporal properties. The other approach is mathematical modeling, which is an abstract model to describe a system using mathematical formulae [18]. Most of these approaches, as a type of quantitative analysis, model the kinetics of pathways and analyzes the trends in the amounts of molecules and the flux of biochemical reactions. But most of them disregard relations among multiple molecules.

There are two main points to consider for understanding biological networks: structural and temporal aspects. The former reminds us to focus on relations between molecules as well as a single molecule. The latter is necessary to understand biological networks as dynamic operations rather than static relations, because every biological process changes over time and interacts with inner or outer conditions. For this reason, we need an approach to analyze biological networks changing over time in both aspects: structural and temporal properties.

3. GRAPH REWRITING RULES

This paper focuses on temporal and structural analysis of biological networks. Our dynamic graph-based relational learning approach discovers graph rewriting rules in a series

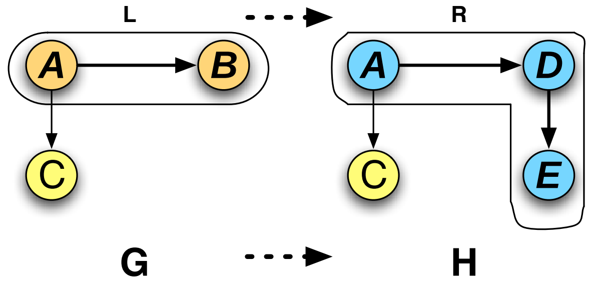


Figure 1: An example of application of graph rewriting rules, where the rule derives a graph H from a graph G by replacing a subgraph L by a subgraph R.

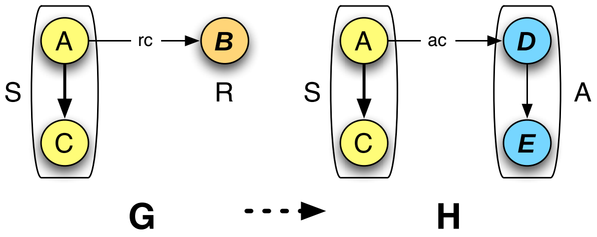


Figure 2: An example of application of graph rewriting rules, which shows an removal rule {R, rc} from a graph G and an addition rule {A,ac} to a graph H. The removal and addition substructures are connected to G and H by edges rc and ac. S represents the common subgraph between G and H.

of graphs changing their structures over time. Each graph rewriting rule represents topological changes between two sequential graphs. Here, we define graph rewriting rules for our approach.

Graph rewriting is a method to represent topological changes of graphs using graph rewriting rules [8, 21]. Generally, graph rewriting rules identify subgraphs in a graph and modify them. Each graph rewriting rule defines a transformation between L and R, where L and R are subgraphs in two graphs G and H respectively, such that L is replaced by R, L is deleted, or R is created [19]. As shown in figure 1, L is identified first in graph G. Then L is replaced by R to produce graph H. There are also several algorithms to discover the node or edge replacement graph grammar using the minimum description length principle [12, 15]. However, their scope is limited to static graphs.

Traditional approaches to the identification of graph rewriting rules determine which subgraphs will be replaced by other subgraphs. Our approach is focused on representing changing structures between two graphs rather than just what subgraphs change. We define our graph rewriting rules to represent how substructures change between two graphs rather than just what subgraphs change. First, we discover maximum common subgraphs between two sequential graphs G_1 and G_2. Then, we derive removal substructures from G_1 and addition substructures from G_2. Figure 2 shows an instance of this process. A maximum common subgraph (denoted by S) is discovered between two graphs, G_1 and G_2. Then the remaining structure in G_1 and G_2 becomes removal (denoted by R) and addition (denoted by A) substructures respectively. These substructures with connection edges rc and ac are elements of graph rewriting rules: removal and addition rules respectively. For this approach, we define several preliminary terms.

A directed graph G is defined as $G = (V, E)$, where V is a set of vertices and E is a set of edges. An edge e ($\in E$) is directed from x to y as $e = (x, y)$, where $x, y \in V$. Here, we define a dynamic graph DG as a sequence of n graphs as $DG = \{G_1, G_2, \cdots, G_n\}$, where each graph G_i is a graph at time i for $1 \leq i \leq n$. Then, we define a set of removal substructures RG and a set of addition substructures AG as follows.

$$RG_i = G_i/S_{i,i+1}, \ AG_{i+1} = G_{i+1}/S_{i,i+1}$$

RG_i denotes a set of removal substructures in a graph G_i, AG_{i+1} denotes a set of addition substructures in the next graph G_{i+1}, and $S_{i,i+1}$ is a maximum set of common subgraphs between two sequential graphs G_i and G_{i+1} in a dynamic graph DG.

A prior graph G_i is transformed to a posterior graph G_{i+1} by application of a set of graph rewriting rules $GR_{i,i+1}$ as denoted by

$$G_{i+1} = G_i \bigoplus GR_{i,i+1}$$

A set of graph rewriting rules $GR_{i,i+1}$ between two sequential graphs G_i and G_{i+1} is defined as follows.

$$\begin{aligned} GR_{i,i+1} &= \{(m, p, CE_m, CL_m), \cdots, \\ &\quad (n, q, CE_n, CL_n), , \cdots, \} \end{aligned}$$

m and n are indices of graph rewriting rules in a set $GR_{i,i+1}$. p and q are indices of a removal substructure in RG_i and an addition substructure in AG_{i+1} respectively. CE and CL are defined as a set of connection edges and a set of labels of the connection edges. Each element of RG and AG corresponds to a set of CE and CL, unless a removal (addition) substructure does not connect to the G_i (G_{i+1}). CE_k and CL_k represent connections between substructures and the original graphs ($k = m$ or n) as follows.

$$CE = \{(d, X, Y), \cdots\}, \ \ CL = \{label_{xy}, \cdots\}$$

d represents whether the edge is directed or undirected using d and u. X and Y denote the starting and ending vertices of the edge. Because the connection edge links the substructure to the original graph, one end of this edge is from the substructure and the other is from the original graph. The end vertex from the substructure starts with “s” followed by the index of the vertex, and the end vertex from the original graph starts with “g” followed by the index of the vertex. For example, $(d, g1, s3)$ represents the directed edge from a vertex 1 in the original graph to another vertex 3 in the substructure. $label_{xy}$ represents a label for the corresponding connection edge between two vertices X and Y. The number of elements of CE (CL as well) represents the number of connections between substructures and the original graph. If a substructure is not connected to the original graph, both sets of CE and CL are empty.

