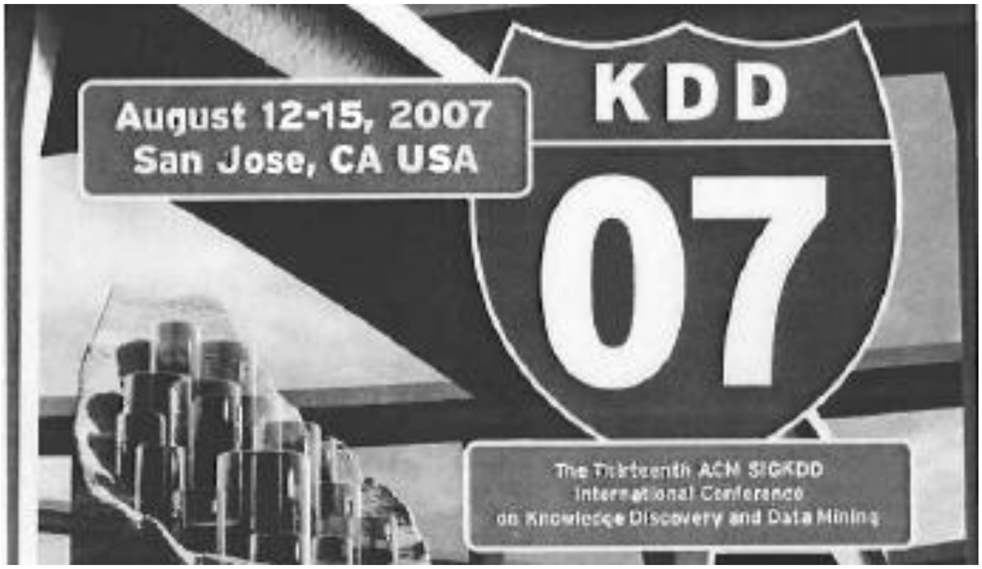
7th International Workshop on Data Mining in Bioinformatics (BIOKDD 2007)

Held in conjunction with SIGKDD conference, August 12, 2007


August 12-15, 2007
San Jose, CA USA
KDD
07
The Thirteenth ACM SIGKDD
International Conference
on Knowledge Discovery and Data Mining

Workshop Chairs

Jake Y. Chen
Stefano Lonardi
Mohammed Zaki

BIOKDD ‘07: Workshop on Data Mining in Bioinformatics August 12th, 2007 San Jose, CA, USA

in conjunction with
13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining

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

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

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

REMARKS

Bioinformatics is the science of managing, mining, and interpreting information from biological processes. 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 will play essential roles in understanding these fundamental problems and developing novel therapeutic/diagnostic solutions in post-genome medicine.

Data mining approaches seem ideally suited for bioinformatics, since the field is awash with data from high-throughput experimental instruments. The extensive databases of biological information available create both challenges and opportunities for developing novel knowledge discovery and data mining methods. To provide avenues to data mining researchers active in bioinformatics, we have been organizing the Workshops on Data Mining in Bioinformatics (BIOKDD), held annually in conjunction with the ACM SIGKDD Conference in 2001-2006. This is the 7th 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 bioinformatics offers. In our CFP, we encouraged paper submissions that present novel data mining techniques in the following sample topics:

- 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 full day event in conjunction with the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Jose, CA, August 12-15, 2007. The workshop was accepted in the 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/biokdd07.

This year, we accepted 10 papers out of 24 submissions into the workshop program and proceedings due to the exceptionally high quality of the submissions. Among these papers, 7 of the papers are accepted as full presentations (30 minutes each) and 3 of the papers are accepted as short presentations (20 minutes each). 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.

In closing, we want to thank Atul Butte, M.D., Ph.D. who agreed to give the keynote talk for this year's program. Dr. Butte is an Assistant Professor in Medicine (Medical Informatics) and Pediatrics at the Stanford University School of Medicine and the Lucile Packard Children's Hospital. His talk is entitled “Exploring Genomic Medicine Using Integrative Biology”.

WORKSHOP CO-CHAIRS

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

PROGRAM COMMITTEE

Amandeep Sidhu (Curtin University, Australia), Eamonn Keogh (University of California, Riverside), Daisuke Kihara (Purdue University), Giuseppe Lancia (University of Udine, Italy), Guojun Li (ShanDong University, China), Haixu Tang (Indiana University), Huanmei Wu (IUPUI), Isidore Rigoutsos (IBM T. J. Watson Research Center), Jason Wang (New Jersey Institute of Technology), Jie Zheng (NCBI, USA), Jignesh M. Patel (University of Michigan), Knut Reinert (Freie Universitt Berlin, Germany), Li Liao (University of Delaware), Luke Huan (University of Kansas), Fenglou Mao (University of Georgia), Muhammad Abulaish (Jamia Millia Islamia, India), Natasa Przulj (University of California, Irvine), Pan Du (Northwestern University), Phoebe Chen (Deakin University, Australia), Rahul Singh (San Francisco State University), Richard Scheuermann (University of Texas Southwestern), Simon Lin (Northwestern University), Xiang Zhang (Purdue University), Teresa Przytycka (NCBI/NLM, USA), Tony Hu (Drexel University), Xiaoyan Zhu (Tsinghua University, China), Yi Pan (Georgia State University), Yu-Ping Wang (University of Missouri)

ACKNOWLEDGEMENTS

We would like to thank all the program committee members, contributing authors, invited speaker, and attendees for contributing to the success of the workshop. Special thanks are also extended to the SIGKDD ’07 conference organizing committee, particularly Qiang Yang, for coordinating with us to put together the excellent workshop program on schedule.

WORKSHOP SCHEDULE AND INDEX TO PROCEEDING

Gene Selection by Matrix Reordering and Replicator Dynamics

Wenyuan Li
Department of Computer Science
University of Texas at Dallas
Richardson, TX 75083, USA
wenyuan.li@utdallas.edu

Xiuwen Zheng
Department of Biostatistics
University of Washington
Seattle 98195, USA
zhengx@u.washington.edu

Ying Liu[*]
Department of Computer Science & Department of Molecular and Cell Biology
University of Texas at Dallas
Richardson, TX 75083, USA
ying.liu@utdallas.edu

ABSTRACT

In most microarray data sets, there are often multiple sample classes, which are categorized into the normal or diseased type. The traditional feature selection methods consider multiple classes equally without paying attention to the up/down regulation across the normal and diseased classes, while the specific gene selection methods particularly consider the differential expressions across the normal and diseased, but ignore the existence of multiple classes. More importantly, most existing filter gene selection algorithms rank genes by individually considering each gene's expression values across classes, not by fully exploiting the overall inherent structure in microarray data. In this paper, we propose to employ matrix reordering techniques by taking into account the global between-class data distribution and local within-class data distribution in Microarray data for gene selection. In particular, we generalized a well-known population genetic algorithm, i.e., replicator dynamics, to reorder microarray data matrix with multiple classes. Our results show that our matrix reordering algorithm can effectively improve the accuracy of classifying the samples.

1. INTRODUCTION

The high-throughput genomic technologies have been poised to revolutionize early disease diagnosis, such as cancer, and biomarker discovery. DNA microarrays, among the most rapidly growing tools for genome analysis, are introducing a paradigmatic change in biology by shifting experimental approaches from single gene studies to genome-level analyses. Analysis of these high-throughput data poses both opportunities and challenges to the biologists, statisticians, and computer scientists. Unfortunately, one of important features in microarray data is the very high dimensionality with a small number of samples. There are tens of tens of thousands of features or genes and at most several hundreds of samples in the data set. This is so called "curse of dimensionality", which results in that most standard machine learning techniques, including supervised classification algorithms, are not directly and effectively applied. Instead, feature selection methods are generally used to first filter those features that contain a large degree of noisy, redundant and irrelevant information, and thus enable the subsequent use of disease classification algorithms. Consequently, a *biomarker* can be identified for disease screening and diagnosis, which is a subset of genes or proteins whose abundance is correlated with the state of a particular disease or condition.

Recent feature selection methods fall into two categories: *filter methods* and *wrapper methods* [18]. Filter methods select the features by evaluating the goodness of the features based on the intrinsic characteristics, which determines their relevance or discriminant powers with regards to the class labels [8, 19]. Most existing filter methods follow the methodologies of statistical tests (e.g. t-test, F-test) and information theory (e.g. mutual information or information gain) to rank the genes. In wrapper methods, gene selection is closely "embedded" in the classifier. The goodness and usefulness of a gene subset is evaluated by the estimated accuracy of the classifier, which was trained only with the subset of genes. Wrapper methods are computationally expensive for data sets with large number of features. Because of its computational efficiency, filter methods are adopted by most of works in microarray data analysis, but with the cost of having lower prediction accuracy than wrapper methods. Because most existing filter gene selection algorithms rank genes by individually considering each gene's expression values across classes, the overall inherent structure in microarray data matrix and relationships among genes and samples are still not clearly exploited.

Microarray data are often represented as a matrix $W_{m\times n}$, where each row is a gene and each column corresponds to a sample or condition. Therefore, from the viewpoints of matrix computation, some *particular trends*, *overall inherent structure* or *distinct patterns* can be discovered through matrix reordering: both rows and columns. This is the second "blessing of dimensionality" stated by [9]. Therefore, in this study, we focused on designing a matrix reordering method that is able to select genes from microarray data for biomarker discovery. Unlike existing matrix reordering techniques which are unsupervised learning, our matrix reordering algorithm considers class information in microarray

[*]Corresponding author.

BIOKDD '07, August 12, 2007, San Jose, California, USA
Copyright 2007 ACM 978-1-59593-839-8/07/0008 ...$5.00.

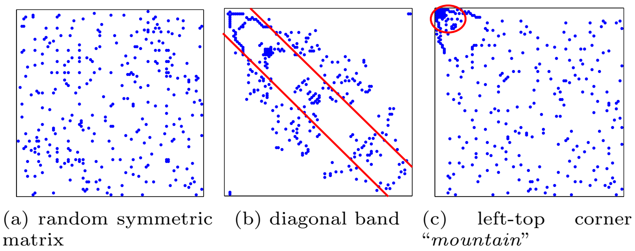

(a) random symmetric matrix
(b) diagonal band
(c) left-top corner "*mountain*"

Figure 1: Illustration of matrix reordering techniques for revealing particular patterns in the matrix. A blue dot indicates the value of 1 in a random symmetric matrix $W = (w_{ij})_{n\times n}$ where $w_{ij} \in \{0,1\}$. The patterns discovered in each image are highlighted by red lines or circles. (a). original random sparse symmetric matrix W; (b). diagonal band discovered by reordering W in (a) using Cuthill-McKee algorithm; (c). left-top corner "*mountain*" by reordering W in (a) using replicator dynamics.

data for the purpose of biomarker discovery. It simultaneously takes into account the global between-class data distribution (differentially expression) and local with-class data distribution (collection of low or high values). More importantly, microarray data sets may have more than two classes. Therefore, in the design of our matrix-based gene selection method, data with multiple classes is also considered.

Matrix reordering techniques have been developed more than thirty years ago in matrix computation field for permutating rows and columns of a matrix so that some particular structures can be revealed in the reordered matrix. They were often applied to sparse matrices, such as adjacency matrices of sparse graphs [7, 1, 10] and term-document matrix [4]. For example, [7] proposed a matrix reordering algorithm for a particular pattern "diagonal band", whose purpose is to collect high values (or non-zeros) to the diagonal band area of the reordered matrix. Fig. 1 shows how matrix reordering techniques can reveal underlying structures in a matrix. First, a random sparse symmetric matrix is generated in Fig. 1(a). When Cuthill-McKee algorithm is applied to this matrix, its diagonal band pattern is immediately discovered in the reordered matrix as shown in Fig. 1(b).

However, the pattern of diagonal band is not useful for biomarker discovery, because biomarker discovery is to identify a subset of genes which can significantly differentiate samples among different classes: genes with high values in one class and low values in other classes. Therefore, an essential step in the biomarker patterns is the collection of high or low values in single classes, e.g., differentially expressed genes. Hence, our method is focused on reordering microarray matrix for grouping high values together (denoted as "mountain" in short) and low values together (denoted as "valley" in short). In this way, the data distribution among classes can be revealed in the reordered matrix and thus it may be useful to biomarker discovery. Nonetheless, matrix reordering techniques can effectively and efficiently arrive at this target. One of the established algorithms is "replicator dynamics", which is able to reorder the symmetric matrix W so that high values "*mountain*" are collected to the left-top corner of the reordered matrix. We apply it to the above example matrix in Fig. 1(a) and the "*mountain*" can be clearly seen in the reordered matrix as shown in Fig. 1(c). From Fig. 1, we can see that matrix reordering techniques can reveal particular patterns, e.g., diagonal band, collection of high or low values, in the reordered matrix. However, few matrix reordering methods are able to analyze microarray data, which are unsymmetric and with multiple classes. More importantly, none of those methods were designed for gene selection. Therefore, in this study, a novel matrix reordering algorithm is designed for the purpose of biomarker discovery.

We started from a basic problem of revealing distinct "mountain" in unsymmetric single-class matrix. This is a building block problem for simultaneously exploring both "mountains" and "valleys" in unsymmetric multiple-classes matrix. To approach this basic problem, we developed a "Generalized Replicator Dynamics" (shortly denoted as GRD), which is based on a well-known population model in population genetics. As replicator dynamics is only applicable to symmetric matrices, instead, GRD we developed is applicable to general matrices. GRD can be proved to converge quickly and guarantee the optimization of the basic problem. By applying GRD to the data in a single class, the data matrix can be reordered by the solution of the basic problem so that the most distinct "*mountain*" (high values) or "*valley*" (low values) can be collected to the left-top corner of the reordered matrix. In this way, the value distribution of the data matrix can be clearly seen by drawing the reordered matrix. To discover "*mountains*" and "*valleys*" in multiple-class data matrix at the same time, we further extended GRD to be applicable from single-class data to multiple-class data. We called this Extended GRD as "EGRD" As a matrix reordering method, EGRD simultaneously rearranges the features and samples in the matrix so that "*mountains*" and "*valleys*" appear in the left-top corners within each class for the purpose of gene selection. In the top of reordered matrix, biologists may clearly find those genes or proteins, which show more obvious differences between diseased and healthy sample classes, because they are located in the top of those "*mountains*" or "*valleys*" in diseased or healthy sample classes. At the same time, *mountains* and *valleys* can provide analysts more information of how samples and features jointly contribute to the state of the particular disease, that is useful to understand biomarkers discovered.

The rest of the paper is organized as follows. We first presented replicator dynamics and showed its ability of symmetric matrix reordering for collecting the distinct *mountain* in the left-top corner of the reordered matrix in Section 2. In Section 3, GRD was developed for the general single-class matrix reordering. Based on GRD, in Section 4, then we moved to the design of EGRD for the general multiple-class matrix reordering. Finally, in Section 5, we conducted experiments on microarray data for their biomarker discovery. The results were evaluated and compared with other popular feature selection methods through cross validation methodology. In Section 5.2, conclusions and future works are presented.

2. REPLICATOR DYNAMICS FOR SYMMETRIC MATRIX REORDERING

Replicator Dynamics (RD) is one of the population dynamical methods which is also a kind of discrete dynamical system. It was first introduced and studied in evolutionary game theory to model the evolution of animal behavior [13].

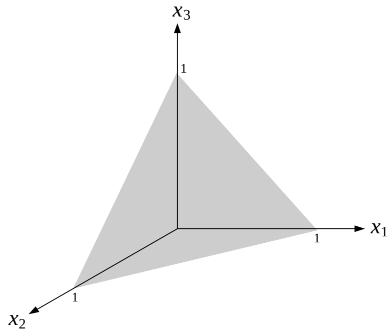


Figure 2: An example superplane Δ_3 (grey triangle) in $\mathbb{R}^3$.

Motivated by the population evolution, the idea of replicator dynamics has been independently studied in many fields, such as population genetics [6], mathematical ecology [3], computer vision [16] and so on. Next we will first introduce the problem that RD can solve and then review RD in detail.

Given a non-negative symmetric matrix $W = (w_{ij})_{n\times n}$, replicator dynamics assigns the i-th row or column a ranking value $x_i \geqslant 0$ for measuring its contribution to the collection of high values. These ranking values form a ranking vector $\mathbf{x} = (x_1, x_2, \ldots, x_n)^T$. Then replicator dynamics will maximize the following quadratic function,

$$L_W(\mathbf{x}) = \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}x_ix_j = \mathbf{x}^TW\mathbf{x} \qquad (1)$$

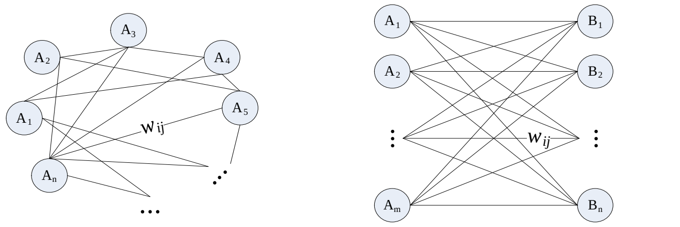
It is obvious that, after maximization process of L_W and obtaining the solution $\mathbf{x}^*$, those high values of w_{ij} in "mountain" most probably corresponds high values of x_i^* and x_j^* so that their multiplication $w_{ij}x_i^*x_j^*$ is high enough to maximize $L_W(\mathbf{x})$. Therefore, the decreasing order of elements in $\mathbf{x}^*$ is the reordering of W for collecting high values to the left-top corner. In practice, replicator dynamics restricts the ranking vector $\mathbf{x}$ as $\mathbf{x} \in \Delta_n$, where Δ_n is a superplane in n-dimensional Euclidean space as shown in Fig.2,

$$\Delta_n = \left\{ \mathbf{x} \in \mathbb{R}^n \Big| \quad \sum_{i=1}^{n} x_i = 1, \text{and } x_i \geqslant 0 \ (i = 1, 2, \ldots, n) \right\} \qquad (2)$$

Because replicator dynamics is a natural selection model in population genetics [12], in the next, for clearly expressing the ideas of generalizing replicator dynamics to unsymmetric matrix in single or multiple classes in the next two sections, we need to first introduce the mechanics of replicator dynamics for natural selection phenomenon in nature.

Consider a single chromosomal locus with n alleles $A_1, \ldots, A_n$. Let $x_1^{(t)}, \ldots, x_n^{(t)}$ denote the gene frequencies at the mating stage in the parental generation (the t-th generation). The assumption of random mating leads to $x_i^{(t)}x_j^{(t)}$ for the probability that a zygote carries the gene pair (A_i, A_j). Let w_{ij} be the probability that an (A_i, A_j)-individual survives to adult age. Since the gene paris (A_i, A_j) and (A_j, A_i) belong to the same genotype, the selective value $w_{ij} \geqslant 0$ and $w_{ij} = w_{ji}$. The selection matrix $W = (w_{ij})_{n\times n}$ is therefore symmetric.

If N is the number of zygotes in the new generation, the $(t+1)$-th generation, then $x_i^{(t)}x_j^{(t)}N$ of them carry the gene pair (A_i, A_j) of which $w_{ij}x_i^{(t)}x_j^{(t)}N$ survive to adulthood. Therefore, the total number of individuals reaching the mating stage is $\sum_{r,s=1}^{n} w_{rs}x_r^{(t)}x_s^{(t)}N$. Let f_{ij} denote the frequency of the gene pair (A_i, A_j) in the adult stage of the $(t+1)$-th generation, we can obtain,



(a) replicator dynamics (b) generalized replicator dynamics

Figure 3: Alleles A_i or B_j as vertices and their mating survival probabilities w_{ij} as edge weights in replicator dynamics and generalized replicator dynamics.

$$f_{ij} = \frac{w_{ij}x_i^{(t)}x_j^{(t)}N}{\sum_{r,s=1}^{n} w_{rs}x_r^{(t)}x_s^{(t)}N} \qquad (3)$$

Since $x_i^{(t+1)}$ is the frequency of the allele A_i in the adult stage of the $(t+1)$-th generation, we have $x_i^{(t+1)} = \sum_{j=1}^{n} f_{ij}$. This leads to the relation

$$x_i^{(t+1)} = x_i^{(t)} \frac{\sum_{j=1}^{n} w_{ij}x_j^{(t)}}{\sum_{r,s=1}^{n} w_{rs}x_r^{(t)}x_s^{(t)}} \qquad i = 1, \ldots, n \qquad (4)$$

Eq.(4) is the *selection model.* It can be rewritten in the matrix form as follows,

$$x_i^{(t+1)} = x_i^{(t)} \frac{(W\mathbf{x}^{(t)})_i}{\mathbf{x}^{(t)T}W\mathbf{x}^{(t)}} \qquad i = 1, 2, \ldots, n \qquad (5)$$

where $(W\mathbf{x}^{(t)})_i$ denotes the i-th component of the vector $W\mathbf{x}^{(t)}$, and the state of the gene pool of the t-th generation is given by the vector $\mathbf{x}^{(t)} = (x_1^{(t)}, \ldots, x_n^{(t)})^T$ of gene frequencies. $\mathbf{x}^{(t)}$ has non-negative components summing up to one, and belongs to the simplex Δ_n. To succinctly state n formulas in Eq.(5), we use the dot product function (i.e., given two vectors $\mathbf{x}$ and $\mathbf{y}$, $\mathbf{x}.*\mathbf{y} = (x_1y_1, \ldots, x_ny_n)^T$ is a vector of dot product of $\mathbf{x}$ and $\mathbf{y}$) and normalization function (i.e., $\mathtt{t}_1(\mathbf{x}) = (\frac{x_1}{|\mathbf{x}|}, \ldots, \frac{x_n}{|\mathbf{x}|})$, where $|\mathbf{x}| = \sum_{i=1}^{n} x_i$) to rewrite it as a formula,

$$\mathbf{x}^{(t+1)} = \mathtt{norm}_1\big(\mathbf{x}^{(t)}.*(W\mathbf{x}^{(t)})\big) \qquad (6)$$

Eq.(6) describes the action of selection from one generation to the next, and therefore the map sending $\mathbf{x}^{(t)}$ to $\mathbf{x}^{(t+1)}$ defines a discrete dynamical system on the space Δ_n, called *Replicator Dynamics.*

Definition 1 (Replicator Dynamics). *Let $W_{n\times n}$ be a non-negative symmetric matrix. Given the vector $\mathbf{x}^{(t)} = (x_1^{(t)}, \ldots, x_n^{(t)})^T \in \mathbb{R}_+^n$ being the status of the system in the t-th iteration, we define the dynamical system as Eq.(6).*

Since the selection model from evolutionary biology defines a discrete dynamical system *replicator dynamics*, we are interested in its stationary states and the optimization ability. Before that, we first introduce the average fitness of the population.

Definition 2. (Average Fitness of Population in Selection Model). *Given $x_i^{(t)} x_j^{(t)}$ the frequency of the zygote of (A_i, A_j) and the selective value w_{ij} the probability that it survives to adult age, we define $\sum_{i,j=1}^{n} w_{ij} x_i^{(t)} x_j^{(t)}$ is the average fitness (or average selective value) of the population in the (t)-th generation. The average fitness can be written in the matrix form as $L_W(\mathbf{x}^{(t)}) = \mathbf{x}^{(t)T} W \mathbf{x}^{(t)}$ and therefore the same as the Lagrangian of the graph $G(A, W)$, where A is the set of alleles representing the vertices.*

The fundamental theorem of natural selection tells us that under selection model, the average fitness increases from generation to generation. Refer to [13, 12] for detailed proof of this theorem.

Theorem 1. (Fundamental Theorem of Natural Selection by Replicator Dynamics). *For the replicator dynamics given by Eq.(5), the average fitness $L_W(\mathbf{x}^{(t)})$ increases with the generation t increasing in the sense that*

$$L_W(\mathbf{x}^{(t+1)}) \geqslant L_W(\mathbf{x}^{(t)}) \tag{7}$$

with equality if and only if $\mathbf{x}^{(t)}$ is an equilibrium point $\mathbf{x}^*$.

3. GENERALIZED REPLICATOR DYNAMICS FOR UNSYMMETRIC MATRIX REORDERING IN SINGLE CLASS

Given a non-negative unsymmetric matrix $W = (w_{ij})_{m \times n}$ without class information (i.e., only one single class), similar to the problem formulation in symmetric matrix described in the above section, the problem of collecting high values to the left-top corner of the reordered W can be formulated as follows.

We assign the vector $\mathbf{x} = (x_1, x_2, \ldots, x_m)^T$ to rank rows of W and the vector $\mathbf{x} = (y_1, y_2, \ldots, y_n)^T$ to rank columns of W. Then we generalize the optimization function $L_W(\mathbf{x})$ in Eq.(1) from symmetric matrix to unsymmetric matrix in the following,

$$L_W(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{m} \sum_{j=1}^{n} w_{ij} x_i y_j = \mathbf{x}^T W \mathbf{y} \tag{8}$$

$\mathbf{x}$ and $\mathbf{y}$ are subject to $\mathbf{x}^S \in \Delta_m$ and $\mathbf{y}^S \in \Delta_n$ respectively.

Therefore, to maximize the function $L_W(\mathbf{x}, \mathbf{y})$, in the next, we generalize replicator dynamics for maintaining the optimization ability of replicator dynamics in unsymmetric matrices. The mechanics in replicator dynamics is automatically generalized as well, including natural selection model and fundamental theorem.

The selection model above is based on the selection matrix $W_{n \times n}$ that describes the survival probability of the zygotes of any two alleles (A_i, A_j). Therefore, W is symmetric and the adjacency matrix of a weighted graph whose vertex set is alleles and edge weight is w_{ij} in W. This weighted graph is shown in Fig.3(a). In this section, we generalize the replicator dynamics to a more general selection matrix $W_{m \times n}$ that denotes the probability of the zygotes of any two alleles (A_i, B_j) from allele types A and B. Here, we suppose that there are two types (or sets) of alleles $A = \{A_1, \ldots, A_m\}$ and $B = \{B_1, \ldots, B_n\}$. There are restrictions of mating in these two types of alleles: the mating can only happen between different types of alleles. For example, the allele A_i can mate with any B-type allele B_j, but always fail with any other A-type allele. Therefore, the selection matrix $W_{m \times n}$ and two sets of alleles A and B forms a bipartite graph as shown in Fig.3(b).

Let $x_1^{(t)}, \ldots, x_m^{(t)}$ denote the gene frequencies of A-type alleles $A_1, \ldots, A_m$, and $y_1^{(t)}, \ldots, y_n^{(t)}$ the gene frequencies of B-type alleles $B_1, \ldots, B_n$, at the mating stage in the parental generation (the t-th generation). The assumption of random mating leads to $x_i^{(t)} y_j^{(t)}$ for the probability that a zygote carries the gene pair (A_i, B_j).

If N is the number of zygotes in the new generation, the $(t+1)$-th generation, then $x_i^{(t)} y_j^{(t)} N$ of them carry the gene pair (A_i, B_j) of which $w_{ij} x_i^{(t)} y_j^{(t)} N$ survive to adulthood. Therefore, the total number of individuals reaching the mating stage is $\sum_{r=1}^{m} \sum_{s=1}^{n} w_{rs} x_r^{(t)} y_s^{(t)} N$. Let f_{ij} denote the frequency of the gene pair (A_i, B_j) in the adult stage of the $(t+1)$-th generation, we can obtain,

$$f_{ij} = \frac{w_{ij} x_i^{(t)} y_j^{(t)} N}{\sum_{r=1}^{m} \sum_{s=1}^{n} w_{rs} x_r^{(t)} y_s^{(t)} N} \tag{9}$$

Since $x_i^{(t+1)}$ is the frequency of the allele A_i in the adult stage of the $(t+1)$-th generation, we have $x_i^{(t+1)} = \sum_{j=1}^{m} f_{ij}$. This leads to the relation

$$x_i^{(t+1)} = x_i^{(t)} \frac{\sum_{j=1}^{n} w_{ij} y_j^{(t)}}{\sum_{r=1}^{m} \sum_{s=1}^{n} w_{rs} x_r^{(t)} y_s^{(t)}} \qquad i = 1, \ldots, m$$

It can be rewritten in the matrix form as follows,

$$x_i^{(t+1)} = x_i^{(t)} \frac{(W\mathbf{y}^{(t)})_i}{\mathbf{x}^{(t)T} W \mathbf{y}^{(t)}} \qquad i = 1, 2, \ldots, m \tag{10}$$

The m formulas in Eq.(10) can be rewritten in a formula as,

$$\mathbf{x}^{(t+1)} = \mathtt{norm}_1\big(\mathbf{x}^{(t)}. * (W\mathbf{y}^{(t)})\big) \tag{11}$$

For B-type alleles, since $y_j^{(t+1)}$ is the frequency of the allele B_j in the adult stage of the $(t+1)$-th generation, we have $y_j^{(t+1)} = \sum_{i=1}^{m} f'_{ij}$, where f'_{ij} is computed according to Eq.(9) by substituting $x_i^{(t)}$ with $x_i^{(t+1)}$. This leads to the relation

$$y_j^{(t+1)} = y_j^{(t)} \frac{\sum_{i=1}^{m} w_{ij} x_i^{(t+1)}}{\sum_{r=1}^{m} \sum_{s=1}^{n} w_{rs} x_r^{(t+1)} y_s^{(t)}} \qquad j = 1, \ldots, n$$

Its matrix form is,

$$y_j^{(t+1)} = y_j^{(t)} \frac{(W^T \mathbf{x}^{(t+1)})_j}{\mathbf{y}^{(t)T} W^T \mathbf{x}^{(t+1)}} \qquad j = 1, 2, \ldots, n \tag{12}$$

The n formulas in Eq.(12) can be rewritten in a formula as,

$$\mathbf{y}^{(t+1)} = \mathtt{norm}_1\big(\mathbf{y}^{(t)}. * (W^T \mathbf{x}^{(t+1)})\big) \tag{13}$$

The state of the gene pool of the t-th generation is given by the vector $\mathbf{x}^{(t)} = (x_1^{(t)}, \ldots, x_m^{(t)})^T$ of gene frequencies in A-type alleles and the vector $\mathbf{y}^{(t)} = (y_1^{(t)}, \ldots, y_n^{(t)})^T$ of gene frequencies in B-type alleles. $\mathbf{x}^{(t)}$ and $\mathbf{y}^{(t)}$ have non-negative components summing up to one, and belong to the simplex Δ_m and Δ_n respectively. Eq.(11) and Eq.(13) are the *generalized selection model* for two types of alleles A and B. It describes the action of selection between two types of alleles from one generation to the next, and therefore the map sending $\mathbf{x}^{(t)}$ and $\mathbf{y}^{(t)}$ to $\mathbf{x}^{(t+1)}$ $\mathbf{y}^{(t+1)}$ defines a discrete dynamical system on the spaces Δ_m and Δ_n, called *Generalized Replicator Dynamics* (GRD).

Definition 3 (Generalized Replicator Dynamics). *Let $W_{m\times n}$ be a non-negative matrix. Given the vector $\mathbf{x}^{(t)} = (x_1^{(t)}, \ldots, x_m^{(t)})^T \in \mathbb{R}_+^m$ and the vector $\mathbf{y}^{(t)} = (y_1^{(t)}, \ldots, y_n^{(t)})^T \in \mathbb{R}_+^n$ being the status of the system in the t-th iteration, we define the discrete dynamical system as Eq.(11) and Eq.(13).*

Correspondingly, we studied the the fixed points and optimization ability of generalized replicator dynamics. Next the average fitness of the population and the fundamental theorem of natural selection in the generalized selection model are given.

Definition 4. (Average Fitness of Population in Generalized Selection Model). *Given $x_i^{(t)}y_j^{(t)}$ the frequency of the zygote of (A_i, B_j) and the selective value w_{ij} the probability that it survives to adult age, we define $\sum_{i=1}^m \sum_{j=1}^n w_{ij}x_i^{(t)}y_j^{(t)}$ is the average fitness (or average selective value) of the population in the (t)-th generation. The average fitness in the matrix form is $L_W(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}) = \mathbf{x}^{(t)T}W\mathbf{y}^{(t)} = \mathbf{y}^{(t)T}W^T\mathbf{x}^{(t)}$ and therefore the same as the generalized function of a bipartite graph $G(A, B, W)$, where A and B are two sets of alleles representing the vertices.*

Theorem 2. (Fundamental Theorem of Natural Selection by Extended Replicator Dynamics). *For the generalized replicator dynamics given by Eq.(11) and Eq.(13), the average fitness $L_W(\mathbf{x}^{(t)}, \mathbf{y}^{(t)})$ increases with the generation t increasing in the sense that*

$$L_W(\mathbf{x}^{(t+1)}, \mathbf{y}^{(t+1)}) \geqslant L_W(\mathbf{x}^{(t)}, \mathbf{y}^{(t)}) \qquad (14)$$

with equality if and only if $\mathbf{x}^{(t)}$ and $\mathbf{y}^{(t)}$ are two equilibrium points $\mathbf{x}^*$ and $\mathbf{y}^*$ respectively.

Proof. See http://www.utdallas.edu/∼ying.liu /BIOKDD2007.html □

If let W be symmetric, $\mathbf{x}$ and $\mathbf{y}$ are associated with the same set of vertices and thus equal to each other. Hence Eq.(11) and Eq.(13) are reduced to Eq.(6) and therefore replicator dynamics become a special instance of generalized replicator dynamics. In practice, the iteration of about 50 is enough for generalized replicator dynamics to get converged. Therefore, its computational complexity is $\mathbf{O}(k(2h+m+n))$, where k is the number of iterations, h, m and n are the number of non-zeros, numbers of rows and columns in W respectively. If ignoring k, the final complexity is $\mathbf{O}(2h + m + n)$. Therefore, generalized replicator dynamics is very efficient.

4. GENERALIZED ERD FOR UNSYMMETRIC MATRIX REORDERING IN MULTIPLE CLASSES

In Section 2 and Section 3, we have shown how to discover distinct "mountain" within a single class by matrix reordering. In this section, we shall focus on a more complicated problem of finding the "mountain" and "valley" which are collected parallel (i.e. with the same rows or genes/proteins) but on the left-top corner of each class submatrix [1]. Those genes (or rows) which contribute to the parallel "mountain" and "valley" on the top of the reordered matrix, are deemed to be potential genes or proteins for biomarker. The more top they are placed, the higher differential expressions they have. Those top-ranked genes in distinct parallel "mountain" and "valley" contribute much more differential expressions across negative and positive classes. Therefore, the solution of parallel "mountain" and "valley" can not only rank differentially expressed genes, but also visually show the expression values' distribution within class (i.e., collecting low/high values to left-top corner of each class submatrix) and between class (i.e., parallel collecting low and high values in negative and positive class respectively).

RD and GRD are designed to approach the problems of collecting the high values to the left-top corner of the matrix rearranged by the element orders of the solution $\mathbf{x}^*$ and $\mathbf{y}^*$. However, they only investigate the data which has no class labels. In this section, a similar but more complicated task, parallel "valley and mountain" (up regulation) and parallel "mountain and valley" (down regulation) across multiple classes, is considered. Because RD and GRD have been proved that they are able to quickly approximate the optimization of the functions $L_W(\mathbf{x})$ and $L_W(\mathbf{x}, \mathbf{y})$ respectively, such capability of reordering matrix can be introduced to our task of gene selection for biomarker discovery. In the following, we will present how we customize and generalize ERD to our target in microarray data analysis.

Considering the general case of microarray data, suppose the data set consists of m genes and n samples with k classes, whose number of samples are $n_1, \ldots, n_k$ respectively and $n_1 + \ldots + n_k{=}n$. Without losing the generality, we suppose the first k_- classes are negative, the following k_+ classes are positive, and $k_- + k_+ = k$. Therefore, a general gene-sample matrix $W_{m\times n} = [\ \underbrace{W_i^-}_{1\leqslant i\leqslant k_-}\ ,\ \underbrace{W_i^+}_{1\leqslant i\leqslant k_+}\ ]$ is shown with submatrix blocks in Fig.4(a). Like fold change, the difference of values between negative and positive classes can show the up or down tendency [2].

Because the target of analyzing differentially expressed genes is to find up-regulated or down-regulated genes between negative and positive sample classes, the basic resonance model should be changed, from collecting high values to the left-top corner of W', to:

1. **Within-class data distribution:** A series of low values collections in each W_i^- into the left-top corner, and simultaneously a series of high values collections in each W_i^+ into the left-top corner.

[1] Each sample class forms a submatrix where rows are the whole set of genes and columns are the samples in this class.

[2] The up tendency means that low values are in samples of the negative class, while high values are in samples of the positive class. Vice versa for the down tendency.

2. **Between-class data distribution:** Controlling the differences of left-top corners between the negative classes W_i^- and W_i^+.

Therefore, to meet these two goals, we extended generalized replicator dynamics, called EGRD, according to this task as follows.

1. Transformation of W: before performing EGRD, we need to transform the original gene-sample matrix W to W'. The structure of W is made of the submatrix blocks W_i^- and W_i^+ of negative classes and positive classes as shown in Fig.4(a). In the case of finding up tendency and differentially expressed genes, since we need to collect the low values of W_i^- into the left-top corner, we reverse the values of W_i^- so that low values become high and vice versa. In other words, we perform the transformation by $W_i'^- = 1 - W_i^-$. In this way, the result of collecting high values of $W_i'^-$ and $W_i'^+$ into their own left-top corners naturally lead to the result of collecting the low values of W_i^- into the left-top corners and the high values of W_i^+ into the left-top corners. This is an essential step to meet the first goal aforementioned. We can also use other reverse functions in stead of the simple $1 - x$ function used in Fig.4(b). Similarly, we can transform W by $W_i'^+ = 1 - W_i^+$ in the case of finding down-regulated and differentially expressed genes.

2. The k partitions of the allele set B: an implicit requirement in the first goal is that the relative order of each class (submatrix $W_i'^-$ or $W_i'^+$) should be kept the same after performing EGRD and sorting W'. For example, after running our algorithm, it is required that all columns of the submatrix $W_2'^-$ appear after all columns of $W_1'^-$, although we can change the order of columns or samples within $W_1'^-$ or $W_2'^-$. To satisfy this requirement, we partition the original vector $\mathbf{y}$ of gene frequencies in B-type alleles into k parts corresponding to k classes or submatrices. Specifically, $\mathbf{y} = (\mathbf{y}_1; \ldots; \mathbf{y}_k)$ [3], where each $\mathbf{y}_i$ corresponds to a sample class. In the process of EGRD, we separately normalize each $\mathbf{y}_i$ and then sum them together with the factor α to control the differentiation between the negative and positive classes.

3. The factor α for controlling the differentiation between the negative and positive classes: the gene frequency vector of $\mathbf{y}$ is divided into $k = k_- + k_+$ parts, each of which is normalized independently. Therefore, we can control the differentiation between the negative and positive classes, by magnifying the resonance strengths $\mathbf{x}_i^{+(t+1)} = \mathtt{norm}_1(\mathbf{x}^{+(t)} .* (W_i'^+ \mathbf{y}_i^{+(t)}))$ of k_+ positive classes, or minifying the frequency subvectors $\mathbf{x}_i^{-(t+1)} = \mathtt{norm}_1(\mathbf{x}^{-(t)} .* (W_i'^- \mathbf{y}_i^{-(t)}))$ of k_- negative classes. In formal,

$$\mathbf{x}^{(t+1)} = \mathtt{norm}_1\Big(\underbrace{\mathbf{x}_1^{-(t+1)} + \ldots + \mathbf{x}_{k_-}^{-(t+1)}}_{k_- \text{ negative classes}} + \underbrace{\alpha \mathbf{x}_1^{+(t+1)} + \ldots + \alpha \mathbf{x}_{k_+}^{+(t+1)}}_{k_+ \text{ positive classes}} \Big) \tag{15}$$

where $\alpha \geqslant 1$ and α as a scaling factor is multiplied with the normalized positive classes' resonance strength vectors. With the increasing of α, the proportions of positive classes in the gene frequency vector $\mathbf{x}$ will increase and thus result in the increasingly large differences in the top-left corners between positive and negative classes. In this way, the user can tune α to get a suitable differential contrast of two types of classes.

4. Smoothness of gene frequency vectors of B-type alleles: In practice, we found that the partitioned gene frequency vectors of B-type alleles $\mathbf{y}_i^+$ or $\mathbf{y}_i^-$ often converges to the extreme distribution of elements: few elements approach to 1 while the rest approximate to 0. Therefore, to smooth the element distribution of $\mathbf{y}_i^+$ and $\mathbf{y}_i^-$, we introduced the sigmoid function [4] that is widely used in neural networks. Therefore, we define the new normalization function incorporating the sigmoid function as $\mathtt{normsig}_1(\mathbf{y}) = \mathtt{norm}_1(\mathtt{sig}(\mathtt{norm}_1(\mathbf{y})))$. In this way, the gene frequency vectors are smoothed. We have made experiments to test the convergence of the EGRD after using the normalization function $\mathtt{normsig}_1$. The empirical results show that it can quickly converge.

To summarize the above changes of the resonance model, we draw the architecture of the EGRD in Fig.5 and express its process in the following formulas:

$$\begin{aligned}
\mathbf{x}_i^{-(t+1)} &= \mathtt{norm}_1\big(\mathbf{x}^{(t)} .* (W_i'^- \mathbf{y}_i^{-(t)})\big), & i &= 1, \ldots, k^- \\
\mathbf{x}_i^{+(t+1)} &= \mathtt{norm}_1\big(\mathbf{x}^{(t)} .* (W_i'^+ \mathbf{y}_i^{+(t)})\big), & i &= 1, \ldots, k^+ \\
\mathbf{x}^{(t+1)} &= \mathtt{norm}_1\big(\textstyle\sum_{i=1}^{k^-} \mathbf{x}_i^{-(t+1)} + \alpha \sum_{i=1}^{k^+} \mathbf{x}_i^{+(t+1)} \big) \\
\mathbf{y}_i^{-(t+1)} &= \mathtt{normsig}_1\big(\mathbf{y}^{-(t)} .* ((W_i'^-)^T \mathbf{x}^{(t+1)})\big), & i &= 1, \ldots, k^- \\
\mathbf{y}_i^{+(k+1)} &= \mathtt{normsig}_1\big(\mathbf{y}^{-(t)} .* ((W_i'^+)^T \mathbf{x}^{(t+1)})\big), & i &= 1, \ldots, k^+
\end{aligned} \tag{16}$$

where $\mathbf{x}_i, \mathbf{x}_i^+, \mathbf{x}_i^- \in \mathbb{R}^{m \times 1}$ and $\mathbf{y}_i^- \in \mathbb{R}^{n_i^- \times 1}$, $\mathbf{y}_i^+ \in \mathbb{R}^{n_i^+ \times 1}$. Comparing Eq.(11) and Eq.(13) in GRD with Eq.(16), we partitioned the matrix W' to k submatrix blocks and divided the gene frequency vector of B-type alleles $\mathbf{y}$ into k subvectors. Therefore, two equations in the extended replicator dynamics are expanded to the $(2k + 1)$ equations in EGRD.