We describe more detail with an example. Figure 3 shows an instance of graph rewriting rules between the synthetic biological networks, G_1 and G_2. The thick-drawn substruc-

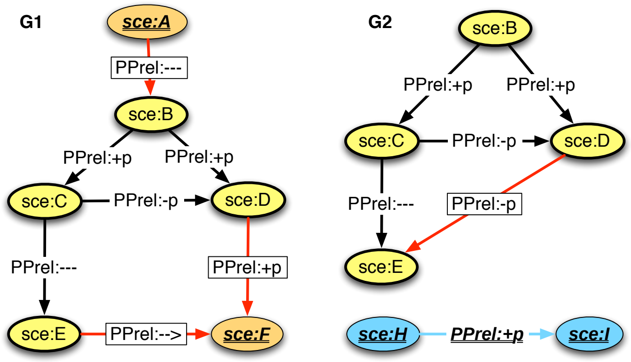


Figure 3: An instance of graph rewriting rules between graph G1 and G2 in the synthetic biological networks

tures in both graphs represent the maximum common substructures. The underline labeled elements in G_1 represent removal substructures (from G_1) with the rectangle labeled connection edges. The underline labeled elements in G_2 represent addition substructures (to G_2), where this addition rule does not have any connection edges.

$GR_{1,2}$ represents a set of graph rewriting rules, which is applied to G_1 and produces G_2 using $G_2 = G_1 \bigoplus GR_{1,2}$ as described in the previous section. It has four graph rewriting rules. For example, r_1 (r denotes removal.) represents an index into the set of removal rules including a removal subgraph ($rSub_1$), which contains a single vertex A. $rSub_1$ was connected by an edge $(d, s1, g2)$, which is labeled by $PPrel : - - -$. This edge is a directed edge (indicated by 'd'). One end of this edge is $s1$, which denotes a vertex number 1 in $rSub_1$ (s denotes the substructure.). The other end is $g2$, which denotes a vertex number 2 in G_1 (g denotes the original graph.). a_1 and a_2 represent addition rules similarly. But these two cases look somewhat different. a_1 has $\emptyset$ (emptyset) as the addition substructure, because a_1 is a rule representing a blue edge $PPrel : -p$ in G_2 without any addition substructure. a_2 also has $\emptyset$s for edges and edge labels, because $aSub_1$ represents a disconnected graph including vertices H and I in G_2.

$$\begin{aligned} GR_{1,2} \quad = \quad & \{(r_1, rSub_1, \{(d, s1, g2)\}, \{PPrel : - - -\}), \\ & (r_2, rSub_2, \{(d, g4, s1), (d, g5, s1)\}, \\ & \{PPrel : +p, PPrel : - - >\}), \\ & (a_1, \emptyset, \{(d, g3, g4)\}, \{PPrel : -p\}), \\ & (a_2, aSub_1, \emptyset, \emptyset)\} \end{aligned}$$

The graph rewriting rules show how two sequential graphs are structurally different. After collecting all sets of graph rewriting rules in a dynamic graph, we also discover temporal patterns in graph rewriting rules, which can describe how the graphs change over time as well as what structures change.

4. APPROACH

This section describes our graph rewriting rule discovery system, DynGRL, that discovers graph rewriting rules in a dynamic graph. Our approach extends Cook and Holder's earlier work [5, 6], which is a graph-based relational learning approach to discover subgraphs. Their approach evaluates discovered subgraphs using the Minimum Description Length (MDL) principle to find the best subgraphs that minimize the description length of the input graph after being compressed by the subgraphs. The description length of the substructure S is represented by $DL(S)$, the description length of the input graph is $DL(G)$, and the description length of the input graph after compression is $DL(G|S)$. The approach tries to minimize the $Compression$ of the graph as follows.

$$Compression = \frac{DL(S) + DL(G|S)}{DL(G)}$$

Their approach, which is called as $DiscoverSub()$ in our algorithms, tries to maximize the $Value$ of the subgraph, which is simply the inverse of the $Compression$. Even though we can use a frequent subgraph mining approach [16, 23] for $DiscoverSub()$, we choose the compression-based approach, because there is no need to choose a proper minimum support and many times the best-compressing subgraph better captures the patterns of interest than the most frequent subgraph. A more detailed comparison between the two approaches is left for future work.

The algorithm starts with a dynamic graph DG consisting of a sequence of n graphs as shown in algorithm 1. First, the algorithm creates a list of n virtual graphs, VGL, corresponding to n time series of graphs at line 1. Our approach uses a virtual graph to specify the application locations of graph rewriting rules. Because a graph may have multiple graph rewriting rules and several same-labeled vertices and edges, the exact locations of connection edges and rewriting rules are important to reduce the discovery error. The next procedure is to create a two-graph set, $Graphs$, including two sequential graphs G_i and G_{i+1} (line 5) and to specify the $limit$ based on unique labeled vertices and edges of G_i and G_{i+1} (line 6). UVL and UEL denote the number of unique vertex labels and edges in G_i and G_{i+1}. The $Limit$ specifies the number of substructures to consider when searching for a common substructure (line 6). The $Limit$ based on the number of labels in the input graph bounds the search space within polynomial time and ensure consideration of most of the possible substructures.