Algorithm of EGRD will appear here. We also formally summarize it as Algorithm 1 `EGRD` for the data reliability assessment.

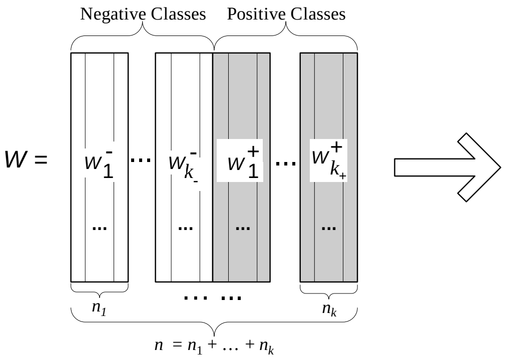
In practice, `GERD` can quickly converge. Considering that `EGRD` is a extended generalized replicator dynamics by partitioning the matrix into k submatrices, its computational complexity is the same as the extended replicator dynamics on the whole matrix, i.e., $\mathbf{O}(2h + m + n)$.

5. EXPERIMENTAL RESULTS

In this section, we conducted the experiments on the Leukemia data set and compared our method with five popular filter feature selection methods, T-statistics (T) [14], Information Gain (IG) [5], ReliefF [15], Correlation-based Feature Selection (CFS) [11] and Redundancy Based Filter (RBF) [19].

[3] The concatenation of $k = k_- + k_+$ vectors is expressed in MATLAB format.

[4] The sigmoid function is defined on the scalar number x as, $\mathtt{sig}(x) = \frac{1}{1+\exp(-x)}$. Therefore, for a vector $\mathbf{x}$, the corresponding sigmoid function is $\mathtt{sig}(\mathbf{x}) = \big(\mathtt{sig}(x_1), \ldots, \mathtt{sig}(x_n)\big)^T$



	up regulation	down regulation
$W_i'^- =$ $W_i'^+ =$	$1 - W_i^-$ W_i^+	W_i^- $1 - W_i^+$
$W' =$	$[\underbrace{1 - W_i^-}_{1\leqslant i\leqslant k_-}, \underbrace{W_i^+}_{1\leqslant i\leqslant k_+}]$	$[\underbrace{W_i^-}_{1\leqslant i\leqslant k_-}, \underbrace{1 - W_i^+}_{1\leqslant i\leqslant k_+}]$

(a) original matrix $W = [\underbrace{W_i^-}_{1\leqslant i\leqslant k_-}, \underbrace{W_i^+}_{1\leqslant i\leqslant k_+}]$

(b) transformed matrix $W' = [\underbrace{W_i'^-}_{1\leqslant i\leqslant k_-}, \underbrace{W_i'^+}_{1\leqslant i\leqslant k_+}]$

Figure 4: Transformation of the matrix W: the transformed matrix W' has the same structure of submatrix blocks as shown in (a), but with different submatrix $W_i'^-$ and $W_i'^+$ as listed in (b).

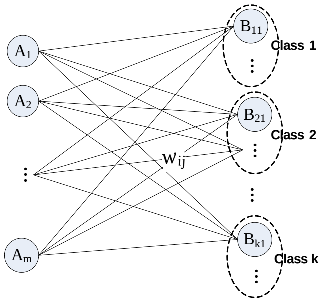


Figure 5: Alleles A_i and $B_{l\_}$ with k classes as vertices and their mating survival probabilities w_{ij} as edge weights in generalized extended replicator dynamics.

Algorithm 1 `EGRD`

```
Input:  (1) $W_{m\times n}$, genomic or proteomic matrix from $m$
            gene set $G$ and $n$ samples set $S$;
        (2) $(n_1,\dots,n_k)^T$, sizes of the $k$ sample classes
            with the submatrix structure as in Fig.4(a).
        (3) $(k_-,k_+)^T$, numbers of negative and positive
            classes.
        (4) tendency option, down or up;
        (5) $\alpha$, differentiation factor.
Output: (1) $(g_1,\dots,g_m)$, ranking sequence of $m$ genes;
        (2) $(s_1,\dots,s_n)$, ranking sequence of $n$ samples.
1: preprocess $W$ so that the values of $W$ in [0,1].
2: transform $W$ to $W'$ according to formulas in Fig. 4(b)
   with the knowledge of the matrix structure given by
   $(n_1,\dots,n_k)^T$, and $(k_-,k_+)^T$ and tendency option.
3: iteratively run formulas in Eq.(16) to obtain the con-
   verged $\mathbf{x}^*$ and $\mathbf{y}_i^*$ ($i$=1, 2, ..., $k$).
4: sort $\mathbf{x}^*$ in decreasing order to get the ranking sequence
   $(g_1,\dots,g_m)$, and sort each of $\mathbf{y}_1^*,\dots,\mathbf{y}_k^*$ in decreasing
   order to get the sorted sample sequence {comment: Be-
   cause the positions of all sample classes in $W'$ keep not
   changing as shown in Fig.4(a), each sorting of $\mathbf{y}_i^*$ can
   only change the order of samples within the i-th sample
   class $W_i'\cdot$.}.
```

Among them, the first three methods are based on the methodology of ranking relevant genes; while the last two methods, i.e., CFS and RBF, do not rank genes, but aim to select a minimum gene subset with optimum feature relevance and reduced redundancy. Therefore, in the experiments, CFS and RBF only report the number of minimum gene subset discovered. We firstly used the EGRD [5], T and IG to rank the genes and compared them over different feature sizes, k=2,4,10,20,50,100,200. Each resulting feature subset was used to train an SVM classifier [6] with the linear kernel function. Because of the small number of samples, the Leave-One-Out Cross Validation (LOOCV), a popular performance validation procedure adopted by many researchers, was performed to assess the classification performance.

5.1 Leukemia Data

We used the Leukemia gene expression data [2], where besides the classes "ALL" (Acute Lymphoblastic Leukemia) and "AML" (Acute Myelogenous Leukemia), a new class of "MLL" (Mixed-Lineage or Myelogenous/Lymphoblastic Leukemia) samples was identified. It contains 12,582 genes and 72 samples with these 3 sample classes. Therefore, we performed three experiments to test our method by using one class versus the rest of classes as positive versus negative: (1) ALL versus MLL&AML, (2) MLL versus ALL&AML and (3) AML versus ALL&MLL. In each experiment, the gene expression matrix partition for our method is $W = [W_1^-, W_1^+, W_2^+]$ with one negative and two positive classes. In all three experiments, α was set to 10 for EGRD. The results are shown in Table 1, 2 and 3. As shown in the three tables, our method EGRD outperforms the other methods in,

- High Accuracy: in all three experiments, EGRD maintains very high accuracies in different k. In the experiment "MLL versus ALL&AML", where the class MLL

[5] Because EGRD can rank genes/proteins in terms of up and down regulation respectively, in this experiment of comparing k top-ranking genes/proteins, we selected $0.5k$ top-ranking genes/proteins in up regulation and $0.5k$ top-ranking genes/proteins in down regulation to form k top-ranking genes given by EGRD.

[6] The SVM*light* was used.

is hard to distinguish, EGRD can still obtain high accuracy even when k is very small.

- Compact biomarker: observing the accuracies of three methods from the small k to the large, EGRD is able to quickly obtain high accuracies even when k is small, while the methods T and IG require larger k to arrive at the same accuracy (the numbers in bold in three tables show the minimum k each method requires to get the highest accuracy). This means that EGRD outperforms the other methods in terms of discovering the compact or minimal biomarker. For example, in Table 1, the top 2 ranking genes discovered by EGRD can achieve 95.8% classification accuracy, while the accuracies of the other two methods' top 2 ranking genes are less than 80%. Similar cases also appear in Table 2 and 3.

- Stability: not only can the small number of selected genes achieve higher accuracies than the other methods, but also as k increases (more biomarkers were selected), high classification accuracies are maintained. This is a stable property with k increasing, and may be interesting to the biologists when they try to analyze more relevant genes contributing to the diseases.

Table 1: LOOCV accuracy rate (%) of ALL versus MLL&AML.

k=	*2*	*4*	*10*	*20*	*50*	*100*	*200*
T	79.2	86.1	91.7	93.1	**98.6**	98.6	98.6
IG	76.4	80.6	95.8	**98.6**	98.6	98.6	98.6
RliefF	63.9	86.1	95.8	95.8	98.6	98.6	**100**
EGRD	95.8	**100**	100	100	100	100	100
CFS: find 55 genes with 100%							
RBF: find 2 genes with 91.7%							

Table 2: LOOCV accuracy rate (%) of MLL versus ALL&AML.

k=	*2*	*4*	*10*	*20*	*50*	*100*	*200*
T	69.4	65.2	81.9	80.6	84.7	86.1	**93.1**
IG	72.2	88.9	88.9	88.9	**98.6**	98.6	97.2
RliefF	72.2	88.9	95.8	94.4	94.4	94.4	**97.2**
EGRD	84.7	91.7	97.2	98.6	**100**	98.6	98.6
CFS: find 111 genes with 100%							
RBF: find 7 genes with 87.5%							

Table 3: LOOCV accuracy rate (%) of AML versus ALL&MLL.

k=	*2*	*4*	*10*	*20*	*50*	*100*	*200*
T	66.7	77.8	97.2	98.6	**100**	98.6	97.2
IG	79.2	76.4	87.5	93.1	**97.2**	97.2	97.2
RliefF	86.1	84.7	95.8	94.4	97.2	97.2	**97.2**
EGRD	88.9	94.4	97.2	97.2	97.2	97.2	**98.6**
CFS: find 147 genes with 100%							
RBF: find 4 genes with 90.3%							

An important factor, which enables EGRD to perform well, is that the matrix reordering has the global searching ability to take into account the value distribution of the whole matrix with multiple classes. This is different from the way of individually considering genes, samples, or gene-to-gene. Our ultimate goal is to obtain the minimal biomarker while keeping a relatively high classification accuracy. In the experiment of "ALL versus MLL&AML", compact biomarker is already discovered by EGRD because, for the 4 genes selected, EGRD can achieve 100% accuracy. In the third experiment as listed in Table 3, we found 4 genes which achieve the accuracy 94.4% with EGRD. Similarly, in the third experiment, although CFS can obtain 100% accuracy, the size of the biomarker it discovers is too big (147 genes). On the contrary, our method achieves the accuracy 95.8% while the size of the biomarker is very small (only 2 genes).

To test if the biomarker found by our methods is biologically meaningful or not, for instance, we checked two genes found by EGRD in Table 1 with Entrez Gene in NCBI Website (`http://www.ncbi.nlm.nih.gov/entrez`). These two genes are MME, which is underexpressed, and LGALS1, which is overexpressed. By investigating the result of Armstrong *et al.* [2], these two genes were also ranked as the first genes in the underexpressed and overexpressed genes respectively. MME is a common acute lymphocytic leukemia antigen which is an important cell surface marker in the diagnosis of human acute lymphocytic leukemia (ALL); while LGALS1 was also reported to be highly correlated with ALL [17].

5.2 Conclusion

In this work, we have introduced a novel perspective of matrix reordering for ranking both genes and samples in multiple-class microarray data. It comprehensively considers the global between-class data distribution and local within-class data distribution, and therefore improves the accuracy of the biomarker discovery. Meanwhile, it identifies an overall tendency of the whole matrix for analyzing the data. Experiments on microarray data have demonstrated its efficiency and effectiveness of both visualization and biomarker discovery.

6. REFERENCES

[1] P. Amestoy, T. Davis, and I. Duff. An approximate minimum degree ordering algorithm. *SIAM Journal on Matrix Analysis and Applications*, 17(4):886–905, 1996.

[2] S. Armstrong and *et. al.* MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia. *Nature Genetics*, 30(1):41–47, 2002.

[3] L. E. Baum and J. A. Eagon. An inequality with applications to statistical estimation for probabilistic functions of markov processes and to a model for ecology. *Bull. Amer. Math. Soc.*, 73:360ÍC363, 1967.

[4] M. Berry, B. Hendrickson, and P. Raghavan. Sparse matrix reordering schemes for browsing hypertext. In *Proc. of the AMS-SIAM Summer Seminar on Mathematics of Numerical Analysis: Real Number Algorithms*, Park City, UT, 1995.

[5] T. Cover and J. Thomas. *Elements of Information Theory.* Wiley, New York, 2nd edition, 2006.

[6] J. Crow and M. Kimura. *An Introduction to Population Genetics Theory.* Harper & Row, New York, 1970.

[7] E. Cuthill and J. McKee. Reducing the bandwidth of sparse symmetric matrices. *Proc. of the 24th National Conference of the ACM*, 1969.

[8] C. Ding and H. Peng. Minimum redundancy feature selection from microarray gene expression data. *Journal of Bioinformatics and Computational Biology*, 3(2):185–205, 2005.

[9] D. L. Donoho. High-dimensional data analysis: The curses and blessings of dimensionality. Math Challenges of the 21st Century, August 2000.

[10] W. Gansterer and T. Korimort. Matrix reordering by hypertree decomposition. Technical Report AURORA TR2003-19, University of Vienna, 2003.

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

[12] J. Hofbauer and K. Sigmund. *The Theory of Evolution and Dynamical Systems.* Cambridge University Press, 1988.

[13] J. Hofbauer and K. Sigmund. *Evolutionary Games and Population Dynamics.* Cambridge University Press, 1998.

[14] R. Johnson and D. Wichern. *Applied Multivariate Statistical Analysis.* Prentice Hall, New Jersey, 5th edition, 2002.

[15] K. Kira and L. Rendell. A practical approach to feature selection. In *Proc. of ICML*, 1992.

[16] M. Pelillo. The dynamics of nonlinear relaxation labeling processes. *J. Math. Imaging Vision*, 7(4):309ĺC323, 1997.

[17] T. Rozovskaia and *et. al.* Expression profiles of acute lymphoblastic and myeloblastic leukemias with all-1 rearrangements. *Proc. of National Academy of Sciences USA*, 100(13):7853–7858, 2003.

[18] I. Tabus and J. Astola. *Genomic Signal Processing and Statistics*, chapter Gene Feature Selection. Hindawi Publishing Corporation, 2005.

[19] L. Yu and H. Liu. Redundancy based feature selection for microarray data. In *Proc. of SIGKDD*, pages 737–742, Seattle, 2004.

Investigating the use of Extrinsic Similarity Measures for Microarray Analysis [*]

D. Ucar, F. Altiparmak, H. Ferhatosmanoglu and S. Parthasarathy[†]
Department of Computer Science and Engineering
The Ohio State University
Columbus, Ohio
Contact : srini@cse.ohio-state.edu

ABSTRACT

Genes behaving similarly over changing conditions are believed to be part of the same functional module. Identifying functional modules of genes plays an important role in understanding gene regulatory behavior as well as in facilitating function prediction of unknown genes. Subsequently, determining 'similar' gene pairs or groups based on their gene expression profiles is an important task towards extracting modules from microarray datasets. A prevailing technique is to use a linear similarity measure like Pearson's correlation coefficient or Euclidean distance, to find similar gene pairs. However, the noise inherent in microarray datasets reduces the sensitivity of these measures and produces many spurious pairs with no real biological relevance. In this paper, we explore an extrinsic way of calculating gene similarity based on their relations with other genes. We show that 'similar' pairs identified by extrinsic measures overlap better with known biological annotations available in the Gene Ontology database. Our results also indicate that extrinsic measures are useful to enhance the quality of gene networks constructed from similar gene pairs by reducing spurious edges and introducing missing edges between network nodes.

Categories and Subject Descriptors

H.2.8 [**Database Applications**]: Data Mining

Keywords

Bioinformatics, Microarray analysis, Extrinsic similarity

1. INTRODUCTION AND RELATED WORK

Due to advances in technology (e.g., oligonucleotide microarray chips), scientists are now able to accumulate a wealth of information on the expression of genes during the life cycle of an organism. Such datasets provide vital information that can be used to gain insight into diverse biological questions. To analyze and mine these datasets for potential useful information, various techniques and ideas have been proposed. Of particular interest to many scientists is the problem of identifying gene groups that have similar expression patterns over various samples, known as co-expressed genes. Genes with similar cellular functions have been theorized to behave similarly over different conditions [10]. Thus, obtaining groups of similar genes is fundamental to understanding the molecular and biochemical processes that sustain the physiological state of the cell [23].

There has been a growing interest in representing co-expressed genes as an association network to explore the system-level functionality of genes [25, 6]. Here, nodes represent genes and two nodes are linked if the corresponding genes are significantly co-expressed (correlated) across the samples. Earlier approaches have used expression levels of two genes over all samples to surmise their correlation. However, this similarity notion does not necessarily imply that genes are functionally related. Given the noise inherent in microarray datasets, it is our hypothesis that intrinsic similarity measures are not adequate to distinguish accidentally regulated genes from those that are biologically motivated. We argue that since any given gene is likely to fluctuate in its measured expression level due to many possible sources of error, a similarity based on two genes' measurements is more error-prone than using relative positions of many genes as a reference to deduce the same information. In addition, gene products act as complexes to accomplish certain cellular level tasks [22], which is potentially suitable to infer two gene's similarity via their relations with other genes. Thus, we propose and investigate the use of extrinsic similarity measures to induce gene similarity.

The use of extrinsic measures and their advantages have been previously studied for various data mining problems [8, 9]. Das et al [8], proposed using extrinsic measures on market basket data in order to derive similarity between two products from the buying patterns of customers. Palmer et al [18], defined an extrinsic similarity measure (REP) with an analogy to electric circuits. Both groups concluded that extrinsic measures can give additional insight into the data. Recently, Ravasz et al [19], proposed the Topological Overlap Measure (TOM), which is one of the few to use extrinsic

[*] This work is supported in part by DOE Early Career Principal Investigator Award No. DE-FG02-04ER25611 and NSF CAREER Grant IIS-0347662.

[†] To whom correspondence should be addressed.

BIOKDD'07 August 12, 2007, San Jose, California,USA.
Copyright 2007 ACM 978-1-59593-839-8/07/0008 ...$5.00.

properties along with the intrinsic ones. Their measure infers similarity of two nodes in a biochemical network in terms of their pairwise similarity as well as the number of common neighbors they share.

In this paper, we introduce a methodology for the application of extrinsic similarity measures on microarray datasets. We propose two different extrinsic measures motivated by the notion of *mutual independence analysis*. The proposed similarity measures are evaluated on two well-studied cancer microarray datasets [1, 4]. In order to quantify the biological concordance of different similarity notions, we employ domain based validation metrics. We find that extrinsically similar gene pairs better overlap with known biological annotations from the Gene Ontology (GO) database when compared to the Pearson's correlation coefficient and the TOM. To further analyze their usability for gene function inference, we construct association networks from 'similar' gene pairs identified by different measures. Our analyzes show that association networks constructed based on our extrinsic measures contain less spurious and more biologically verified edges compared to their counterparts generated using other measures. We obtain densely connected clusters of genes from these networks to study their usability in understanding the molecular and biological processes that sustain health or cause cancer. We find that clusters extracted from the extrinsically similar gene networks show evidence of cancer related pathways and functional modules such as signal transduction pathway, apoptosis etc.

To summarize, our main contributions in this study are:

- Introducing the notion of *mutual independence* of two genes based on their associations with other genes
- Proposing two extrinsic similarity measures suitable for microarray analysis motivated by the *mutual independence* analysis
- Investigating and demonstrating the efficacy of using extrinsic measures in inferring pairwise gene similarities, constructing gene networks and clustering genes

2. SIMILARITY MEASURES

To quantify the resemblance of two points, one needs a measure of similarity. Similarity measures can be categorized into two: *extrinsic* and *intrinsic* similarity measures. An *intrinsic* similarity of two points i and j is purely defined in terms of the values of i and j. On the other hand, an *extrinsic* similarity measure takes into account other points to infer i and j's similarity.

Previous studies have shown the usability of external similarity measures in other domains [8, 9]. To our knowledge, usability of *extrinsic* similarity measures have not been investigated for identifying 'similar' genes. A prevailing method to infer similarity of two genes from their expression patterns is to use a linear *intrinsic* similarity (e.g. Euclidean distance, Pearson's correlation coefficient) measure. We discuss *intrinsic* similarity measures next.

2.1 Intrinsic Measure

Intrinsic similarity is purely defined on the points in question. In the context of microarray analysis, the *intrinsic* similarity of two genes is defined on these genes' expression levels over all samples.

In a typical microarray experiment, each gene is expressed at some certain level at each condition which is defined as the gene's expression profile. More formally, a gene (say, x) is associated with a profile vector (V_x) composed of its expression values over all samples, such that $V_x = [x_1, x_2, ..., x_n]$, where n denotes the number of samples in the dataset. Thus, *intrinsic* similarity between genes x and y, is a measure defined on their profile vectors, V_x and V_y.