The inner loop (lines 7 to 14) represents the procedure to discover common substructures between two sequential graphs. $DiscoverSub()$ is used to find the maximum common subgraph. Although to find the maximum common subgraph is NP-Complete, $DiscoverSub()$ can be used as a polynomial-time approximation to this problem using $Limit$ and $iteration$ as described later. After discovery of the best substructure, the algorithm checks whether the substructure is a subgraph of both graphs G_i and G_{i+1}. In the affirmative case, the best substructure is added into $ComSubSet$ and the two target graphs are compressed by replacing the substructure with a vertex. If the best substructure does not belong to one of the two graphs, the algorithm just compresses the graphs without adding any entry into $ComSubSet$. After compression, the algorithm discovers another substructure at the next iteration until there is no more compression.

Using the complete list of common substructures, $ComSubSet$, the algorithm acquires removal substructures, $remSubs$, and addition substructures, $addSubs$, (lines 15 and 17). First, the algorithm identifies vertices and edges not part of common substructures and finds each disconnected substructure in G_i and G_{i+1} using the modified Breadth First Search

```
Algorithm 1 DynGRL discovery Algorithm
Require: $DG = \{G_1, G_2, \cdots, G_n\}$
 1. Create $VGL = \{VG_1, VG_2, \cdots, VG_n\}$
 2. $RRL = \{\}$
 3. for $i = 1$ to $n - 1$ do
 4.   $RemRuleSet$ =AddRuleSet $= ComSubSet = \{\}$
 5.   $Graphs = \{G_i, G_{i+1}\}$
 6.   $Limit = UVL + 4(UEL - 1)$
 7.   while No more compression do
 8.     $BestSub$ = DiscoverSub($Limit, Graphs$)
 9.     if $BestSub \in G_i$ & $G_{i+1}$ then
10.       Add $BestSub$ into $ComSet$
11.     end if
12.     Compress $Graphs$ by $BestSub$
13.     Mark $BestSub$ on $VG_i$ and $VG_{i+1}$
14.   end while
15.   Get $remSubs$, $CE$ from $VG_i$
16.   Add $remSubs$ into $RemSubSet$ and $CE$ into
      $RemCESet$
17.   Get $addSubs$, $CE$ from $VG_{i+1}$
18.   Add $addSubs$ into $AddSubSet$ and $CE$ into
      $AddCESet$
19.   Create $RR$ from $RemSubSet$, $AddSubSet$,
      $RemCESet$, $AddCESet$
20.   Add $RR$ into $RRL$
21. end for
22. return $RRL$
```

(mBFS), which adds each edge as well as each vertex into the queues as visited or to be visited. The marked substructures in G_i and G_{i+1} are removal and addition substructures respectively. While mBFS searches these removal and addition substructures, it also finds connection edges, CE, as described previously. These edges are added into $RemCESet$ and $AddCESet$, where removal and addition substructures are added into $RemSubSet$ and $AddSubSet$ respectively (in lines 16 and 18). Using these rewriting substructures and connection edges, rewriting rules (RR) are created and stored into RRL (in lines 19 to 20).

The main challenge of our algorithm is to discover maximum common subgraphs between two sequential graphs, because this problem is known to be NP-hard [10]. To avoid this problem, first we use the $Limit$ to restrict the number of substructures to consider in each iteration. The $Limit$ is computed using the number of unique labels of vertices and edges in graphs. Second, our algorithm does not try to discover the whole common substructures at once. In each step, the algorithm discovers a portion of common, connected substructure and iterates the discovery process until discovering the whole maximum common subgraphs. Usually, the size of graphs representing biological networks is not too large. Therefore, discovery of graph rewriting rules is still feasible. However, we still have challenges to analyze very large graphs.

5. DATASETS: MICROARRAY DATA AND GRAPH

We prepare dynamic graphs representing the yeast metabolic pathways in combination with microarray data. As described in section 2, microarrays can be used in two ways: monitoring the change of gene expression levels over time or distinguishing patterns in two different states. Here, we use time-based microarray data to generate a dynamic graph, where each column of data represents the gene expression values at a particular time. The microarray data used in our research observes periodic gene expression of *Saccharomyces cerevisiae* using microarray analysis [22]. The microarray data has 36 columns where each column represents one time slice. Their results show more than 50% of genes have three periodic cycles in the gene expression. We normalize each gene expression value of microarray data from 0 to 1, because we are focused on trends of the changes of gene expression values. Figure 4 shows normalized gene expression values of three genes shown in the glycolysis pathway.

Here, we prepare 10 dynamic graphs, each of which contains 36 consecutive graphs representing one yeast metabolic pathway changing over time (36 time slices) corresponding to 36 columns in microarray data. The 10 dynamic graphs represent 10 metabolic pathways: glycolysis (00010), TCA (00020), Pentose phosphate pathway (00030), Purine metabolism (00230), Pyrimidine metabolism (00240), Urea cycle (00220), Glutamate metabolism (00251), Arginine and proline metabolism (00330), Glycerolipid metabolism (00561) and Glycerophospholipid metabolism (00564), where each number denotes the identification number of the pathways in the KEGG data [1]. The first three pathways are involved in the carbohydrate metabolism, the second two pathways are involved in the nucleic acids, the next three pathways are involved in the amino acids metabolism and the last two

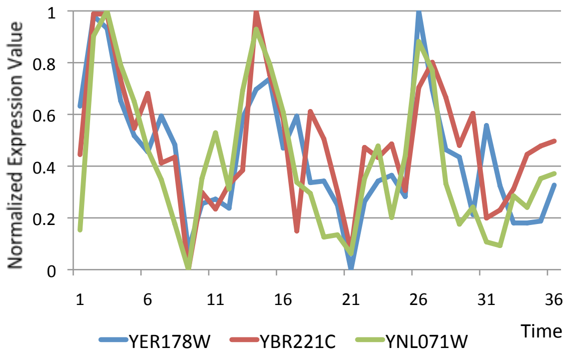


Figure 4: The oscillation curves of the changing gene expression values of three yeast genes: YNL071W, YER178W, and YBR221C.