The most commonly used and accepted measure in the literature for the task at hand is the Pearson's correlation coefficient. This is defined as [16]:

$$r_{xy} = \frac{\sum_{i=1}^{n}(V_x^i - \overline{V_x})(V_y^i - \overline{V_y})}{\sqrt{\sum_{i=1}^{n}(V_x^i - \overline{V_x})^2 \sum_{i=1}^{n}(V_y^i - \overline{V_y})^2}} \tag{1}$$

where $\overline{V_x}$ and $\overline{V_y}$ are the profile averages. Here, V_x^i represents the i^{th} entry of the vector V_x. According to this definition, genes which are positively (or negatively) correlated have a value close to 1 (or -1) whereas dissimilar gene pairs have values close to 0. Absolute value of Pearson's correlation scores is used in this study since both positive and negative correlations can play an important role in gene association.

2.2 Extrinsic Measures

Extrinsic similarity of two attributes (i.e., genes) is defined over other attributes in the dataset. Before defining its specifics, a general definition of an *extrinsic* measure is as follows [8]:

$$ES_P(i,j) = \sum_{k \in P} |f(i,k) - f(j,k)| \tag{2}$$

Here, $f(i,k)$ denotes a function that signifies association between i and k. P refers to the set of attributes that will contribute to the *extrinsic* similarity calculation of attributes i and j.

As noted by Das et al [8], proper choice of the attribute set P and function f is crucial for the usefulness of the resulting *extrinsic* measure. Different choices will result in different similarity notions. In the following section we will discuss a methodology to derive effectual *extrinsic* similarity measures to be used in inferring gene similarity.

2.3 Proposed Methodology

Our goal in developing an *extrinsic* similarity for microarray analysis is to surmise the similarity of two genes by the similarity of their relation with other genes. We believe that use of an extrinsic measure for microarray analysis has a twofold advantage over the use of intrinsic measures. First, it reduces the impact of noise inherent in the dataset on the similarity inference since more evidence are taken into consideration per inference. Second, it suits well with the biological hypothesis that genes act as complexes to accomplish certain tasks in the cell. As hypothesized, two genes behaving similarly with the elements of a gene complex, presumably belongs to that complex and share their functionality. Thus defining two genes' similarity by taking into consideration their relation with other genes can potentially benefit from the modular structure of the genomic interactions.

To define a proper measure, we first need to determine over which set of genes, P, and using which association function, f, *extrinsic* similarity of two genes should be defined.

Here, we investigate the use of close proximity of genes according to *intrinsic* notions when choosing a proper set P. In addition, two functions based on *mutual independence analysis* from the Information Theory are evaluated. We compare the proposed similarity measures with the currently available techniques described in Section 3, as well as the most popular *intrinsic* measure (i.e., Pearson's correlation coefficient).

2.3.1 Choice of Attribute Set (P)

To derive an efficient *extrinsic* measure for microarray analysis, we first need to identify a gene set, P, that will be used to infer the *extrinsic* similarity of two genes. For this purpose, we use the group of genes that are similar to both of the genes under question. Thus, initially for each gene we identify a set of genes that are intrinsically similar to that gene (i.e., the gene's close neighbors). We refer this as a gene's neighborhood list (N_i) and define it as follows:

$$N_i = \{j | j \in G, |r_{ij}| > \kappa\} \tag{3}$$

Here, G denotes the set of all genes in our dataset and $|r_{ij}|$ refers to the absolute value of the Pearson's correlation coefficient of genes i and j. Effect of the threshold parameter κ, on the *extrinsic* measures and guidance of the size of neighborhood lists to set this parameter is discussed in Section 6[1]. Next, the attribute set P that will be used to infer two genes' similarity is designated as the intersection of their neighborhood lists (i.e., $P = N_i \cap N_j$). Using common neighbors of two genes as the set of attributes (P) has two important implications. First, it significantly reduces the required number of calculations. Thus, instead of using the whole gene set (G), a smaller size set is taken into consideration. Secondly, it filters out irrelevant information which improves the success of the *extrinsic* measure. By using the *intrinsic* similarity to determine elements in set P, we take advantage of both *extrinsic* and *intrinsic* properties. Our hypothesis is that this helps to reduce the noisy inference that can be introduced into the similarity inference by using these measures separately. It is noteworthy that an extrinsic measure can be easily expandable to other groups of related genes. For instance, one can prefer using an attribute set containing genes mapped to close chromosomal locations with two genes whose similarity is under investigation.

2.3.2 Choice of Association Function (f)

After establishing the notion of an *extrinsic* similarity, and defining the set P, the next step is to determine which association function (f) to use for our calculations. Das et al [8], proposed using the *confidence* of association rules in an application on market basket dataset. Their approach and its applicability on gene expression datasets will be discussed in details in Section 3. We propose using two appropriate functions that are motivated by the *mutual independence analysis.* We leverage mutual independence of two genes by analyzing their frequency of occurrence and co-occurrence in the neighborhood lists.

Before defining mutual dependency of two genes, first, we explore three possible type of relations between any two genes motivated by Das et al [8]. Accordingly, two genes can either be, *complementary*, *independent* or *correlated.* If two genes are *complementary*, then they do not to co-occur in the neighborhood lists. If they are *independent*, neighbors of gene i are neighbors of gene j with the same probability as the genes that are not neighbors of gene i. And if they are *correlated*, neighbors of gene i are also neighbors of gene j. These concepts are formally defined using neighborhood lists as follows:

Definition 1: *Frequency of occurrence* for a gene i, $P(i)$, is defined as the frequency of encountering that gene in all neighborhood lists. Since Pearson's correlation coefficient is a symmetric measure a gene has as many neighbors as the number of times it occurs in all neighborhood lists. Thus, frequency of a gene's occurrence can be simplified to the following:

$$P(i) = \frac{|N_i|}{|G|} \tag{4}$$

where '|ů|' denotes the number of elements (cardinality) in its argument. Note that *frequency of occurrence* is an indication of the discriminatory nature of a gene's expression profile. Genes with indistinct expression profiles such as the housekeeping genes will have higher values of *frequency of occurrence.*

Definition 2: *Frequency of co-occurrence* for genes i and j, $P(i, j)$, is defined as the frequency of encountering these two genes together in the neighborhood lists. More formally, based on the symmetric Pearson's measure, $P(i, j)$ can be defined as follows:

$$P(i,j) = \frac{|\{a | a \in G, i \in N_a, j \in N_a\}|}{|G|} \tag{5}$$

By itself high *frequency of co-occurrence* does not imply that two genes are *correlated.* In order to conclude that two genes are not randomly co-occurring (*independent*) but there is a biological trigger behind their co-occurrence (*correlated*), we need to test if one gene's *frequency of occurrence* is helpful in predicting that of the other gene which is a notion known as mutual independence. Note that, in this context, independence of two genes implies that occurrence of a gene in a neighborhood list makes it neither more nor less probable for the other gene to occur in that list. Thus, mutual independence of two genes only holds when $P(i, j) = P(i)P(j)$. We propose using two different independence tests to leverage *mutual dependency* of two genes.

Specific Mutual Information Measure:
The Specific Mutual Information (*smi*) is a measure of association commonly used in the Information Theory to infer mutual dependency. *Smi* of two variables, X and Y, given their joint distribution, $P(X, Y)$, and individual distributions, $P(X)$ and $P(Y)$, is defined as follows:

$$I(X,Y) = \frac{O}{E} = \frac{P(X,Y)}{P(X)P(Y)} \tag{6}$$

where $P(X, Y)$ is the observed value (O) for joint probability of events X and Y, whereas $P(X)P(Y)$ is its expected value (E).

This test can be used to deduce the type of relation between two genes. If their *smi* value is 1, it can be concluded that these two genes are *independent.* On the other hand, a value greater than 1 implies being *correlated* and a value smaller than 1 implies being *complementary.*

[1] Our analysis indicated that relatively loose values produce more useful *extrinsic* measures.

If two genes have similar relations with their common neighbors, it is reasonable to conclude that they are similar. Based on this analysis and the notion of specific mutual information, we propose the following *extrinsic* measure to quantify dissimilarity of two genes (i and j).

$$smi_P(i,j) = \frac{\sum_{k \in P} \left| \frac{P(i,k)}{P(i)P(k)} - \frac{P(j,k)}{P(j)P(k)} \right|}{|P|} \qquad (7)$$

This definition ensures that two genes having similar relations (i.e., *complementary*, *correlated* or *independent*) with their common neighbors are closely related to each other (*smi* value close to 0). Whereas two genes that have different relations with their common neighbors are dissimilar and associated with higher values of *smi*. Note that, the *smi* measure is normalized by dividing by the size of the attribute set P.

Chi-Square Based Measure:
Pearson's chi-square test is another method to assess *mutual dependency* of two events. Formally, it is defined as follows:

$$chi(X,Y) = \frac{(O-E)^2}{E} = \frac{(P(X,Y) - P(X)P(Y))^2}{P(X)P(Y)} \qquad (8)$$

This test tells us how far the observed value deviates from the expected value under the assumption of independence.

According to this definition, two genes will have zero *chi* value if they are *independent*. They will have higher *chi* values otherwise. We employ a signed version of this test to surmise the type of relation between two genes. Given this, external dissimilarity of two genes based on the chi-square analysis, $chi_P(i,j)$, is defined as follows:

$$\frac{\sum_{k \in P} \left| \frac{s_{ik}(P(i,k)-P(i)P(k))^2}{P(i)P(k)} - \frac{s_{jk}(P(j,k)-P(j)P(k))^2}{P(j)P(k)} \right|}{|P|} \qquad (9)$$

where s_{ab} denotes the sign of the term $P(a,b) - P(a)P(b)$. Note that signs are included into the measure to differentiate a *correlated* pair from a *complementary* one. Similar to the *smi* measure, two genes that have similar relations with their common neighbors will have smaller *chi* values whereas two genes that have dissimilar relations with their common neighbors will have higher values[2]. *Chi* measure is also normalized by dividing by the size of the attribute set.

3. PREVIOUS WORK

3.1 Topological Overlap Measure

Recently, Ravasz et al [19], proposed the Topological Overlap Measure (TOM) which takes into a step in using *extrinsic* measures to infer similarity between two nodes of a biological network. This measure is considered as an improvement over the *intrinsic* similarity which amalgamates an additional external knowledge derived from the network topology (i.e., number of common neighbors). According to their definition, two nodes have high topological overlap if they are connected to roughly the same group of nodes. More formally, TOM of two genes i and j can be expressed as follows:

$$TOM(i,j) = \frac{|N_i \cap N_j| + r_{ij}}{min\{|N_i|, |N_j|\} + 1 - r_{ij}} \qquad (10)$$

where r_{ij} is the pairwise similarity between these two genes. The inclusion of the *intrinsic* similarity (r_{ij}), into this definition, makes TOM measure explicitly dependent on the *intrinsic* similarity of two nodes in question. Drawbacks of this dependency will be discussed in Section 6.

3.2 Confidence of Association Rules

Das et al [8, 9], previously studied the *extrinsic* similarity of attributes in a market basket dataset where *confidence* of association rules are used as the association function, f. In a market-basket problem, each customer fills their market basket with a subset of large number of items (e.g., bread, milk). Such datasets are mined for association rules of the form $(X_1, ..., X_n \Rightarrow Y)$ to identify the relation between items. The *confidence* of an association rule is defined as the frequency of encountering the head of the rule $(X_1, ..., X_n)$ among all the groups containing the body (Y). Das et al [8], proposed using the *confidence* of association rules as the association function f. Thus, their proposed *extrinsic* similarity measure reduces to the following.

$$ES_P(A,B) = \sum_{D \in P} |conf(A \Rightarrow D) - conf(B \Rightarrow D)| \qquad (11)$$

where $conf(A \Rightarrow D)$ is defined as $\frac{P(A,D)}{P(A)}$.

For the task at hand, an analogy to a market basket is a neighborhood list. Accordingly, we use the *frequency of occurrence* $(P(i))$ and the *frequency of co-occurrence* $(P(i,j))$ to derive a corresponding *confidence* based *extrinsic* measure suitable for microarray analysis. We again normalize this measure by dividing it by the size of the set P.

We compare the newly proposed *extrinsic* similarity measures (*smi* and *chi*) with the existing ideas in the literature (i.e., TOM and *confidence*) as well as the most commonly used and accepted intrinsic measure for microarray analysis, namely the Pearson's correlation coefficient.

4. DOMAIN BASED EVALUATION

'Similar' pairs identified according to different similarity measures are evaluated based on the Pairwise Semantic Similarity measure of Resnik [17]. This measure makes use of known annotations in the Gene Ontology (GO) database. GO is a controlled vocabulary designed to accumulate the result of all investigations in the area of genomic and biomedicine by providing a large database of known associations.

Biological relevance of two genes can be quantified with respect to the significance of their shared GO annotations using the Semantic Similarity (SS) measure defined by Resnik [17]. Resnik's measure is preferred among other semantic similarity measures [11, 12], since it has been shown to outperform the others and suit better for use in GO [20].

Pairwise SS scores are used to infer functional relevance of probe pairs. For this purpose, we plot SS values for all annotated pairs of the arrays under study and observe that for both arrays SS values roughly follow normal distributions. We believe that to reduce the impact of missing information in GO database, it is desirable to limit ourselves to upper and lower tail of the distribution for inference. Accordingly, we label each pair as a 'TP' if their SS score is greater than the 95^{th} percentile of all pairwise SS values. Similarly, a pair is accepted as a 'FP' when their SS value is smaller than the 5^{th} percentile of the distribution. We run an analysis to test the effect of using greater percentile cut-offs on the overall

[2] Only the positive information is considered for the chi square test.

results which is presented in the Experiments section. We want to note that, not every gene pair will be classified as a 'TP' or a 'FP' using this labeling methodology. A pair that is composed of at least one unannotated gene is not labeled since there is not enough information to conclude about the biological concordance of these two genes. In addition, a gene pair with an SS score between the percentile cut-offs is not labeled since considering it as a 'TP' or a 'FP' pair is a matter of specifying the granularity of biological similarity.

Pairs extracted by using different similarity notions are accumulated into association networks. We define the Clusterwise Positive Predictive Value measure ($CPPV$) to evaluate the biological quality of the dense regions extracted from these clusters. $CPPV$ of a cluster, (say, C_i), is defined as $CPPV_i = \frac{|TP_i|}{|TP_i|+|FP_i|}$. Here, TP_i and FP_i denote the set of 'TP' and 'FP' pairs in that cluster. Our calculations are based on every possible gene pair in a cluster. Higher values of $CPPV$ imply that the cluster is enriched in 'TP' pairs. On the contrary, lower values indicate that the cluster is composed of biologically dissimilar genes.

5. DATASETS AND PRE-PROCESSING

For this study, we employ two well-studied cancer datasets. First dataset is composed of gene expression values of 62 colon tissue samples where the Affymetrix Hum6000 array with 6819 probes is used [1]. 42 of these are collected from colon adenocarcinoma patients and 20 of them are collected from normal colon tissue of the patients. Among all probes, 2000 were selected from 6817 by Alon et al according to the highest minimum intensity [1]. Second dataset is composed of 86 lung adenocarcinoma and 10 normal samples which is analyzed by the Affymetrix HuGene FL array [4]. Beer et al [4] trimmed the dataset of genes expressed at extremely low levels resulting in 4966 probes for investigation.

Initially, we consider 2000 and 4966 probes for colon and lung adenocarcinoma datasets respectively. We perform thresholding, log transformation and normalization (quantile normalization) on these two datasets as suggested by our analysis. In addition to these, we further standardize datasets using a robust standardization method, median absolute deviation (MAD). Genes with zero MAD values implying that they are co-expressed at very similar levels across all of the samples are excluded from further analysis. After preprocessing 1578 genes for colon cancer and 4228 genes for lung cancer datasets are examined.

6. EXPERIMENTS

We discuss the usability of external similarity measures as a way of identifying similar genes throughout this section. First, we give results for biological relevance of gene pairs that are identified as 'similar' with different measures. Then, co-expression networks generated from these 'similar' pairs are analyzed for biological soundness. Finally, genes in each of these networks are clustered to study the effect of *extrinsic* similarity on the quality of gene clustering.

6.1 Setting the κ parameter

Before comparing newly proposed measures with the existing ones, we first investigate the effect of κ parameter on the neighborhood lists. To choose a suitable κ threshold, there are two things that we should take into consideration. First, we want a gene's neighborhood list to be composed only of genes that are within close proximity of that gene. Second, it is not desirable to have a set that is only composed of a few genes since this would limit the power of inference based on common neighbors. Accordingly, we vary κ parameter between 0.3 and 0.9 and observe the average size of neighborhood lists for each of these values. As expected, for both datasets, smaller values of κ resulted in lists bigger in size with many dissimilar genes. On the other hand, higher κ values resulted in very small size lists which are very restrictive to draw any conclusions. Given that observation, we believe that average size of the neighborhood lists can guide us for setting the κ parameter. Consequently, a reasonable κ threshold value, 0.5, is determined for both datasets where neighborhood lists contain around 40 genes. We test the effect of κ parameter on the efficacy of extrinsic similarity measures in the next section.

6.2 Effect on Top 'Similar' Pairs

In the first experiment, we compare gene pairs that are labeled as 'similar' according to the discussed measures. For each measure, gene pairs are sorted starting from the most 'similar' one. These pairs are labeled as 'TP' or 'FP's based on their semantic similarity scores[3]. Different number of top scoring pairs (varying between 1000 and 10000) are compared based on the number of 'FP' and 'TP's among them (depicted in the below table) [4].

	Pearson		TOM		Confidence		Smi		Chi	
	TP	FP	TP	FP	TP	FP	TP	FP	TP	FP
1000	24	48	24	48	35	47	34	34	**47**	**25**
2000	51	88	50	87	65	107	72	**64**	**75**	65
3000	74	133	75	134	**111**	140	**111**	99	100	**94**
4000	109	176	109	177	140	201	**153**	136	132	**122**
5000	153	219	154	220	170	243	**195**	180	168	**150**
6000	193	265	194	265	187	309	**224**	222	204	**178**
7000	226	322	225	321	236	352	**268**	256	242	**214**
8000	265	365	265	366	265	380	**296**	285	294	**252**
9000	297	403	299	405	304	422	328	315	**330**	**283**
10000	337	445	338	447	330	464	361	343	**366**	**305**

In each case, *smi* and *chi* measures produce more 'TP' pairs compared to the TOM and the Pearson measures. In addition, *smi* and *chi* measures also generate significantly less 'FP' pairs in comparison to other measures. These results confirm that *smi* and *chi* measures better capture the biological relevance of two genes than the available measures in the literature. This improvement can be attributed to two reasons: the noisy nature of microarray datasets and the functional modularity of genes. *Intrinsic* measures directly possess and reflect the noise inherent in the data since they are purely defined on the expression levels of genes under study. As high values of 'FP' counts for the Pearson measure imply, erroneous measurements have a drastic impact on this *intrinsic* measure. It is notable that despite taking into consideration an *extrinsic* feature, TOM is similarly affected by the noise inherent in the dataset. This result shows that TOM is mainly dominated by the *intrinsic* factor in its definition. On the other hand, *extrinsic* measures are dependent on more evidence where mutual independence is inferred from all neighborhood lists. As a result, impact of erroneous measurements expected to be less severe on the *extrinsic* similarity measures. Our experimental results

[3] Not every gene pair can be labeled as a 'TP' or a 'FP'.

[4] Colon cancer dataset follows similar trends.

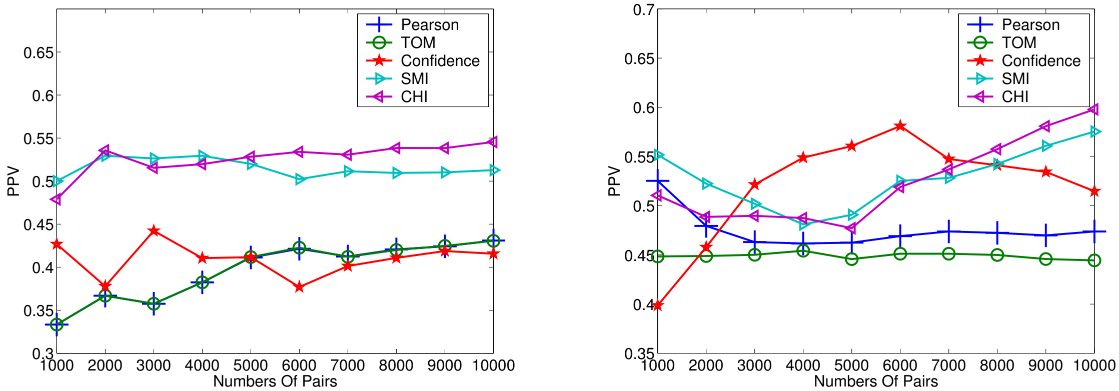


Figure 1: PPV of the top 'similar' pairs identified from our experimental datasets ($\kappa = 0.5$): (a)Colon Cancer (b)Lung Cancer.

are also in accordance with this expectation where extrinsic measures generate less 'FP' pairs. In addition, inferring two genes' similarity from a set of other genes can benefit from the group level interactions known to take place between gene products when accomplishing certain cellular tasks [22]. High 'TP' counts associated with *extrinsic* measures are also in accordance with this biological premise. Poor results of the *confidence* measure indicate that choosing a proper association function f is also vital when defining an *extrinsic* similarity measure.