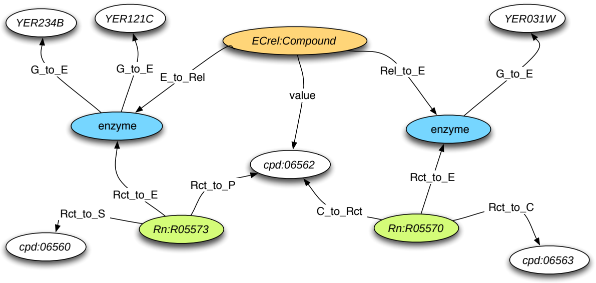


Figure 5: An instance of the graph representation for a metabolic pathway.

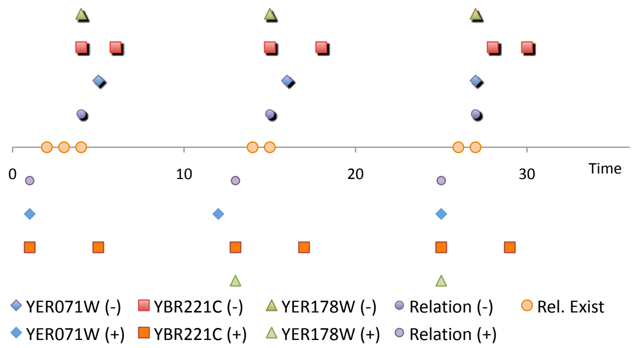


Figure 6: A visualization of time points when the substructure including each gene is removed from or added to graphs representing the glycolysis pathway at the experiment of threshold 0.6. The points above the time axis represent the time points when the substructures including the specified genes or relation are removed (Genes with (-)). The points below the time axis represent the time points when the substructures including the specified genes or relation are added (Genes with (+)). Relation points represent the time points when the enzyme-enzyme relations are shown in the pathway.

pathways are involved in the lipid metabolism.

First, we generate a static graph to represent each metabolic pathway from the KEGG PATHWAY database [1], where vertices represent compounds, genes, enzymes, relations and reactions, and edges represent relationships between vertices. Figure 5 shows an example of the graph representation. "ECrel:Compound" represents a relation between two enzymes (gene products). One enzyme is produced by one or more genes, which is represented as edges "G_to_E". "RN:R$xxxxx$" represents a reaction and "cpd:C$yyyyy$" represents chemical compounds, where $xxxxx$ and $yyyyy$ represent the identification number in the KEGG database. Here, we assume only genes change over time based on gene expression values and other molecules like compounds remain the same amount.

We use a threshold t to apply the numeric gene expression values on graph. At each time, we assume a gene, which has more than t gene expression value, is shown in the graph. One particular point is our graph representation has enzyme vertices, which do not exist in the KEGG data. One enzyme needs one or more genes to synthesize. At a specific time, only one gene can be expressed out of two genes, which are needed for one enzyme. Naturally, the enzyme is not synthesized at that time. We use enzyme vertices to represent this scheme. Only when all genes are expressed, the enzyme vertex is shown in the graph. At that time, the reaction, which is catalyzed by the enzyme, is also shown. In this way, we can observe the structure of the glycolysis pathway based on microarray gene expression at each time.

6. EXPERIMENTS AND RESULTS

Our approach discovers graph rewriting rules in each dynamic graph. First, we discuss temporal patterns in graph rewriting rules. Then, we represent how the discovered substructures in the rewriting rules link to the original graphs at the specific time.

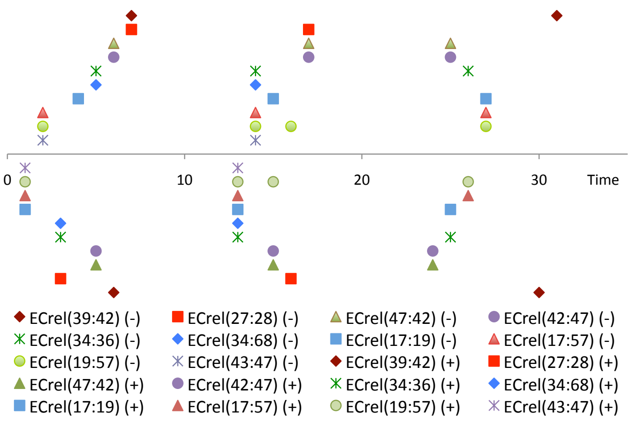


Figure 7: A visualization of time points when a particular substructure is removed from or added to graphs representing the glycolysis pathway at the experiment of threshold 0.6. Each substructure includes a relation, which is an enzyme-enzyme relation between two gene, where $ECrel(x, y)$ represents the relation, and x, y represent the id of enzymes.

6.1 Temporal patterns

As described in the previous section, the goal of this research is to discover temporal patterns in graph rewriting rules to describe structural changes of metabolic pathways over time. Because the result of the microarray data [22] represents three periodic cycles of gene expression, we observe similar temporal patterns in graph rewriting rules. Here, we are focused on graph rewriting rules involving enzyme-enzyme relations as well as genes. The enzyme-enzyme relation represents a relationship between two enzymes. As shown in figure 5, one or more genes produce an enzyme, and the enzyme can have a relation with one other enzyme. The relation vertex labeled as "ECrel:Compound" exists, only when there exist two enzyme vertices. Each enzyme vertex exists only when the linked genes exist (biologically, the linked genes produce the enzyme). The left enzyme exists only when two genes, YER178W and YER221C exist. The right enzyme exists only when one gene YAL038W exists.

Figure 6 shows a visualization of the changes to the partial pathway including the above three genes of the glycolysis pathway. The complete pathway is shown in figure 10 (Sub F). The points above the time axis represent the time points when the substructures including the specified genes or relation are removed. The points below the time axis represent the time points when the substructures including the specified genes or relation are added. The points on the axis represent the time when the relation exists. The result clearly shows the temporal patterns in removal and addition rules as three cycles. Three genes are added and the relation is shown in the pathway. After several time intervals, one of three genes starts to be removed from the pathway and the relation disappears, too. Like the microarray research [22], we can notice the genes are added and removed three times periodically. In addition, we discover the removal and

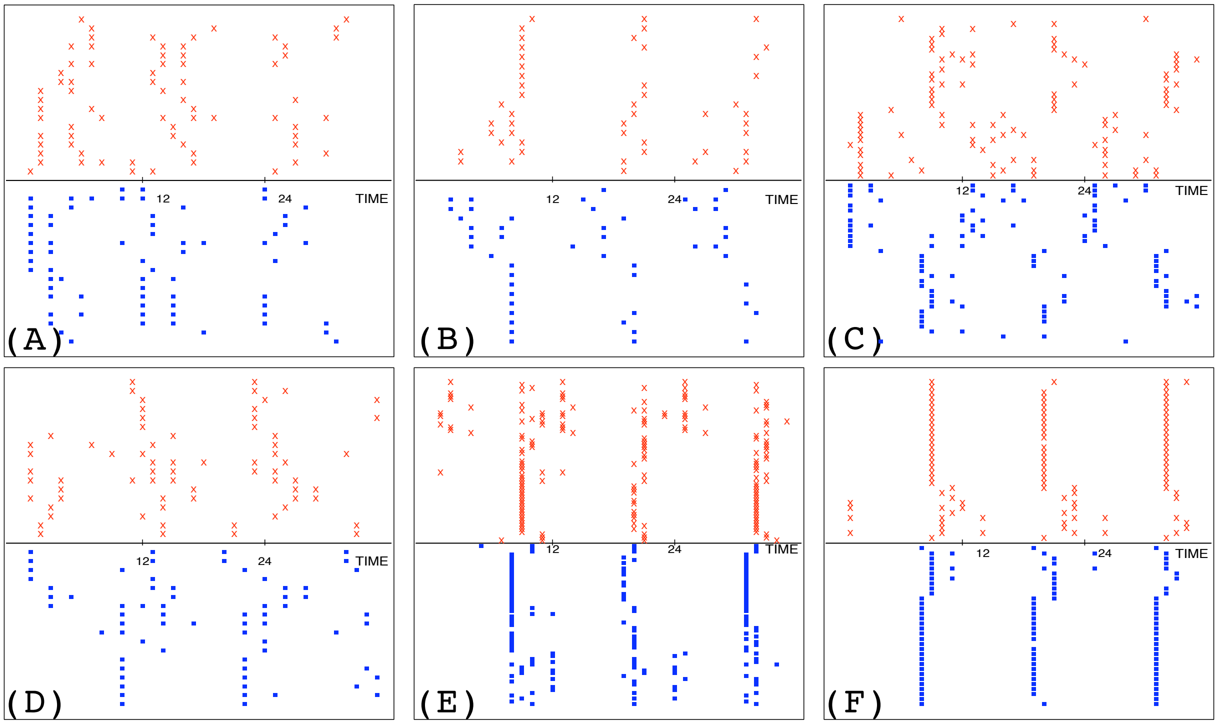


Figure 8: Visualization of three periodic cycles in removals and additions of Enzyme Relations in TCA cycle (A), urea cycle (B), glutamate metabolism (C), glycerophospholipid metabolism (D), purine metabolism (E) and pyrimidine metabolism (F) at the experiment of threshold 0.5. The points marked as "X" above the time axis represent removals, and the points marked as rectangles represent additions.

Table 1: Running time of ten dynamic graphs. Pathway denotes the name of the pathway represented by the dynamic graph. Max. Size and Min. Size denote the maximum and minimum size of a graph in the dynamic graph. Total Size denotes $\sum size(G_i)$ for $G_i \in DG$. Time is in seconds

Pathway	Max. Size	Min. Size	Total Size	Time
00010	522	65	7738	69.86
00020	294	46	4667	9.44
00030	192	57	4069	3.82
00220	236	58	4147	4.58
00251	394	110	7928	172.88
00330	184	61	4277	4.65
00561	183	44	2425	3.38
00564	231	57	4937	4.96
00230	643	161	10259	54.06
00240	486	85	6040	18.03

addition of some relations also show temporal cycles. Suppose there are two genes and a relation between two genes. One gene is always shown in the pathway, and the other is shown three times periodically. The relation is also shown three times like the latter gene, because the relation is activated only when both genes are activated. Because most genes and proteins work together, the temporal patterns in the relations between the molecules are also important as well as the temporal patterns in the existence of genes and proteins.

Figure 7 shows a visualization of three periodic cycles of 10 relations in the glycolysis pathway. In this experiment, the dynamic graph with threshold 0.6 shows a maximum of 13 relations at each time slice. 10 out of the 13 relations clearly show periodic cycles three times. Figure 8 shows the similar temporal patterns in the six other pathways, TCA cycle (A), urea cycle (B), glutamate metabolism (C), glycerophospholipid metabolism (D), purine metabolism (E) and pyrimidine metabolism (F). The points (marked as "X") above the time axis represent the patterns of removals and the points (marked as the rectangles) below the time axis represent the patterns of additions. The two time points with the same distance over the axis represent the removals and additions of the same subgraphs. The six visualizations show the temporal patterns in the graph rewriting rules of the major metabolic pathways. Even though there are some time points that do not show clear cycles, all ten pathways show the three periodic cycles of enzyme-enzyme relations. We can conclude that the removals and additions of the subgraphs including genes and relations show the temporal patterns of three periodic cycles. Table 1 shows the running time of Algorithm 1 on the ten dynamic graphs representing the ten metabolic pathways. Most cases are finished within a minute.