We also evaluate the Positive Predictive Value (PPV $= \frac{TP}{TP+FP}$) of these pairs on both datasets (presented in Figures 1a-b). As can be seen, for both datasets, *smi* and *chi* measures constantly have higher PPVs when same number of similar pairs are analyzed. For colon cancer dataset, when compared to Pearson correlation, on average *smi* and *chi* measures improved the PPVs 30% and 34% respectively. For the lung cancer dataset, *smi* and *chi* measures again produce higher PPVs (on average an increase by 11% and 10%) than the Pearson measure. On the other hand, for both datasets TOM does equivalently or poorly when compared to the Pearson measure. Our analyzes also show that *confidence* is not a robust similarity measure due to the fact that it only considers two genes co-occurrence without analyzing their independence. As a result, it is impossible to tell if two genes are *correlated*, *independent* or *complementary* based on their *confidence* scores. This leads to incorrect conclusions about two gene's similarity as implied by the fluctuating pattern of the *confidence* measure in Figures 2a-b.These results also suggest that *mutual independence* based analysis generates more robust external similarity measures when compared to the *confidence* based analysis.

In the next experiment, we evaluate the PPV of top pairs for different values of κ. We re-run our analysis on colon cancer dataset for different κ thresholds (depicted in Figure 1a ($\kappa = 0.5$) and Figures 2a-b ($\kappa = 0.45$ and $\kappa = 0.55$)). In each case, pairs identified by our *extrinsic* measures have systematically higher PPVs than the other measures. As in the previous cases, confidence measure produces inconstant PPVs and TOM does equally well with the Pearson correlation. These results show that although κ threshold has an impact on the efficacy of extrinsic measures, within a reasonable range (can be chosen by considering the average size of neighborhood lists) of κ values, *extrinsic* measures would be better alternatives to *intrinsic* measures.

6.3 Effect on Similarity Networks

In this experiment, we construct association networks by connecting the top scoring gene pairs identified by each measure. To keep the same size for all networks, we only used the top 0.01% of 'similar' gene pairs in each case. Accordingly, from the colon cancer dataset a network of 12,438 edges and from the lung cancer dataset a network composed of 89,359 edges are constructed. To investigate the biological quality of these networks, we identify the 'TP' and 'FP' pairs (i.e., edges) in each network. Here, we again observe that the advantage of using extrinsic measures is two-fold as shown in the below table. First, they reduce the number of 'FP' edges and secondly they increase the number of 'TP' edges. As a result, for the colon cancer dataset PPV is increased by 18% and 20% when *smi* and *chi* measures are employed respectively. For the lung cancer dataset, both measures improve the PPV by 15 % when compared to the Pearson measure. Networks identified using the TOM, do not have higher PPVs than the networks generated by the Pearson correlation, implying that TOM fails to contribute to a standard intrinsic similarity measure. These results suggest that extrinsic measures are not only effective in reducing the false inferences, but they also introduce certified edges missed by the existing similarity measures. Given this, we believe that well-suited *extrinsic* measures, can give additional insight into the gene similarity networks which cannot be captured by an *intrinsic* measure.

	Colon Cancer			Lung Cancer		
	TP	FP	PPV	TP	FP	PPV
Pearson	427	548	0.44	3571	4027	0.47
TOM	420	539	0.44	2913	4125	0.41
Confidence	409	583	0.41	2881	3719	0.44
Smi	445	419	0.52	**4494**	3814	**0.54**
Chi	**449**	**395**	**0.53**	4309	**3702**	**0.54**

We also evaluate the effect of using different percentile cut-offs that are used to infer 'TP' and 'FP' pairs. For this purpose, we re-analyze the gene network generated from the colon cancer dataset by varying the percentile cut-offs. We vary upper tail percentile cut-offs between 0.05, 0.1 and 0.2

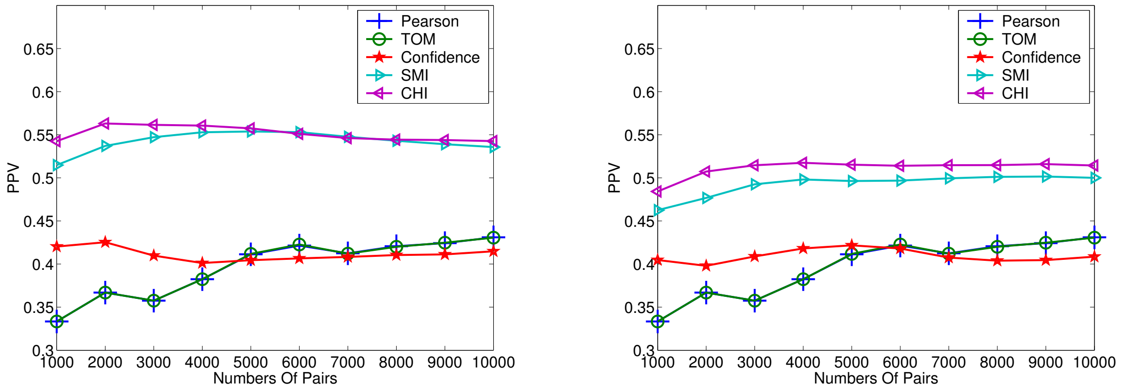


Figure 2: PPV of the top 'similar' gene pairs identified from Colon cancer dataset for different values of κ (a)0.45 and (b)0.55.

and correspondingly lower tail cut-offs between 0.95, 0.9 and 0.8. We then analyze the PPV of colon cancer 'similarity' networks using these varying cut-offs (depicted in Figure 3). As can be seen from this figure, although changing the cut-offs effect the mere value of PPVs, networks generated from *extrinsic* measures do consistently better than their intrinsic counterparts for any cut-off setting. However, we also note that when a wider (lower and upper) tail is considered for our analysis, the improvement of *extrinsic* measures over *intrinsic* measures decreases. For example when we compare *smi* measure with Pearson, the increase in PPV decreases from 18% to 12% when the 20^{th} (and 80^{th}) percentile is used instead of the 5^{th} (and 95^{th}) percentile. This can be attributed to the existence of missing information in the GO database. As expected, inference based on wider tails are more severely affected by the partial information than the inference based on extreme tails.

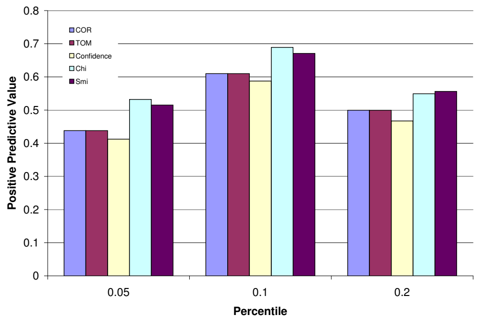


Figure 3: Evaluation of colon cancer network for various percentile cut-offs.

6.4 Effect on Network Clusters

In this experiment, we examine the quality of clusters extracted from different gene similarity networks. Extracting groups of genes that are tightly connected in a co-expression network is important for the inference of functional annotation [10, 21, 3]. However, it is not yet clear which clustering/partitioning method is the most useful one for this purpose. To identify dense regions from our networks, we employ the most commonly used clustering algorithm, i.e., hierarchical clustering with UPGMA. To our knowledge, no entirely reliable method exists for identifying the correct number of clusters (i.e., k) in a dataset. That is why, we perform hierarchical clustering for a range of different numbers of clusters ($100 \leq k \leq 1000$). Modularity measure proposed by Newman et al [14] is used to estimate the correct number of clusters for each network. As suggested by the modularity analysis, colon and lung cancer networks are initially partitioned into 500 and 400 clusters respectively. Each clustering arrangement is validated using the cluster validation measure ($CPPV$). We then eliminate the clusters with zero $CPPV$ values and plot $CPPV$ of the remaining ones (depicted in Figures 4a-b). As can be observed from these figures, *smi* and *chi* networks produce more clusters with high $CPPV$ values for both datasets. These results confirm that networks generated based on external similarity notions are better sources for obtaining biologically more meaningful clusters.

We next investigate the importance of identifying biologically sound groupings for reaching a better understanding of cancer and consequently developing new treatments.

7. DISCUSSION

In this section, we investigate the usability of clusters extracted from different gene similarity networks by running a dataset specific analysis. For this part of our analysis, we make use of the colon cancer dataset which is composed of tumorous and non-tumorous tissues of the human colon and rectum. As being the third most common cancer and the second leading cause of cancer-related death in US, a better understanding of the development and progression of this disease can be crucial for determining novel targets and strategies for its treatment.

Our experimental results show that by using *extrinsic* similarity notions, we obtain clusters with higher $CPPV$ implying pairwise similarities of genes in the same cluster. However, pairwise similarities do not prove that the cluster is composed of many genes that are involved in the same pathway or molecular function. We further analyze the extracted clusters to investigate the ones that are functionally coherent. For this purpose, we employ an enrichment analysis that signifies the statistical value of a cluster's functional homogeneity. We calculate an enrichment score (i.e., p-value) which is defined as the chance of observing that particu-

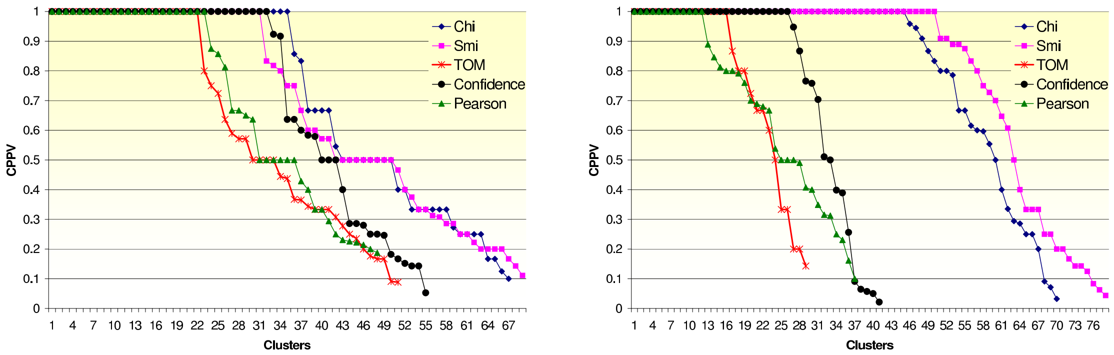


Figure 4: Distribution of $CPPV$ for clusters extracted from (a) Colon cancer ($k = 500$) and (b) Lung cancer datasets ($k = 400$).

lar grouping, or better, given the background distribution[5]. Among all clusters, the ones that are significantly enriched in genes from the same functional group are determined and presented in the following table. Recommended cut-off of 0.05 is used for all our validations. A more detailed analysis of these significant clusters is revealed that they can be very useful in understanding and treating the colorectal cancer. We discuss several of these clusters and their relation with colon cancer in the rest of this section.

Several of the clusters extracted from the *chi* network, are annotated with the GO terms related to the *Signal Transduction Pathway* (i.e., *receptor signaling protein activity, signal transducer activity, scavenger receptor activity*). This is an important pathway targeted for colorectal cancer treatment [7]. Thus, studying these clusters might be important for understanding the role of signal transduction in colorectal cancer, and accordingly introducing promising molecular targets, and strengthening the existing therapeutic approaches. An additional use of these clusters might be to understand the interactions between various functional groups that initiate and maintain colorectal cancer. One can study the edges between clusters in order to reveal this information. Other measures cannot disclose the biological signal regarding the role of *Signal Transduction Pathway* in colon cancer from our test data.

From the *smi* network, we extract a cluster that is composed of genes associated with the GO term *cytoskeleton*. Recent evidence indicates that the interaction of a tumor suppressor gene (APC) with the cytoskeleton might contribute to colorectal tumor initiation and progression [15]. That is why, we believe that locating these genes together in a cluster is triggered by the role they play in colon cancer tumorigenesis. Unfortunately, it is still unknown that how APC interacts with the cytoskeleton and how their interaction plays a role in the formation of colorectal tumors [15]. We believe that once functionally coherent (and less error-prone) clusters are identified, relations between these clusters can be used to reveal the function level interactions vital for understanding the cause of some diseases.

Besides revealing pathways and functional groups associated with the colon cancer, significant clusters can also be employed for function prediction. Determining the functions of genes is a central problem in biology [21, 5, 13]. An unannotated gene that is located into a cluster with a significant functional annotation can be predicted to be part of this same functional module. Our hypothesis is that clusters that are functionally more coherent are better sources for function prediction. As an example, one of the *smi* clusters is associated with the GO term *tRNA metabolism*. In this group, a gene (H05910) does not have a known annotation. This suggests that the unknown gene might have an unrevealed task in this biological process. Using other similarity measures the same gene is located into clusters that are not enriched in any functional gene groups which provides no information for function prediction and identification.

GO Term	Measure	p-value
receptor signaling protein activity	*Chi*	.000291
signal transducer activity	*Chi*	.000091
scavenger receptor activity	*Chi*	.000278
immunological synapse	*Chi*	.000590
Ras GTPase binding	*Chi*	.000209
phosphoprotein binding	*Chi*	.000160
mRNA metabolism	*Chi*	.000480
protein homooligomerization	*Chi*	.000217
regulation of metabolism	*Chi*	.000049
positive regulation of I-kappaB kinase/NF-kappaB cascade	*Chi*	.000062
secretion	*Chi*	.000250
general RNA polymerase II transcription factor activity	*Smi*	.000761
phosphatase regulator activity	*Smi*	.000965
secretory granule	*Smi*	.000309
leading edge	*Smi*	.000189
non-membrane-bound organelle	*Smi*	.000359
cytoskeleton	*Smi*	.000453
cation channel activity	*Smi*	.000096
DNA-directed RNA polymerase activity	*Smi*	.000603
hematopoietin/interferon-class cytokine receptor activity	*Smi*	.000965
FAD binding	*Smi*	.000774
translation initiation factor activity	Pearson	.000500
synaptic transmission	Pearson	.000031
obsolete molecular function	Pearson	.000283
synaptic transmission	TOM	.000030
protein N-terminus binding	TOM	.000217
acetyl-CoA C-acyltransferase activity	Conf.	.000279
helicase activity	Conf.	.000025
golgi apparatus	Conf.	.000339

[5] All three ontologies are employed. For more details please refer to our previous work [24].

8. CONCLUSION

In this paper, we have introduced the notion of *mutual independence* of genes based on their relations with their common neighbors. We have presented suitable *extrinsic* similarity measures for microarray analysis that make use of the *mutual independence analysis*. We have investigated the efficacy of the proposed measures and run thorough analysis to compare them with other measures available in the literature. Our experimental results prove that using the *extrinsic* measures it is possible to identify gene pairs that are biologically more relevant. In addition, association networks generated based on these measures are shown to contain more 'TP' edges and less 'FP' edges.

Our analysis also shows that different similarity notions can reveal different aspects of a microarray dataset as implied by the diverse annotations extracted from different networks. Previously, we have studied different ensemble techniques to improve clustering results on a scale-free protein interaction network [2]. We believe that an ensemble approach in integrating different aspects of a dataset captured by different similarity measures could work well in microarray analysis. In the future, we plan to investigate this. As an extension, we would also like to work on characterizing the group level interactions among genes and gene products using the multivariate information analysis.

9. ACKNOWLEDGMENTS

The authors would like to thank Sitaram Asur for his valuable comments.

10. REFERENCES

[1] U. Alon and et al. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl. Acad*, 96:6745–6750, 1999.

[2] S. Asur, D.Ucar, and S. Parthasarathy. An ensemble framework for clustering protein-protein interaction networks. *In Proc. 15th Annual Int'l Conference on Intelligent Systems for Molecular Biology (ISMB)*, 2007.

[3] G. Bader and C. Hogue. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics*, 4:2, 2003.

[4] D. Beer and et al. Gene-expression profiles predict survival of patients with lung adenocarcinoma. *Nature Medicine*, 9:816, 2002.

[5] A. Butte and I. Kohane. Mutual information relevance networks: Functional genomic clustering using pairwise entropy measurements. *Pac Symp Biocomput*, 5:418–429, 2000.

[6] S. Carter, C. Brechbhler, M. Griffin, and A. T. Bond. Gene co-expression network topology provides a framework for molecular characterization of cellular state. *Bioinformatics*, 20:14:2242–2250, 2004.

[7] S. J. Cohen, R. B. Cohen, and N. J. Meropol. Targeting signal transduction pathways in colorectal cancer-more than skin deep. *Journal of Clinical Oncology*, 23:5374–5385, 2005.

[8] G. Das, H. Mannila, and P. Ronkainen. Similarity of attributes by external probes. *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98)*, pages 23–29, 1998.

[9] G. Das, H. Mannila, and P. Ronkainen. Similarity of attributes by external probes. *Report C-1997-66, University of Helsinki, Department of Computer Science*, October 1997.

[10] M. Eisen, P. Spellman, P. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci*, 95:25:14863–14868, 1998.

[11] J. Jiang and D. Conrath. Semantic similarity based on corpus statistics and lexical taxonomy. *In Proc. Int'l Conf. Research in Computational Linguistics*, ROCKLING X, 1997.

[12] D. Lin. An information-theoretic definition of similarity. *In Proc. 15th Int'l Conf. Machine Learning*, 1998.

[13] T. Murali, C. Wu, and S. Kasif. The art of gene function prediction. *Nature Biotechnology*, 24:1474–1475, 2006.

[14] M. Newman and M. Girvan. Finding and evaluating community structure in networks. *Physical Review E*, 69:026113, 2004.

[15] I. Näthke. Cytoskeleton out of the cupboard: colon cancer and cytoskeletal changes induced by loss of apc. *Nature Reviews Cancer 6*, pages 967–974, 2006.

[16] B. Ostel. Statistics in research basic concepts and techniques for research workers. *Iowa State University Press, Ames, Iowa, USA*, 1963.

[17] R. P. Using information content to evaluate semantic similarity in a taxonomy. *In Proceedings of the 14th International Joint Conference on Artificial Intelligence*, 1:448–453, 1995.

[18] C. Palmer and C. Faloutsos. Electricity based external similarity of categorical attributes. *7th Pacific-Asia Conference on Knowledge Discovery and Data Mining(PAKDD 2003)*, 2003.

[19] E. Ravasz and et al. Hierarchical organization of modularity in metabolic networks. *Science*, 297:5586:1551–1555, 2002.

[20] J. L. Sevilla and et al. Correlation between gene expression and go semantic similarity. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 2:4, 2005.

[21] B. Snel, P. Bork, and M. Huynen. The identification of functional modules from the genomic association of genes. *Proc Natl Acad Sci*, 99:5890–5895, 2002.

[22] V. Spirin and L. A. Mirny. Protein complexes and functional modules in molecular networks. *PNAS*, 100:21, 2003.

[23] J. Stuart, E. Segal, D. Koller, and S. Kim. A gene coexpression network for global discovery of conserved genetic modules. *Science*, 302:5643:249–255, 2003.

[24] D. Ucar, S. Asur, U. V. Catalyurek, and S. Parthasarathy. Improving functional modularity in protein-protein interactions graphs using hub-induced subgraphs. *PKDD*, pages 371–382, 2006.

[25] B. Zhang and S. Horvath. A general framework for weighted gene co-expression network analysis. *Statistical Applications in Genetics and Molecular Biology*, 4:1, 2005.

Mining Over-Represented 3D Patterns of Secondary Structures in Proteins

Matteo Comin*
*Department of Information Engineering
University of Padova, Italy
Padova 35131, Italy
ciompin@dei.unipd.it

Concettina Guerra*†
†College of Computing
Georgia Institute of Technology
Atlanta, GA 30332-0280 USA
guerra@dei.unipd.it

Giuseppe Zanotti‡
‡Department of Chemistry and VIMM
University of Padova
Padova 35131, Italy
giuseppe.zanotti@unipd.it

ABSTRACT

We consider the problem of finding over-represented arrangements of Secondary Structure Elements (SSEs) in a given dataset of representative protein structures. While most papers in the literature study the distribution of geometrical properties, in particular angles and distances, between pairs of interacting SSEs, in this paper we focus on the distribution of angles of all quartets of SSEs and on the extraction of over-represented angular patterns. We propose a variant of the Apriori method that obtains over-represented arrangements of quartets of SSEs by combining arrangements of triplets of SSEs. This specific case will pose the basis for a natural extension of the problem to any given number of SSEs. We analyze the results of our method on a dataset of 300 non redundant proteins.

1. INTRODUCTION

The problem of finding recurrent three-dimensional patterns in proteomic data is of biological interest and therefore has been studied in different contexts and with various techniques [6, 16]. In fact, although the information on the fold of a protein is already totally contained in its amino acid sequence, the calculation of the minimal energy among all the possible conformations is a task which is overwhelming even for the fastest computer. For this reason, a great deal of efforts has been spent over the years in order to disclose hidden rules about the organization of secondary structure elements [2, 8].

A simplified description of the three-dimensional protein structure is that of considering it as an arrangement of SSEs. The possible ways SSEs aggregate in space is someway limited: all protein structures, till now determined, can be grouped in a relatively limited number of different folds. Moreover, it is well known that interacting SSEs show marked preferences in their reciprocal orientation. For example, interacting β-strands are very often organized in sheets, where each strand is disposed in a roughly parallel or antiparallel orientation with respect to the neighboring ones [3]. Preferences between interacting α-helices have been also studied extensively and general rules extracted [4, 7, 15]. Nevertheless, it has been shown that the expected uniform random distribution of angles is actually biased toward angles near 90^o[1]. When this geometric bias was taken into account, the observed peaks in the helix-helix angle distribution were significantly attenuated: correcting for statistical bias, the true preference for particular packing angles in soluble proteins is not as strong as previously thought.

Moreover, the relative arrangement of non-interacting SSEs in space is less obvious [11]. In order to analyze their global disposition, in the past we have conducted a statistical analysis on the occurrences of triplets of SSEs [10, 17]. We found that the distribution is far from being random, with a marked preference for specific angle combinations. This knowledge could be used to guide the engineering of stable protein modules or to predict the three-dimensional structure [13].

The present study extends the previous analysis, taking into account quartets of SSEs. It presents an analysis of the distribution of secondary structures within a selected set of non redundant proteins. It constructs frequent patterns of k elements (or itemsets of size k) by joining frequent patterns of size $k-1$.

2. PROBLEM DESCRIPTION

Given a data-set of proteins structures, we address the problem of finding over-represented arrangements of SSEs in terms of geometrical properties. Most papers in the literature study the distribution of geometrical properties, in particular angles, between pairs of interacting SSEs [14, 18]. Here we focus on over-represented configurations consisting of more than two SSEs and analyze the distribution of angles of such configurations. Our task is to design a framework to extract over-represented arrangements of k SSEs, by combining the results obtained with arrangements of $k-1$ SSEs. We discuss in details how to obtain over-represented arrangements of four SSEs by using the distribution of triplets of SSEs instead of generating all quartets of SSEs from the data set. This specific case will pose the basis for a natural extension of the problem to any given number of SSEs.

Each protein structure of the dataset is given with the list of SSEs ordered according to the backbone chain. A line segment is associated to each SSE. For a β-strand the segment

BIOKDD'07, August 12, 2007, San Jose, California, USA.
Copyright 2007 ACM 978-1-59593-839-8/07/0008 ...$5.00.

is the best fit segment of the set of atoms of the strand, for an α-helix it is the best fit axis. For the purpose of our analysis, a line segment is assumed to be a unit vector applied in the origin of a reference system in three-dimensional space. Thus a protein is a list of m unit vectors $(s_1, \cdots, s_m)$.

An arrangement of SSEs is described in terms of the angles formed by all pairs of corresponding vectors. Let α_{hk} be the dihedral angle of s_h and s_k, $0^o \leq \alpha_{hk} \leq 180^o$. A triplet of SSEs (s_{i1}, s_{i2}, s_{i3}), with $i1 < i2 < i3$, is described by three angles α_{12}, α_{13} and α_{23} satisfying the triangle inequality. A quartet of SSEs $S = (s_{i1}, s_{i2}, s_{i3}, s_{i4})$, with $i1 < i2 < i3 < i4$, gives rise to 6 dihedral angles $Q = (\alpha_{12}, \alpha_{13}, \alpha_{23}, \alpha_{24}, \alpha_{34}, \alpha_{14})$. A schematic representation of the unit vectors derived from a quartet of SSEs can be found in Figure 1. It is easy to show that, in the general case, the six angles are not completely independent. More precisely, given 5 of the α_{hk} angles, the sixth angle can take only one of two possible values. The derivation of such values is omitted for lack of space. Furthermore, when three out of four segments are mutually orthogonal then one of the angles formed by the fourth segment with the three segments is uniquely determined by the other two angles. Another important question, that will be considered in section 4, is whether it is possible to superimpose, by a rigid transformation, two quartets forming the same angles.

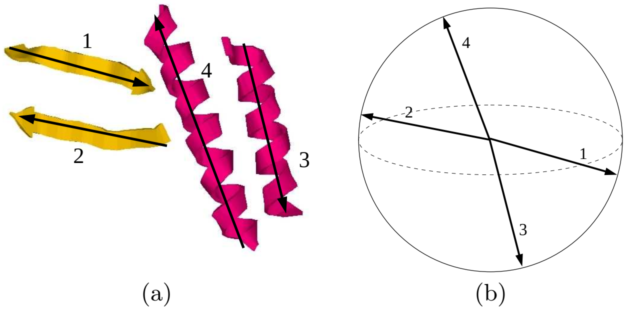


Figure 1: (a) An example of vector discretization for a quartet of SSEs. (b) The unit vectors translated to the origin (into the unit sphere).

The angular values are discretized into uniform intervals, with every interval represented by an integer. More precisely, in our work the range $0^o - 180^o$ is divided into 10 intervals, and an angle α represented by the integer i such that $i * 18^o \leq \alpha < (i + 1) * 18^o$. Thus a quartet of SSEs is represented by 6 integer values each in the range [0,10]. In the following we refer to the discretized angles simply as angles.

3. DISCOVERY OF OVER-REPRESENTED PATTERNS

Our approach is similar to the Apriori algorithm used for data mining applications. Apriori finds frequent associations of attributes of k elements (or itemsets of size k) by joining frequent associations of itemsets of size $k - 1$. Similarly, our algorithm finds over-represented arrangements of quartets of segments from over-represented triplets of segments; it does so by joining over-represented triplets of angles to obtain over-represented sextuplets of angles.

However, our approach differs substantially from Apriori in the way the patterns are joined together to obtain patterns of larger size. At the basis of the Apriori mining algorithm is the anti-monotone property that states that all non empty subsets of a frequent set must also be frequent. In other words, if an itemset cannot pass the test of being frequent, then all its supersets will fail the same test.

The anti-monotone property does not hold for the angles formed by sets of segments. Consider a frequent sextuple of angles $Q = (\alpha_{12}, \alpha_{13}, \alpha_{23}, \alpha_{24}, \alpha_{34}, \alpha_{14})$ and all quartets S of segments with angles Q. Even though Q is frequent, it is possible that triplets that are subsets of Q are not frequent. This is the case of the triplet of angles $T = (\alpha_{13}, \alpha_{23}, \alpha_{24})$ that cannot be formed (in the general case) by a triplet of segments which is a subset of an element of S, because the three angles involve all 4 segments of a single element of S. However, there are four triplets of angles subsets of a frequent sextuple Q that must be frequent. These are $(\alpha_{12}, \alpha_{13}, \alpha_{23})$ and $(\alpha_{23}, \alpha_{24}, \alpha_{34})$, $(\alpha_{13}, \alpha_{14}, \alpha_{34})$ and $(\alpha_{12}, \alpha_{14}, \alpha_{24})$. Indeed, the four triplets are formed by the four different ways of choosing three segments out of four. Frequent triplets of angles are extracted by comparing the observed frequencies of triplets of angles with those of randomly distributed vectors.

We now describe our mining procedure. We start by giving an overview of our approach, and then describe each step in detail.

PROCEDURE: *Pattern Discovery*

1. Initialization: From the given protein data set generate the set A of all ordered triplets of angles associated to ordered triplets of SSEs, sorted according to the order along the backbone.

2. Build an hash table indexed by the triplets of angles that stores all triplets of segments.

   Derive the 3D histogram of the distribution of the triplets of A from the hash table. The histogram has $b = 10$ bins along each axis, for a total of b^3 bins or cells.

3. Build the distribution of triplets of angles of random unit vectors and derive the corresponding 3D histogram.

4. Based on the deviation between the histogram of observed triplets of angles and that of random triplets, determine the subset $C \subset A$ of triplets that are over-represented.

5. Join step: construct candidate sextuples of angles from triplets of C.

6. Verification step: prune candidate sextuples to find the over-represented ones.

3.1 Building the Hash Table

We build a four-dimensional hash table with the following index structure: for a given triplet of vectors, three indexes are given by the quantized values of the angles of the triplet, the fourth index depends on the composition of the triplet in terms of the number and position of helices and strands. This index, called *triplet type*, is used when a separate analysis is requested for helices and strands. The size of the cells of the table is the same as the binsize for the histograms.

Each cell of the table contains a list of records, one for every triplet that hashed into it. The following procedure inserts protein P into the hash table and is a variant of the one described in [5].

PROCEDURE: *Insert Protein*
Given protein P, all triplets of secondary structures of P are examined and for each triplet (p_u, p_v, p_z) with $u < v < z$ the following steps are executed:

i. Compute the angles $(\alpha_{uv}, \alpha_{vz}, \alpha_{uz})$ and determine *triplet type.*

ii. Access the cell of the hash table at the location indexed by *triplet type* and by the quantized values of $(\alpha_{uv}, \alpha_{vz}, \alpha_{uz})$.

iii. Append to the list of records at that cell a new record that contains:

- the name of protein P.
- the identifier of each secondary structure element of the triplet.

The above procedure is repeated for all proteins in the data set. The construction of the table is computationally intensive. However, the number of proteins of the dataset to be inserted is relatively small.

3.2 Generating Random Triplets

The selection of the frequent triplets is the crucial point of the overall procedure: a wrong selection can produce a meaningless starting point that can lead to unreliable results. Thus this step must be carefully designed. We observe that the distribution of geometric properties of triplets strongly depends on the features considered. To avoid the bias due to the features considered, we compute the null distribution of such properties.

The random generation of a triplet of angles is decomposed into the generation of three versors. A versor is a vector of unit length that we assume to be in the semisphere identified by a positive value of the z coordinate. A versor is now uniquely determined by two parameters: its coordinate $z \in [0, 1]$, and its Azimuth $\beta \in [0, 2\pi]$. We have already observed that the triangular inequality holds for any three angles α, β, γ of a triplet of segments; it translates into the following three constraints: $\alpha + \beta \geq \gamma$, $\alpha + \gamma \geq \beta$, $\beta + \gamma \geq \alpha$. This implies that not all cells of the hash table can be populated by triplets of segments; in other words, there are cells that will remain empty. Furthermore, some cells can only be partially populated. Thus when deciding which cells correspond to most frequent triplets of angles, we have to take into account the above consideration and normalize by the volume of the region of the cell that can in fact be populated. This region is determined by considering that the above three constraints correspond to the equations of the three boundary planes $\alpha + \beta = \gamma$, $\alpha + \gamma = \beta$, $\beta + \gamma = \alpha$ delimiting the populated area in 3D space. By intersecting each cell of the 3D array with the three boundary planes we find out which region, if any, has to be excluded and consequently compute the volume V_c of the populated region. Thus the frequency of a cell (α, β, γ) will be: $Count(\alpha, \beta, \gamma)/V_c(\alpha, \beta, \gamma)$.

Given a data set of n real proteins to analyze, we generate the distribution of angles of n sets of random vectors, each corresponding to a protein of the dataset and containing the same number of SSEs of such protein.

The generation of the ensemble of random vectors is repeated several times and, at the end, each cell of the hash table has the average of the values of the cell over all random generations. This results in a 3D histogram representing all triplets of angles, where each triplet has attached a mean and a variance. For the selection of over-represented angles we experimented with different selection policies. To preserve a reasonable number of candidates we select the configurations of angles that have a frequency above the mean.

3.3 Join and Verification Steps

The operation *join* merges four frequent triplets (α_{12}, α_{13}, α_{23}) and $(\alpha_{23}, \alpha_{24}, \alpha_{34})$, $(\alpha_{13}, \alpha_{14}, \alpha_{34})$ and $(\alpha_{12}, \alpha_{14}, \alpha_{24})$ into the candidate sextuple $(\alpha_{12}, \alpha_{13}, \alpha_{23}, \alpha_{24}, \alpha_{34}, \alpha_{14})$. The four triplets to be merged are such that the last angle of the first triplet is the same as the first angle of the second; the second element of the first triplet is the same as the first element of the third triplet, and so on. Recall that all angles are discretized. Furthermore, note that two triplets may coincide.

Once a candidate sextuple has been identified in step 5, the verification procedure checks that there is in fact a statistically significant number of quartets of vectors with that sextuple of angles. This number will provide the actual frequency of the sextuple of angles. The verification step is needed because some triplets of segments contributing to the count of frequent triplets of angles cannot be joined into quartets of segments. For instance, the two triplets might be from different proteins. Two triplets of segments (s_1, s_2, s_3) and (t_1, t_2, t_3) associated to SSEs of the same protein and forming angles $(\alpha_{12}, \alpha_{13}, \alpha_{23})$ and $(\alpha_{23}, \alpha_{24}, \alpha_{34})$, respectively, can be joined into a quartet of segments with angles $(\alpha_{12}, \alpha_{13}, \alpha_{23}, \alpha_{24}, \alpha_{34}, \alpha_{14})$ if $(s_2 = t_1$ and $s_3 = t_2)$, i.e. the last two segments of the first triples coincide with the first two of the second triples. Two such triplets of segments are called “consistent” and they contribute one to the frequency count of the associated sextuple.

To efficiently search for consistent triplets, we use the hash table built in step 2 containing the triplets of segments of all proteins. The frequency or count of a candidate sextuple $(\alpha_{12}, \alpha_{13}, \alpha_{23}, \alpha_{24}, \alpha_{34}, \alpha_{14})$ is determined as follows. Access the hash table at the cells $E1$ and $E2$ indexed by $(\alpha_{12}, \alpha_{13}, \alpha_{23})$ and by $(\alpha_{23}, \alpha_{24}, \alpha_{34})$ respectively. For each triplet (s_1, s_2, s_3) in $E1$ with associated protein name P search in $E2$ for all triplets (s_2, s_3, t), with any arbitrary t, of the same protein P. For each such triplet increment the count if the last angle α_{14} is compatible with the candidate sextuple under examination.

4. SPATIAL ARRANGEMENTS OF VECTORS WITH THE SAME ANGULAR PATTERN

It is interesting to determine whether two sets of vectors with the same angular pattern can be superimposed by a 3D rigid transformation, or whether the spatial conformations of the two sets of vectors differ in their 3D shape. Protein structure comparison algorithms that align SSEs also use a shape similarity measure based on the rigid superposition of the structures [21].

We define equivalent two sets of vectors that can be superimposed by a rigid transformation. We first look at the case of triplets of vectors (a, b, c) and their angles (α, β, γ). We recall that the unit vectors are applied into the origin O of a coordinate system without considering the actual location of the SSE in 3D space. It is easy to see that there are two distinct triplets of vectors (a, b, c) and (a, b, c'), where c and c' are non parallel vectors, forming a given triplet of angles (α, β, γ). For example (see Figure 2), consider four vectors forming a regular pyramid with vertex in 0; label two opposite vectors of the pyramid a and b and the other two c and c'. The two triplets of vectors (a, b, c) and (a, b, c') have the same angles but are non equivalent since they are one the mirror of the other.

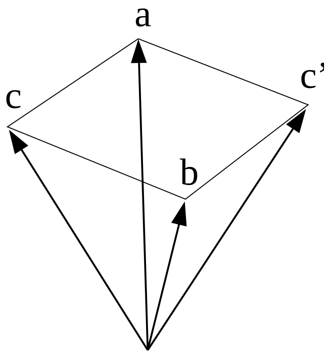


Figure 2: An example of two triplets, (a, b, c) and (a, b, c'), with the same pairwise angles, one the mirror of the other.

Perhaps more convincing is the following proof. All vectors forming a given angle δ with a given vector v are rays of the cone with vertex in O and forming δ angle with v. Given two vectors a and b forming angle α, a third vector forming angles β and γ with a and b, respectively, is at the intersection of two cones. Two cones intersect at either one or two lines. In the first case, the only possible triplet consists of vectors lying on the same plane $(\alpha + \beta = \gamma)$; in the latter there are two non parallel vectors c and c' corresponding to two distinct triplets.

In conclusion, a triplet of angles (α, β, γ) corresponds to two spatial arrangements of unit vectors (a, b, c) and (a, b, c') that are one the mirror of the other; equivalently, there exists a transformation with determinant -1 mapping one triplet of vectors into the other. Loosely speaking, although two triplets of vectors cannot be superimposed by a rotation (with determinant 1), they correspond to a similar configuration in terms of angles.

If we extend this argument to quartets of vectors, the number of non equivalent arrangements doubles. Consider a sextuple of angles $(\alpha_{12}, \alpha_{13}, \alpha_{23}, \alpha_{24}, \alpha_{34}, \alpha_{14})$. To construct all non equivalent quartets of vectors corresponding to it, we follow a build-up approach. From the first three angles $(\alpha_{12}, \alpha_{13}, \alpha_{23})$ we construct either one triplet of vectors (a, b, c) or two (a, b, c) and (a, b, c'). Then, we derive the last vector d. There are four possible cases:

1. If $\alpha_{12} + \alpha_{23} = \alpha_{13}$ and $\alpha_{23} + \alpha_{34} = \alpha_{24}$, then there is a single triplet (a, b, c) and a single triplet (b, c, d). Thus, there exists a unique arrangement of four vectors.

2. If $\alpha_{12} + \alpha_{23} = \alpha_{13}$ but $\alpha_{23} + \alpha_{34} < \alpha_{24}$, then two distinct arrangements are possible, (a, b, c, d) and (a, b, c, d').

3. Otherwise, if $\alpha_{23} = \alpha_{34}$ then four different arrangements are possible, with three distinct vectors as last component of the quartet: (a, b, c, d), (a, b, c, d'), (a, b, c', d') and (a, b, c', d'').

4. In all other cases, the following four arrangements are possible: (a, b, c, d), (a, b, c, d'),(a, b, c', d'') and (a, b, c', d''').

5. RESULTS AND DISCUSSION

We selected a set of 300 non-redundant proteins from different families and computed the set of all triplets of SSEs and their associated linear segments. To include only significant SSEs, we required helices to have at least seven residues, corresponding to two complete turns of a regular helix. Strands were required to have at least three residues for proper fitting of a vector to the C_α coordinates. Secondary structures are represented by the best-fit line segments. A Singular-Value Decomposition (SVD) routine is used to associate a segment to each α-helix and β-strand [9]. Using this dataset we constructed the hash table of triplets of angles and compared it with the random distribution to determine the cells that deviate significantly from the corresponding cells for the random data. The hash table contains 520 non empty cells (containing a total of 398,853 triplets of vectors), of which 242 were selected as frequent (corresponding to 189,270 triplets). The histogram of the triplets of angles selected as frequents is shown in Figure 3.

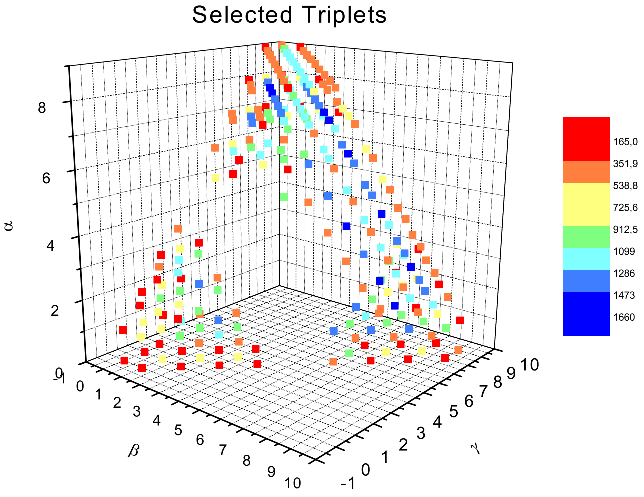


Figure 3: 3D histogram of the distribution of selected angles. Each axis represents an angle and the frequency of each triplet follows the color coding.

5.1 Analyzing Over-represented Patterns of Angles

The pattern discovery process finds a set of over-represented arrangements of four SSEs. Each arrangement is described by six ordered angles, i.e. an angle corresponds to a specific pair of SSEs which is identified by the sequential order of SSEs along the primary structure. Thus two arrangements forming the same six angles, but in a different order, correspond to two different patterns, even though they can be considered geometrically equivalent. We address this issue by merging together patterns composed by the same angles and ignoring the relative order of angles.

By merging patterns, the discovery procedure selects a set of 785 over-represented patterns, formed by 485,021 quartets of segments, out of 2,262 patterns and more than 3,000,000 quartets obtained by the exhaustive search. The top pattern is composed by the discretized angles $(1, 2, 3, 7, 8, 9)$, corresponding to angles in the ranges $(18^o - 36^o, 36^o - 54^o, 54^o - 72^o, 126^o - 144^o, 144^o - 162^o, 162^o - 180^o)$, and has a frequency of 6,439, the top second has similar angles, (1,2,7,8, 8,9), and a smaller frequency of 5,780. The frequency count drops dramatically after the first few patterns. It is interesting to notice that the top 11 angular patterns (out of 785) cover about 10% of the quartets; coverage of the quartets of about 20% is obtained by 29 patterns and that of 50% by 122 patterns.

The overall discovery procedure is relatively fast; it takes approximately 20 minutes on a standard PC (AMD Athon 2.6 GHz). On the same machine, the exhaustive generation of all possible quartets of SSEs takes more than 3 days.

We observed that over-represented patterns of angles tend to form clusters in the six-dimensional space corresponding to six angles. Thus, we further analyzed the set of over-represented patterns by clustering them using as distance the Euclidean distance between angular patterns in six-dimensional space.