Figure 9 shows the temporal patterns in maplink-relations, which represent the relations between two enzymes that belong to two different pathways. $Link(+)$ denotes the time points when two pathways are linked to each other, and $Link(-)$ denotes the time points when they are disconnected. Because these relations are also activated by the

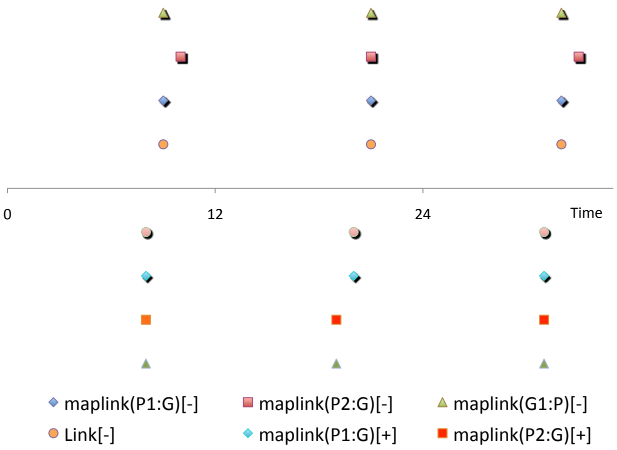


Figure 9: A visualization of three periodic cycles in time points when two pathways (Purine metabolism (00230) and Glutamate metabolism (00251)) are linked to each other at the experiment of threshold 0.5.

gene expression values, they also show three periodic cycles like enzyme-relations. In fact, all metabolic pathways in a cell are connected to each other. Practically, we classify the metabolic pathways for each function such as glycolysis, urea cycle and so on. The temporal patterns in the maplink-relations show when two pathways are connected or disconnected to each other. In addition to the temporal patterns, our results show the structural properties related to these patterns in next section.

Our results show three periodic cycles of enzyme-relations and maplink-relations over ten major metabolic pathways. We can observe similar temporal patterns in the four major categories of pathways. These temporal patterns of relations describe periodic cycles in the behaviors of the yeast biosystem corresponding to the periodic cycles of the gene expression of the yeast. The major events and behaviors of the biosystems accord with the metabolic cycles [22].

The experiments show that DynGRL discovers graph rewriting rules from dynamic graphs representing the yeast metabolic pathways changing over time. These graph rewriting rules represent temporal patterns that describe how the structure of the metabolic pathways change over time by showing which elements change periodically. These temporal patterns and graph rewriting rules help us to understand dynamic properties of the metabolic pathways. The results show not only temporal patterns in structural changes of metabolic pathways, but also temporal patterns in the connections between two different pathways.

6.2 Structural patterns

The other goal of this research is to show structural patterns in metabolic pathways as well as temporal patterns. Because an advantage of the graph representation is visualization, we can understand metabolic pathways better using structural analysis with temporal analysis. This section illustrates the use of discovered substructures with graph rewriting rules.

Figure 10 shows structural changes of the dynamic graph representing the partial glycolysis pathway introduced in figure 6. G_i represents the graph at time i. This dynamic graph contains 36 time series of graphs starting with a single vertex graph in time 1 to no vertex in time 36. The blue edge with the boxed labels between two sequential graphs represents the graph transformation using removal (-) or addition (+) of one of the six substructures (Sub A to F). For example, graph G_5 is transformed to G_6 with removal of Sub C and addition of Sub B. The red edges with the dot boxed labels in the rules represent the connection edges as described previously. The connection edges describe how the discovered substructures connect to the original graph.

As described previously, we show the graph rewriting rules between two graphs as a formula. Here, we show two examples of graph rewriting rules $GR_{1,2}$ and $GR_{5,6}$ as follows,

$$\begin{aligned} GR_{1,2} &= \{a_1, add_A, CE, CL)\}, \\ & \qquad CE = \{(d, S2, G2)\},\ CL = \{G\_to\_E\} \\ GR_{5,6} &= \{(r_1, rem_C, \emptyset, \emptyset), (a_1, add_B, \emptyset, \emptyset)\} \end{aligned}$$

where a_m and r_n denote the indices of the removal and addition rule in each graph rewriting rule, add_x and rem_y denote the substructure (Sub A to F) in figure 10. CE and CL denote the connection edges and connection edge labels respectively. The connection edge with a label G_to_E links Sub A to a gene YER178W in G_1 so that an enzyme is activated by two genes, YBR221C and YER178W, and a relation is created with the other enzyme that is activated by a gene, YNL071W. But CE and CL are all $\emptyset$ in $GR_{5,6}$ because there is no connection edge between the substructures (rem_C and add_B) and the original graphs (G_5 and G_6) respectively.

Figure 11 shows our visualization results of a removal and addition rule. The left figure shows a removal rule in our output and the right figure shows the same rule marked on the KEGG pathway map. The labels marked by "-[]" represent the labeled vertices and edges belonging to the substructures of removal rules. The labels are marked by "+[]" in the case of addition rules. Connection edges between the discovered substructures and original graphs are marked by "()". The removal of a gene YKL060C causes the removal of two enzyme-relations with one other gene YDR050C and a reaction R01070, which is a catalyzed by an enzyme produced by YKL060C (There can exist more than one relation with different properties between two genes in the KEGG data.). The graph also loses several connection edges between the removal structures and original graph. The DynGRL system helps us visualize removal or addition rules on the original graph with the connection edges. The results show how the substructures in graph rewriting rules are structurally connected to the original graphs and how the graphs change after removal or addition rules are applied.

In addition to the change of one element, our results show how the changes are related to other elements (i.e., which elements are removed or added at the same time) as shown in the discovered subgraphs and how the subgraphs are linked to the original graphs. Our results show patterns in the structural changes, not merely changes of amount. It allows us to better understand the structural properties as the pathways change over time.

In summary, we evaluated our algorithm in the experiments with 10 dynamic graphs each containing 36 graphs representing the yeast metabolic pathways in combination with the microarray data of yeast. 35 sets of graph rewrit-

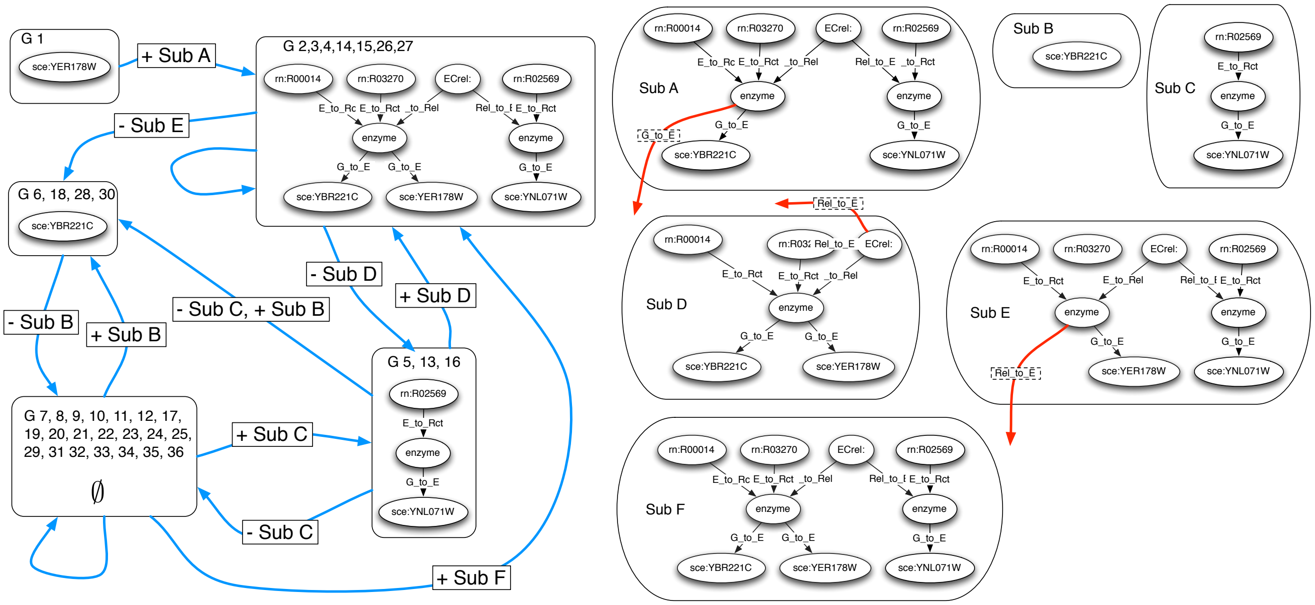


Figure 10: Structural changes of a dynamic graph representing the partial glycolysis pathway. Gi **denotes a graph at time** i **for** $1 \leq i \leq 36$**. The blue arrows with boxed labels between two graphs,** Gx **and** Gy**, represent the transformation from** Gx **to** Gy **by application of the rule in the label of the arrow. Sub** p **(**A **to** F**) represents the substructure in each rule (removal and addition), where the red arrows with the dot boxed labels from the substructures represent the connection edges. For example,** $G1$ **is transformed to** $G2$ **by addition of Sub** A**, which is connected by a connection edge labeled "G_to_E".**

ing rules for removals and additions are discovered during 35 time intervals. Temporal patterns in the graph rewriting rules show a number of substructures are removed and added periodically as showing three cycles. The graph rewriting rules and our visualization results describe how the discovered substructures are connected to the original graph and how the structures of graphs change over time. These temporal patterns and graph rewriting rules help us to understand temporal properties as well as structural properties of biological networks. Some discovered temporal and structural patterns in a specific disease can show us how they are different from normal patterns and help us investigate disease and develop a new treatment.

7. CONCLUSION

This research formalizes graph rewriting rules to describe structurally changing biological networks and proposes an algorithm, DynGRL, to discover graph rewriting rules in a dynamic graph. The algorithm is evaluated with the dynamic graphs representing the yeast metabolic pathways in combination with the microarray data. Our approach represents structural and temporal properties at the same time, and discovers novel patterns in both properties. The results show our dynamic graph-based relational learning approach discovers several novel temporal patterns in graph rewriting rules of the metabolic pathways such that some relations between genes and pathways are shown periodically. Additionally, the results show periodic cycles of temporal patterns in connections between two pathways. DynGRL can also help us to visualize the removed or added substructures to show how the graphs structurally change or how the substructures in rewriting rules are related to the original graphs.

The graph rewriting rules of biological networks can describe how the complex biosystems change over time. The learned temporal patterns in the rewriting rules can describe not only structural changes of metabolic pathways but also temporal patterns in series of the structural changes. Our approaches help us to better explore how biological networks change over time and guide us to understand the structural behaviors of the complex biosystems. Specifically, the temporal patterns in structural changes of the biosystems under specific conditions (e.g., infection) can provide essential information for drug discovery or disease treatment.

The future works follow several directions. First, we need more systematic evaluation for the discovered graph rewriting rules. Our evaluation will also include regenerating a dynamic graph using the discovered graph rewriting rules to compare with the original dynamic graph from real world data. In addition, we will also focus on the fully automated approach to learn temporal patterns in the discovered graph rewriting rules. Finally, we will evaluate how this approach can be used to predict future structures of biological networks using the learned temporal and structural patterns.

8. REFERENCES

[1] Kyoto university bioinformatics center, KEGG website. *http://www.genome.jp/kegg/pathway.*

[2] J. F. Allen and G. Ferguson. Actions and events in interval temporal logic. *Journal of Logic and Computation*, 4:531–579, 1994.