We experimented with different clustering algorithms and different numbers of clusters and, based on the measure of silhouette [12], we selected the k-means algorithm with 3 clusters. Clusters 1 and 3 contain, respectively, the first and second most frequent pattern. Cluster 2 contains the configuration of angles $(0, 1, 1, 2, 2, 3)$ that appears at position 16 in the overall ranking of patterns. The top patterns for each cluster are shown in Figure 4. In Figure 5 the cluster separation is highlighted by plotting the distribution of distances between the centroids of each cluster and the elements of all 3 clusters.

In all clusters the angles vary from 0^o to 72^o and from 126^o to 180^o, while values between 80^o and 100^o are completely absent. This is not surprising because the distribution is biased by the presence of many interacting SSEs. For example, in parallel and anti-parallel β-sheets, each β-strand typically forms a small angle with the two nearby strands. The same is true for interacting α-helices, that pack forming small angles; furthermore, they are hardly found perpendicular to each other [19, 20]. Cluster 2 is the smallest one, with 32,988 elements; it contains SSEs characterized by the same orientation: in fact, the angles between all pairs of SSEs are in the range 0^o to 72^o. The other two clusters are more densely populated; cluster 1 has 221,879 elements and cluster 3 has 230,154 elements. In these two clusters the SSEs are arranged with three SSEs with the same orientation and the other one with the opposite (cluster 1) or with two SSEs in the same orientation and the other two in the opposite orientation. The smaller number of elements in cluster 2 reflects the tendency of SSEs that are close in space to form anti-parallel configurations.

If we restrict the analysis to homogenous configurations, i.e. those containing four strands or four helices, we obtain similar results for the clusters, but with a preference for anti-parallel pairs, corresponding to the top ranked pattern of angles $(1, 2, 7, 8, 8, 9)$.

α_0	α_1	α_2	α_3	α_4	α_5	Frequency
1	2	3	7	8	9	6,439
1	2	3	7	8	8	5,586
1	1	2	7	8	9	4,657
1	2	3	6	8	9	4,085
1	2	3	7	7	8	3,728
1	1	2	6	7	8	3,648
1	2	2	7	8	9	3,401
1	2	3	6	7	9	2,958
1	1	2	8	8	9	2,833
1	1	2	7	8	8	2,494

Cluster 1

α_0	α_1	α_2	α_3	α_4	α_5	Frequency
0	1	1	2	2	3	2,623
1	1	1	2	2	3	2,162
0	1	1	1	2	2	2,123
0	1	1	2	3	3	1,667
0	1	1	2	2	2	1,445
0	1	1	1	1	2	1,311
0	1	1	1	2	3	1,246
0	1	2	2	3	3	1,178
1	1	1	2	3	3	1,039
1	1	2	2	2	3	1,010

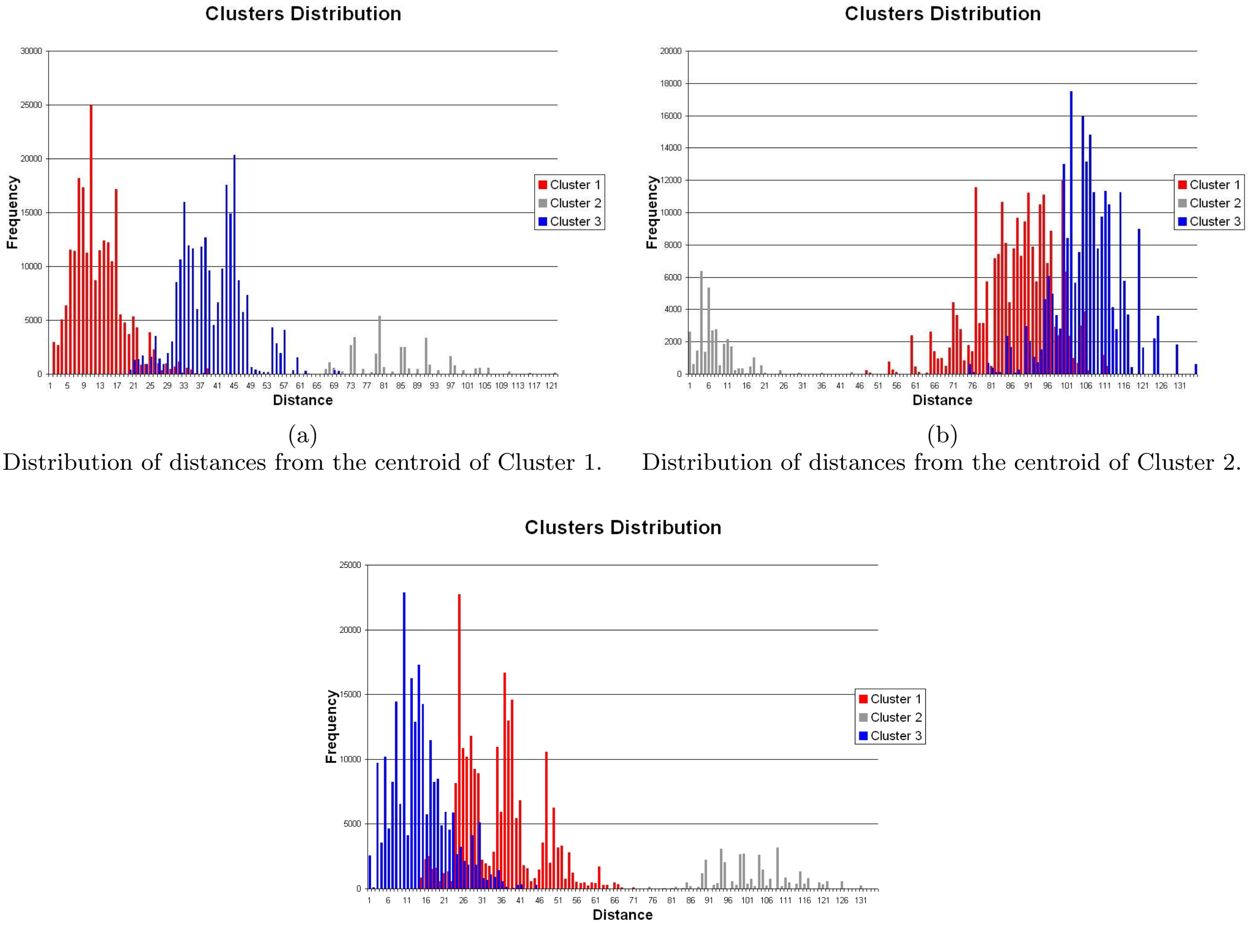
Cluster 2

α_0	α_1	α_2	α_3	α_4	α_5	Frequency
1	2	7	8	8	9	5,780
1	3	6	7	8	9	5,100
1	2	6	7	8	9	4,437
2	3	6	7	8	9	3,884
1	3	7	7	8	8	3,831
1	2	7	7	8	9	3,637
1	1	7	8	8	9	2,916
1	3	6	7	8	8	2,572
1	3	7	7	8	9	2,544
0	3	7	7	8	8	2,525

Cluster 3

Figure 4: The ten top frequent patterns for the three clusters.

The over-represented patterns considered so far have included the SSEs of the selected set of proteins, regardless of their distances. We now consider homogenous patterns of SSEs that are close in space; we define two SSEs to be in contact if the distance between the mid-points of their associated vectors is less than a given threshold (18 in our analysis). Figure 6 shows the number of pairs of vectors in contact for the top configuration. It is interesting to notice that in all cases at least one pair of vectors is in contact, and very often three or more vectors are in contact. Notice that the use of the same threshold penalizes helices, because of their bigger steric hindrance [18]. Nevertheless, more than 65% of the elements have at least two SSEs in contact. To better appreciate the proximity of these over-represented configurations, in Figure 7 we show different examples of four strands, with angles $(1, 2, 7, 8, 8, 9)$. In all these examples the four strands are in contact. Although they display different arrangements, their pairwise angles are similar, thus they fall into the same cell of the hash table. These patterns of angles are obtained with SSEs from the same β-sheet (Figure 7(c)), as well as from different β-sheets (Figure 7(a) and (b)). The fact that most, but not all,

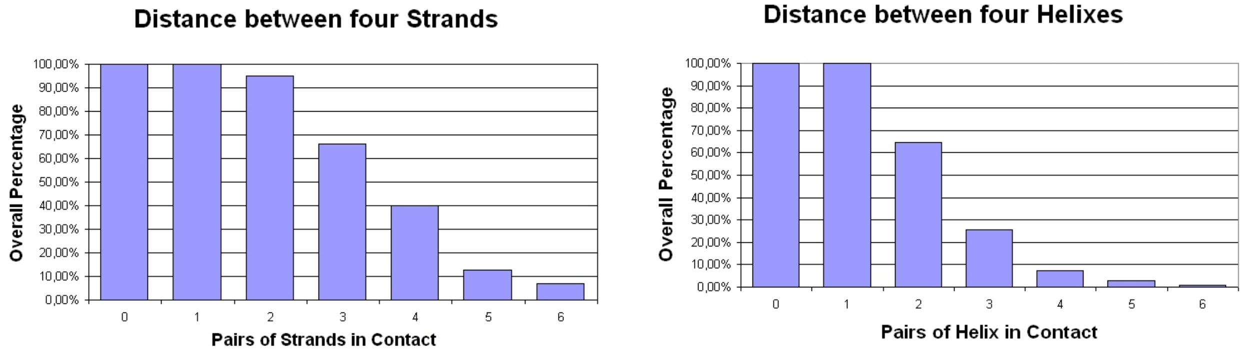


(a)
Distribution of distances from the centroid of Cluster 1.

(b)
Distribution of distances from the centroid of Cluster 2.

(c)
Distribution of distances from the centroid Cluster 3.

Figure 5: Distance distributions between centroids of clusters.



(a) Number of pairs in contact in quartets of strands.

(b) Number of pairs in contact in quartets of helixes.

Figure 6: Number of pairs of segments in contact.

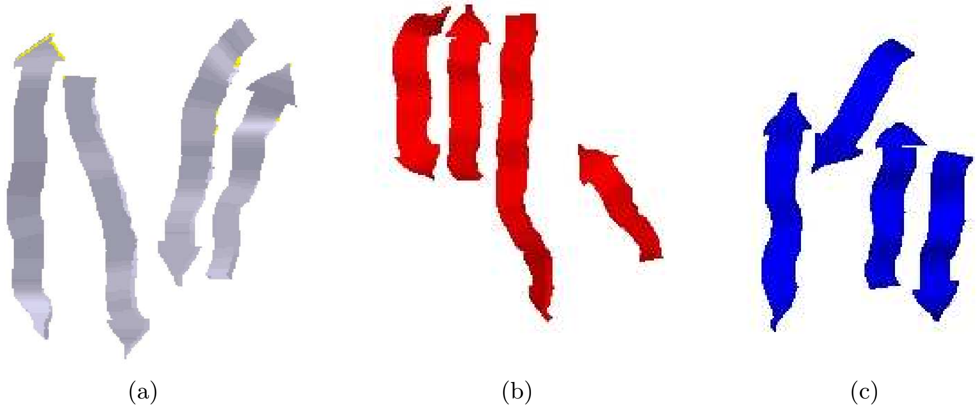


Figure 7: Three examples of the pattern of angles (1,2,7,8,8,9) composed by all strands: (a) Protein 1hpl, SSE: 16-17-18-20; (b) Protein 1acc, SSE: 0-1-2-3; (c) Protein 1aor, SSE: 4-6-8-12.

SSEs are close in space consolidates the idea that arrangements of angles are influenced by atomic interactions, either directly or through other SSEs that do not explicitly belong to the quartet. Finally, as illustrated in Figure 7, secondary structure elements belonging to the same quartet do not necessarily correspond to similar structures, i.e. structures that can be superimposed by rotation and translation. For this reason it is impossible to associate a three-dimensional motif, or a group of motifs, to the most frequent quartets described above. The biological significance of the distributions observed needs a deepener investigation.

6. CONCLUSIONS

We have proposed an efficient algorithm to extract over-represented quartets of SSEs, that avoids the exhaustive generation of patterns. We have shown that a careful analysis of the angular bias of random vectors is essential in the determination of over-represented arrangements of secondary structures. This study provides a generalized framework that can be easily extended to patterns composed by more than four SSEs. The knowledge of over-represented patterns could be used to guide the engineering of stable protein modules or to predict their three-dimensional structures. Other applications can be designed by replacing the null distribution with that of a specific family of proteins.

7. REFERENCES

[1] Bowie J.U. Helix packing angle preferences. *Natural Structural Biology*, 4:915-917, 1997.

[2] Brenner, S.A. Predicting the conformation of proteins from sequences. Progress and future progress. *J. Mol. Recognit.*, 8(1-2):9-28, 1995.

[3] Chothia, C., Levitt, M. and Richardson, D. Structure of proteins: packing of α-helices and pleated sheets. *Proc. Natl. Acad. Sci.*, USA 74, 4130-4134, 1977.

[4] Chothia, C., Levitt, M. and Richardson, D. Helix to helix packing in proteins. *J. Mol. Biol.*, 145:215-250, 1981.

[5] Comin M., Guerra C., Zanotti G. PROuST: a comparison method of three-dimensional structures of proteins using indexing techniques. *J. Comput. Biol.*, 11:1061-1072, 2004.

[6] Efimov, A.V. Structural trees for protein superfamilies. *Proteins*, 28(2):241-60, 1997.

[7] Efimov, A.V. Complementary packing of alpha-helices in proteins. *FEBS Lett.* , 463(1-2):3-6, 1999.

[8] Eisenhaber, F., Persson, B. and Argos, P. Protein structure prediction: recognition of primary, secondary, and tertiary structural features from amino acid sequence. *Crit. Rev. Biochem. Mol. Biol.*, 30(1):1-94, 1995.

[9] Gerstein, M. A Resolution-Sensitive Procedure for Comparing Protein Surfaces and its Application to the Comparison of Antigen-Combining Sites. *Acta Cryst.* , A48, 271-276, 1992.

[10] Guerra C., Lonardi S., and Zanotti G. 3D Matching of Proteins based on Secondary Structures. *Proc. IEEE Symposium on 3DPVT*, Padova, pages 812-821, 2002.

[11] Hespenheide BM, Kuhn LA. Discovery of a significant, nontopological preference for antiparallel alignment of helices with parallel regions in sheets. *1: Protein Sci.*12(5):1119-1125, 2003.

[12] Kaufman, L., Rousseeuw, P. J. (1990). Finding Groups in Data - An Introduction to Cluster Analysis. *Wiley Series in Probability and Mathematical Statistics*, 1990.

[13] Laskowski R.A., MacArthur M.W., Moss D.S., Thornton J.M. PROCHECK: a program to check the stereochemical quality of protein structures. *J. Appl. Cryst.*, 26:283-291, 1993.

[14] Lee S, Chirikjian GS. Interhelical angle and distance preferences in globular proteins. *Biophys J.*, 86(2):1105-1117, 2004.

[15] Nakamura, H. Roles of electrostatic interaction in proteins. *Q. Rev. Biophys.*, 29(1):1-90, 1996.

[16] Persson, B. Bioinformatics in protein analysis. *EXS.*, 88:215-31, 2000.

[17] Platt D.E., Guerra C., Zanotti G., Rigoutsos I. Global secondary structure packing angle bias in proteins. *Proteins: Structure, Function, and Genetics*, 53(2):252-261, 2003.

[18] Reddy, B., and Blundell, T. Packing of secondary structural elements in proteins. Analysis and prediction of inter-helix distances. *J. Mol. Biol.*, 233:464-479, 2003.

[19] Walther, D., Eisenhaber, F. and Argos, P. Principles of helix-helix packing in proteins: the helical lattice superposition model. *J. Mol. Biol.*, 255:536-553, 1996.

[20] Walther D, Springer C, Cohen FE. Helix-helix packing angle preferences for finite helix axes. *Proteins*, 33(4):457-9, 1998.

[21] Yona G. and Kedem K. The URMS-RMS hybrid algorithm for fast and sensitive local protein structure alignment. *Journal of Computational Biology*, 12:12-32, 2005.

Combining Domain Fusions and Domain-Domain Interactions to Predict Protein-Protein Interactions

Thanh Phuong Nguyen
School of Knowledge Science
Japan Advanced Institute of Science and Technology
Ishikawa, Japan
phuong@jaist.ac.jp

Tu Bao Ho
School of Knowledge Science
Japan Advanced Institute of Science and Technology
Ishikawa, Japan
bao@jaist.ac.jp

ABSTRACT

Protein-protein interactions are intrinsic to almost all cellular processes. Protein domains, the fundamental units in a protein are key elements to mediate such interactions. Whereas domain-based methods to predict protein-protein interactions often used only protein domain information; protein-protein interactions, in fact, are also associated with the biological nature of each interacting partner. Integrating both protein domain features and genomic/proteomic features of interacting partners is expected to better predict protein-protein interactions, and to discover reciprocal biological relationships among protein-protein interactions, protein domains, and genomic/proteomic features related to protein-protein interactions.

We present a novel integrative domain-based approach for predicting protein-protein interactions (PPI) using inductive logic programming (ILP). Two principal domain features are domain fusions and domain-domain interactions. Various relevant genomic and proteomic features of PPI are exploited, from five popular genomic and proteomic databases. Integrating protein domain data and various kinds of data from multiple genomic and proteomic databases, we constructed biologically significant ILP background knowledge of nearly 220,000 ground facts. The experimental results from 10-fold cross-validation demonstrated that our approach can better predict protein-protein interactions than other computational methods. When applied to many PPI data sets, our method can more reliably predict PPI in terms of the expression profile reliability indexes. The induced ILP rules give us a lot of interesting biological reciprocal relationships among protein-protein interactions, protein domains, and genomic/proteomic features related to protein-protein interactions.

Supplementary materials are now available at http://www.jaist.ac.jp/∼s0560205/PPI/.

BIOKDD '07, August 12, 2007, San Jose, California, USA.
Copyright 2007 ACM 978-1-59593-839-8/07/0008 ...$5.00.

Keywords

Protein-protein interaction, Domain fusion, Domain-domain interaction, Genomic/proteomic features, Inductive logic programming.

1. INTRODUCTION

Protein-protein interactions are indispensable at almost every level of cell function, in the structure of sub-cellular organelles, in the transport across the various biological membranes, in muscle contraction, signal transduction, and regulation of gene expression, etc. Detecting protein functions via prediction of protein-protein interactions (PPI) has emerged as a new trend, both *in vitro* and *in silico*. Therefore, prediction of protein-protein interactions has become one of the most challenging tasks in the post-genomic era. Experimental techniques have marked unmistakable progress in finding out and verifying protein interactions for diverse organisms, including well-known ones such as two-hybrid assay [9], [28], affinity purification and mass spectrometry [1], phage display [22]. Because of little overlap among these experimental databases, the question about their reliability is raised.

With the recent blooming of public proteomic and genomic databases, numerous computational approaches offer a chance to study more widely and deeply regarding protein-protein interactions. Depending on the source of information used, computational approaches can be categorized in three groups: structure-based approach such as the work of [3], sequence-based approach such as the work of [14], and genome-based approach such as the work of [19]. Besides methods based on a single data source, many bioinformaticians make the effort to integrate multiple data sources to better predict PPI. Jansen *et al.* [10] used a Bayesian network approach for integrating weakly predictive genomic features into reliable predictions of protein-protein interactions. Several kernels for different data sources like protein squences, Gene Ontology annotations, local properties of networks, etc. are combined to infer PPI [2]. Some other efforts were the probabilistic decision tree approach [30], inductive logic programming method [25], probabilistic model [20], etc.

From multiple data sources, these works can extract and combine various genomic and proteomic features related to PPI. The obtained results showed many advantages of multiple data source integration. The shortcoming of their work is that they did not take protein domains into account.

However, it is a fact that the biological mechanism behind protein-protein interactions involves protein domains and their interactions [18].

Protein domains are structural and/or functional units of proteins that are conserved through evolution to represent protein structures or functions. They are believed to be the key regulators in protein-protein interactions. Interactions among domains are needed as stable channels of PPI. Recently, prediction of PPI based on domains has received much attention in many ongoing studies. One of the pioneering works based on protein domains is an association method developed by Sprinzak and Margalit [23]. Kim *et al.* improved the association method by considering the number of domains in each protein [12]. Han *et al.* proposed a domain combination-based method by considering the possibility of domain combinations appearing in both interacting and non-interacting sets of protein pairs [8]. A graph-oriented method is proposed by Wojcik and Schachter called the interacting domain profile pairs (IDPP) method [29]. Chen *et al.* used domain-based random forest framework to predict PPI [4].

The previous work all treasured the biological roles of protein domains in PPI prediction. The main disadvantage of these methods is that most of them merely considered the co-occurrence of domains/domain pairs. To predict PPI comprehensively, it is necessary to combine various domain features and genomic/proteomic features.

In this paper, we present a novel integrative domain-based approach using inductive logic programming to predict protein-protein interactions. The key idea of our computational method is to integrate protein domain features and multiple genomic and proteomic features. To combine efficiently both features of protein domains and different features of genomes and proteomes to predict PPI, we specified two main tasks. The first one is extracting as many useful domain and genomic/proteomic features as possible related to PPI. From seven popular databases, we extracted more than two hundred thousand ground facts of domain fusion, domain-domain interaction features and various other biologically significant genomic/proteomic features. The second one is employing inductive logic programming (ILP) with the huge amount of background knowledge to effectively infer PPI.