[3] R. Bowen. Lactose intolerance (lactase non-persistence). *Pathophysiology of the Digestive System*, 2006.

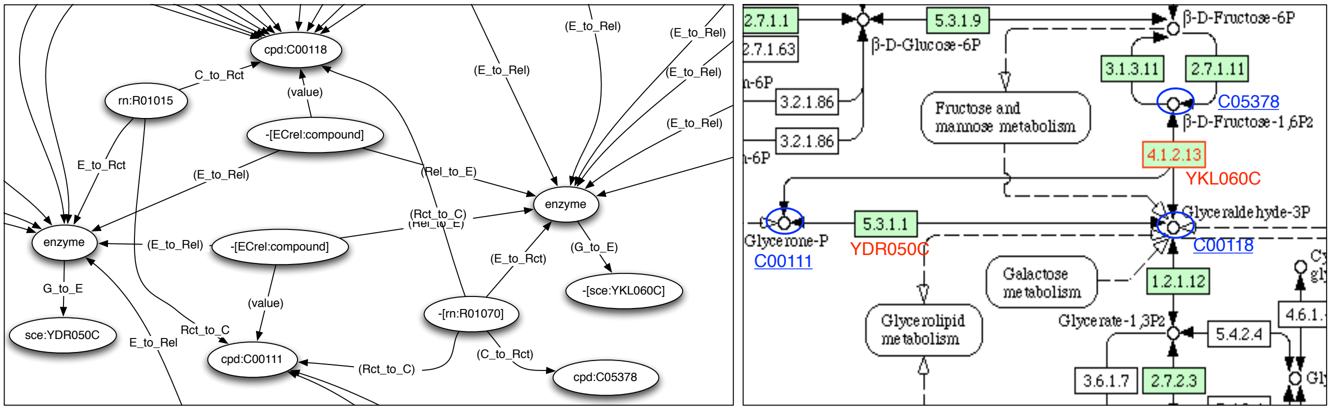


Figure 11: A visualization of discovered substructures of removal rule in a dynamic graph representing the glycolysis pathway in our output (left) and on the KEGG glycolysis pathway map (right). Labels marked by "-[]" represent the removal rules and labels marked by "()" represent the connection edges (left). Red rectangles represent two genes and blue circles represent three compounds in the removal rule (right). When YKL060C is removed, two enzyme-relations between two genes are also removed. Two reactions R01015 and R01070, involved with the three compounds, are also removed.

[4] H. Causton, J. Quackenbush, and A. Brazma. *A Beginner's Guide Microarray Gene Expression Data Analysis*. Blackwell, 2003.

[5] D. Cook and L. Holder. Substructure discovery using minimum description length and background knowledge. *Journal of Artificial Intelligence Research*, 1:231–255, 1994.

[6] D. Cook and L. Holder. Graph-based data mining. *IEEE Intelligent Systems*, 15(2):32–41, 2000.

[7] J. DeRisi, V. Iyer, and P. Brown. Exploring the metabolic and genetic control of gene expression on a genomic scale. *Science*, 278(5338):680–686, October 1997.

[8] H. Dörr. *Efficient Graph Rewriting and Its Implementation*. Springer, 1995.

[9] M. Farach-Colton, Y. Huang, and J. L. L. Woolford. Discovering temporal relations in molecular pathways using protein-protein interactions. In *Proceedings of RECOMB*, 2004.

[10] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co., 1979.

[11] T. Ho, C. Nguyen, S. S. Kawasaki, S. Le, and K. Takabayashi. Exploiting temporal relations in mining hepatitis data. *Journal of New Generation Computing*, 25:247–262, 2007.

[12] I. Jonyer, L. Holder, and D. Cook. Mdl-based context-free graph grammar induction. In *Proceedings of FLAIRS-2003.*, 2003.

[13] H. Kitano. Systems biology: A brief overview. *Science*, 295:1662–1664, 2002.

[14] J. Kukluk, C. You, L. Holder, and D. Cook. Learning node replacement graph grammars in metabolic pathways. In *Proceedings of BIOCOMP*, 2007.

[15] J. P. Kukluk, L. B. Holder, and D. J. Cook. Inference of node replacement recursive graph grammars. In *Proceedings of the SDM*, 2006.

[16] M. Kuramochi and G. Karypis. Frequent subgraph discovery. In *Proceedings of the ICDM*.

[17] D. J. Lockhart and E. A. Winzeler. Genomics, gene expression and dna arrays. *Nature*, 405:827– 836, 2000.

[18] K. Nielsen, P. Sørensen, and F. H. H.-G. Busse. Sustained oscillations in glycolysis: an experimental and theoretical study of chaotic and complex periodic behavior and of quenching of simple oscillations. *Biophysical Chemistry*, 7:49–62, 1998.

[19] K. Nupponen. The design and implementation of a graph rewrite engine for model transformations. Master's thesis, Helsinki University of Technology, Dept. of Comp. Sci. and Eng., May 2005.

[20] J. F. Roddick and M. Spiliopoulou. A survey of temporal knowledge discovery paradigms and methods. *IEEE Transactions on Knowledge and Data Engineering*, 14(4):750–767, 2002.

[21] G. Rozenberg. *Handbook of Graph Grammars and Computing by Graph Transformation*. World Scientific, 1997.

[22] B. Tu, A. Kudlicki, M. Rowicka, and S. McKnight. Logic of the yeast metabolic cycle: Temporal compartmentalization of cellular processes. *Science*, 310:1152–1158, 2005.

[23] X. Yan and J. Han. gspan: Graph-based substructure pattern mining. In *Proceedings of the ICDM*.

[24] C. You, L. Holder, and D. Cook. Application of graph-based data mining to metabolic pathways. In *Proceedings of IEEE ICDM Workshop on Data Mining in Bioinformatics*, 2006.