To demonstrate the advantages of the integration domain features and genomic/proteomic features in PPI prediction, we conducted 10-fold cross validation tests for our methods and two other methods based on single domain features, and also for the non domain-based approach using multiple genomic databases. For all cases, our method performed considerably better than others. The expression profile reliability index (EPR Index) additionally showed the high reliability of our methods when applied to several PPI datasets. At last, analyzing various produced rules, many interesting relationships among PPI and DDI, and protein functions, biological processes were found. Our proposed methods can be tuned to predict PPI for diverse organisms and other genomic and proteomic data sources.

The remainder of the paper is organized as follows. In Section 2, we present our proposed method to predict PPI based on domains using ILP and multiple genomic and proteomic databases. The comparative evaluation of the experiments is given in Section 3. Predictive rules of PPI, as well as discussion, are presented in Section 4. Some concluding remarks are given in Section 5.

2. MATERIALS AND METHODS

In this section, we present our proposed method to predict protein-protein interactions based on domain and multiple genomic and proteomic data using ILP. Two main tasks of the method are: (1) Constructing integrated background knowledge[1] of domain features and multiple genomic and proteomic features, and (2) Learning PPI predictive rules by ILP from the constructed background knowledge. Constructing ILP background knowledge requires two steps. The first one is defining ILP predicates. The second one is extracting ground facts[2] to define extensionally predicates.

When choosing a feature, we concentrated on two points. First is the biological role of that feature in protein-protein interactions or domain-domain interactions, and second is the availability of data of that feature. Based on results of experimental and computational research on PPI, twenty two features of protein domains and genomes/proteomes were chosen and were formulated using ILP predicates. The huge database of more than 220,000 ground facts of twenty two predicates is sufficient for accurate PPI prediction.

We first introduce briefly about Inductive Logic Programming (ILP) and some bioinformatic applications of ILP in Section 2.1. Then the first task in our proposed method is presented in Subsections 2.2, 2.3, and 2.4. Subsection 2.5 describes the second task.

2.1 Inductive Logic Programming

Inductive Logic Programming is the intersection of machine learning and logic programming [15]. ILP aims to develop theories, techniques, and tools for inducing hypotheses from observations using representations from computational logic. ILP studies learning from examples, within the framework provided by clausal logic. Here the examples and background knowledge are given as clauses, and the theory that is to be induced from these, is also to consist of clauses.

An ILP system is generally set with three languages:

L_O : the language of observations
L_B : the language of background knowledge
L_H : the language of hypotheses

Given a consistent set of examples of observations $O \subseteq L_O$ and consistent background knowledge $B \subseteq L_B$, ILP systems find hypotheses $H \in L_H$ such that:

$$B \wedge H \vdash O$$

Distinguishing features of ILP are its ability to take into account background (domain) knowledge in the form of logic programs, and the expressive power of the language of discovered patterns [7]. ILP is particular suitable for bioinformatics tasks because of its ability to take into account background knowledge and work directly with structured data. The ILP system GOLEM has been applied to find the predictive theory about the relationship between chemical structure and activity, eg. the problem of inhibition of E.Coli Dihydrofolate Reductase by two different groups of drugs (pyrimidines and triazines) [13]. Other central concerns of bioinformatics have been convincingly solved by ILP

[1] the term 'background knowledge' is used here in terms of the language of inductive logic programming.

[2] the term 'ground facts' is used here in terms of the language of inductive logic programming.

, such as protein secondary structure prediction [16], protein fold recognition [27], etc.

2.2 Extracting Domain Fusion and Domain-Domain Interaction Data

Protein domains form the structural or functional units of proteins that partake in intermolecular interactions. The existence of certain domains in proteins can therefore suggest the propensity for the proteins to interact or form a stable complex to bring about certain biological functions. Domain fusion and domain-domain interaction features have important biological roles in PPI prediction [26], [18], and these two domain features are extracted in our work.

Let P denote the set of considered proteins p_i. Denote by D the set of all protein domains d_k which belong to proteins p_i. A protein pair (p_i, p_j) that interacts together is denoted by p_{ij}, and a protein pair that does not interact together by $\neg p_{ij}$. Similarly for a domain pair (d_k, d_l), d_{kl} represents an interaction, and $\neg d_{kl}$ a non-interaction.

Domains of interacting proteins have more chance to fuse together than domains of non-interacting proteins. Therefore, when finding a pair of proteins which have fused domains, we can predict an interaction between them. Domain fusion data is referred from Domain Fusion Database [26]. We extracted domain fusion data for protein pairs (p_i, p_j), $\forall p_i, p_j \in P$. The following predicate represents the domain fusion between two proteins:

$$\texttt{domain\_fusion}(+\texttt{protein},\ +\texttt{protein},\ \#\texttt{FUSION}) \tag{1}$$

Note that in the ILP system used - system Aleph (A learning engine for proposing hypothesis) [24], there are some *mode declarations* to build the bottom clauses, and a simple mode type is one of the following: (1) *the input variable* (+), (2) *the output variable* (−), or (3) *the constant term* (#). Predicate (1) means whether two input proteins, A and B, have fused domains or not (valued "yes" by the constant term #FUSION). This predicate is supported by a set of ground facts G_{domain_fusion}, e.g., domain_fusion (ap3m_yeast, ap3b_yeast, yes). After preprocessing, the set G_{domain_fusion} consists of 255 ground facts for protein pairs.

The assumption that proteins interact with each other through interactions of their domains is widely accepted and validated. The domain-domain interaction data is exploited to more reliably predict PPI. We extracted DDI data from **iPfam** database (http://www.sanger.ac.uk/Software/ Pfam/iPfam/). iPfam is a resource that describes domain-domain interactions that are observed in PDB entries. The domains are defined by Pfam. When two or more domains occur in a single structure, the domains are analysed to see if they form an interaction considered by the bonds forming the interaction are calculated.

We considered two features of DDI. The first feature is whether a protein pair (p_i, p_j) has a domain interaction d_{kl}, and if yes, how many d_{kl} it has. This information is formulated by predicate:

$$\texttt{hasddi}(+\texttt{protein},\ +\texttt{protein},\ \#\texttt{DDI}) \tag{2}$$

The set of ground facts for this predicate G_{ddi} includes 573 ground facts, some of them are: hasddi(jsn1_yeast, yip1_yeast,2), hasddi(msh4_yeast,msh5_yeast,5), etc.

The number of domain-domain interactions of a protein is one of the features which may increase or decrease the probability of its interaction with others. So we considered the relationship between PPI and the number of DDI of each interacting partner. This relationship is presented in predicate 3.

$$\texttt{num\_ddi}(+\texttt{protein},\ \#\texttt{NUM\_DDI}) \tag{3}$$

Denoted by G_{num_ddi}, the set of ground facts of the above predicate contains 289 ground facts. We found that there are some proteins having a large number of DDI, for example num_ddi(did4_yeast,20) or num_ddi(bud27_yeast,39), and these proteins potentially interact with many other proteins.

2.3 Extracting Proteomic and Genomic Data from Multiple Databases

In addition to domain fusion and domain-domain interaction features as shown in the previous section, we mined genomic and proteomic data from UniProt database, CYGD database, InterPro database, Gene Ontology database, and Gene Expression database to detect useful genomic and proteomic features for PPI prediction.

As the world's most comprehensive catalog of information on proteins, **UniProt database** (http://www.pir.uniprot.org/) largely provides functional, structural or other categories (in Keyword - KW line); regions or sites of interest in the sequences (in Feature Table - FT lines); describes enzymes coded (EC) and pointers to information related to entries and found in data collections other than Uniprot such as GO database, PIR database, PROSITE database, Pfam database, and Interpro database (in Database cross-Reference - DR line). There are the following predicates for each kind of information for one protein.

$$\texttt{keyword}(+\texttt{protein},\ \#\texttt{KW}) \tag{4}$$
$$\texttt{feature}(+\texttt{protein},\ \#\texttt{FT}) \tag{5}$$
$$\texttt{coded\_enzyme}(+\texttt{protein},\ \#\texttt{EC}) \tag{6}$$
$$\texttt{dr\_go}(+\texttt{protein},\ -\texttt{GO\_TERM}) \tag{7}$$
$$\texttt{dr\_pir}(+\texttt{protein},\ -\texttt{PIR\_ID}) \tag{8}$$
$$\texttt{dr\_prosite}(+\texttt{protein},\ -\texttt{PROTSITE\_ID}) \tag{9}$$
$$\texttt{dr\_pfam}(+\texttt{protein},\ -\texttt{PFAM\_ID}) \tag{10}$$
$$\texttt{dr\_interpro}(+\texttt{protein},\ -\texttt{INTERPRO\_ID}) \tag{11}$$

For example, some extracted data for these predicates are keyword(ace1_yeast,transcription regulation), feature(ldb7 _yeast, chain chromatin structure remodeling complex), coded _enzyme(uqcr1_yeast, ec1.10.2), and dr_go(twoa5d_yeast, go0005935), etc. The first three predicates present general protein features that should effect their interactions. The other give references to other databases. Data from different databases related to PPI are bound by these predicates. We extracted 10,919 ground facts for these UniProt predicates.

The MIPS Comprehensive Yeast Genome Database **CYGD** (http://mips.gsf.de/genre/proj/ yeast/) aims to present information on the molecular structure and functional network of the entirely sequenced, well-studied model eukaryote, the budding yeast *Saccharomyces cerevisiae*.

Among various information provided by CYGD, catalogues of functions, catalogues of subcellular locations, catalogues of phenotypes, catalogues of complexes, and catalogues of proteins should be mined to discover the biological relationship between such catalogues and protein-protein interactions. Also, proteins in the same catalogue have more chance to interact together than other proteins. The set of ground facts extracted from CYGD database G_{CYGD} consists of

2,152 ground facts. Here are some examples: subcell_cat (ahc1 yeast, cytoplasm), phenotype_cat(cyk2 yeast, cell cycle defects), etc.

$$\texttt{function\_cat}(+\texttt{protein},\ \#\texttt{FUNCAT}) \tag{12}$$

$$\texttt{subcell\_cat}(+\texttt{protein},\ \#\texttt{SUBCELLCAT}) \tag{13}$$

$$\texttt{phenotype\_cat}(+\texttt{protein},\ \#\texttt{FENCAT}) \tag{14}$$

$$\texttt{complex\_cat}(+\texttt{protein},\ \#\texttt{COMPLEXCAT}) \tag{15}$$

$$\texttt{protein\_cat}(+\texttt{protein},\ \#\texttt{PROTEINCAT}) \tag{16}$$

InterPro database (http://www.ebi.ac.uk/interpro/) is a database of protein families, domains and functional sites in which identifiable features found in known proteins can be applied to unknown protein sequences. We considered the association between InterPro identifers and GO terms. There are 556 ground facts which support this predicate.

$$\texttt{interpro\_go}(+\texttt{INTERPRO\_ID},\ -\texttt{GO\_TERM}) \tag{17}$$

Gene Ontology database (http://www.geneontology.org) has three organizing principles: molecular function, biological process and cellular component. The terms in an ontology are linked by two relationships, *is_a* and *part_of*. The relationships of interacting partners in a PPI may effect their interaction. Predicates 18, 19, having 438 ground facts (e.g., is_a (go0000002, go0007005), part_of (go0000032, go0007047)) show these relationships:

$$\texttt{is\_a}(+\texttt{GO\_TERM},\ -\texttt{GO\_TERM}) \tag{18}$$

$$\texttt{part\_of}(+\texttt{GO\_TERM},\ -\texttt{GO\_TERM}) \tag{19}$$

Proteins in the same complex are often co-expressed, and then this genomic feature is useful in predicting PPI. The **gene expression** coefficients referred to in [10] between two proteins are presented in the following predicate (having 200,000 ground facts):

$$\texttt{expression}(+\texttt{protein},\ +\texttt{protein},\ \#\texttt{COEFFICIENT}) \tag{20}$$

Two last predicates express information about the number of protein-protein interactions (with 690 ground facts) and interaction generality of two interacting partners (with 1,718 ground facts). Interaction generality is the number of proteins that interact with both interacting partners.

$$\texttt{num\_ppi}(+\texttt{protein},\ +\texttt{protein},\ \#\texttt{NUM\_PPI}) \tag{21}$$

$$\texttt{ig}(+\texttt{protein},\ +\texttt{protein},\ \#\texttt{IG}) \tag{22}$$

2.4 Constructing Background Knowledge for Predicting Protein-Protein Interactions

After twenty two predicates are defined, data in terms of ground facts for these predicates are next exploited from seven databases (two databases for domain features and five others for genomic and proteomic features). In succession, we denote the sets of ground facts extracted from UniProt database, CYGD database, InterPro database, Gene Ontology database, and Gene Expression database by $G_{UniProt}$, G_{GO}, $G_{InterPro}$, G_{CYGD}, and $G_{expression}$. Algorithm 1 presents the procedure to extract data from multiple databases to construct background knowledge for PPI prediction.

Algorithm 1 Extracting domain and protein data from multiple sources.

```
Input:
    Set of proteins $P \supset \{p_i\}$.
Output:
    Sets of ground facts $G_{domain\_fusion}$, $G_{ddi}$, $G_{num\_ddi}$, $G_{UniProt}$, $G_{CYGD}$, $G_{InterPro}$, $G_{GO}$, , $G_{expression}$, $G_{ig}$, and $G_{num\_ppi}$.
Initialize all sets of ground facts $G_L := \emptyset$ ($\forall\ G_L \in G_{domain\_fusion}$, $G_{ddi}$, $G_{num\_ddi}$, $G_{UniProt}$, $G_{CYGD}$, $G_{InterPro}$, $G_{GO}$, , $G_{expression}$, $G_{ig}$, and $G_{num\_ppi}$); $D := \emptyset$.
Extract all domains $d_k$ belonging to proteins $p_i$; $D := D \cup \{d_k\}$.
for each protein pair $(p_i,\ p_j)$
    for all $d_k \in p_i$ and $d_l \in p_j$
        if $fused(d_k, d_l) = \texttt{true}$ then
            $G_{domain\_fusion} := G_{domain\_fusion} \cup \{(p_i,\ p_j)\}$
        if $\exists\ d_{kl}$ then
            $G_{ddi} := G_{ddi} \cup \{(p_i,\ p_j)\}$
            Count the number of DDI $num\_ddi_i$ and $num\_ddi_j$ for proteins $p_i$, and $p_j$ respectively;
    $G_{num\_ddi} := G_{num\_ddi} \cup \{(p_i, num\_ddi_i)\} \cup \{(p_j, num\_ddi_j)\}$.
for each protein $p_i \in P$
    Extract data from UniProt database and CYGD database for $G_{UniProt}$ and $G_{CYGD}$ respectively;
    $G_{UniProt} = G_{UniProt} \cup \{p_i, p_i.data\}$;
    $G_{CYGD} = G_{CYGD} \cup \{p_i, p_i.data\}$.
    Extract mapping data between GO terms $g_i$ and Interpro identifiers $t_i$ related to $p_i$ from InterPro database for $G_{Interpro}$;
    $G_{InterPro} = G_{InterPro} \cup \{t_i, g_i.\}$.
for each protein $p_i \in P$
    for each protein $p_j \in P$
        Extract the relationship $r_{ij}$ between GO terms $(g_i,\ g_j)$ related to $(p_i,\ p_j)$ from GO database;
        $G_{GO} = G_{GO} \cup \{r_{ij}(g_i, g_j)\}$.
        Extract the expression correlation coefficients $e_{ij}$ of $(p_i, p_j)$;
        $G_{expression} = G_{expression} \cup \{p_i, p_j, e_{ij}\}$
        Extract the interaction generality of PPI $n_{ij}$ of $(p_i,\ p_j)$; $G_{ig} = G_{ig} \cup \{p_i, p_j, n_{ij}\}$
        if $\exists\ p_{ij}$ then
            $num\_ppi_i := num\_ppi_i\ +\ 1$;
    $G_{num\_ppi} := G_{num\_ppi} \cup \{(p_i, num\_ppi_i)\}$.
return $G_{domain\_fusion}$, $G_{ddi}$, $G_{num\_ddi}$, $G_{UniProt}$, $G_{CYGD}$, $G_{InterPro}$, $G_{GO}$, , $G_{expression}$, $G_{ig}$, $G_{num\_ppi}$.
```

2.5 Predicting Protein-Protein Interaction Using Inductive Logic Programming

The proposed integrative domain-based ILP framework for predicting PPIs from multiple genomic and proteomic databases is described in Algorithm 2.

The previous framework presents the common procedures of the ILP method. Step 2 and Step 3 are for generating positive and negative examples $S_{interact}$, $S_{\neg interact}$ respectively (see more Subsection 3.1). In Step 4, we extracted background knowledge $S_{background}$ including both domain features and genomic and proteomic features from sets of ground facts of defined predicates (see Section 2.4). In Step 5, in our experiments, system Aleph was applied to induce rules. Aleph is an advanced ILP system that uses a top-down ILP covering algorithm.

Aleph requires three input files to construct theories: positive examples, negative examples and background knowledge. Positive and negative examples can simply be consid-

Algorithm 2 An integrative domain-based ILP framework for PPI prediction

```
Input:
    Set of protein-protein interactions $S_{interact} \supset \{p_{ij}\}$
    Number of negative examples $(\neg p_{ij})$ $N$
    Sets of ground facts $G_{domain\_fusion}$, $G_{ddi}$, $G_{num\_ddi}$, $G_{UniProt}$, $G_{CYGD}$, $G_{InterPro}$, $G_{GO}$, , $G_{expression}$, $G_{ig}$, and $G_{num\_ppi}$.
Output:
    Set of rules $R$ for protein-protein interaction prediction.

$R := \emptyset$.
Extract positive examples for the set $S_{interact}$.
Generate $N$ negative examples $\neg p_{ij}$s by selecting $N$ protein pairs $(p_i, p_j)$ where $p_i,\ p_j \in P$ and $p_i,\ p_j$ are located in different subcellular compartments;
$S_{\neg interact} = \{\neg p_{ij}\}$.
call Algorithm 1 to generate sets of ground facts $G_L$ and $S_{background} = \bigcup G_L$ $(\forall\ G_L \in G_{domain\_fusion}$, $G_{ddi}$, $G_{num\_ddi}$, $G_{UniProt}$, $G_{CYGD}$, $G_{InterPro}$, $G_{GO}$, , $G_{expression}$, $G_{ig}$, and $G_{num\_ppi}$.
Run an ILP program with $S_{interact}$, $S_{\neg interact}$ and $S_{background}$ to induce rules $r$.
    $R := R \cup \{r\}$.
return $R$.
```

ered as ground facts. Background knowledge is in the form of Prolog clauses that encode information relevant to the domain. All predicates appearing in hypothesized clauses have to be declared, and amongst them the target predicate is learned to induce rules. The target predicate in our work is: `has_int(+protein, +protein)`, meaning that two arbitrary proteins, A and B, interact. Aleph learns three inputs and induces rules (hypothesized clauses) in terms of the relationships between the target predicate and other predicates declared in background knowledge.

3. EXPERIMENTAL RESULTS

3.1 Experiment Design

We concentrate on predicting PPI for *Saccharomyces cerevisiae*, a budding yeast, due to the availability of *Saccharomyces cerevisiae* data. We carried out experimental comparative evaluation for protein-protein interaction prediction.

To assess the performance of our method for PPI prediction, we did three comparative tests to demonstrate: (1) the advantages of the integration of multiple proteomic and genomic features in our method, (2) the advantages of domain-based approach, and (3) the reliability of our method. First, ROC curves of 10-fold cross validation tests were produced to compare our proposed method with other domain-based methods, particularly AM method and SVMs method. Second, we also conducted 10-fold cross validation tests for an ILP method with multiple genomic databases, but not using domain features, and compared those results with our method in terms of sensitivity and specificity. At last, applying our method to several PPI datasets like Ito dataset [9], Uetz dataset [28], MIPS dataset (http://mips.gsf.de/proj/ppi/), DIP dataset (http://dip.doe-mbi.ucla.edu/), etc., we estimated EPR indexes [5] to show the reliability of our method.

For three comparative tests for PPI prediction, we used the core data of Ito data set [9] with more than two IST hits[3], as positive examples, and selected at random 1000 protein pairs whose elements are in separate subcellular compartments as negative examples. Each interaction in the interaction data originally shows a pair of bait and prey ORF (Open Reading Frame). After removing all interactions in which either bait ORF or prey ORF is not found in UniProt database, we obtained 718 interacting pairs from the original 841 pairs. Subsection 3.2 shows the experimental results of PPI prediction.

3.2 Predicting Protein-Protein Interactions

With the same positives and negatives datasets, we conducted 10-fold cross validation tests for our method, AM method and SVMs method. AM method calculated the probability of protein pairs based on protein domains [23]. In our experiment, the probability threshold is set to 0.05. For SVMs method, we used SVM^{light} [11]. The linear kernel with default values of the parameters was used. For Aleph, we selected *minpos* $= 2$ and *noise* $= 0$, i.e. the lower bound on the number of positive examples to be covered by an acceptable clause is 2, and there are no negative examples allowed to be covered by an acceptable clause. We also used the default evaluation function *coverage* which is defined as $P - N$, where P, N are the number of positive and negative examples covered by the clause.

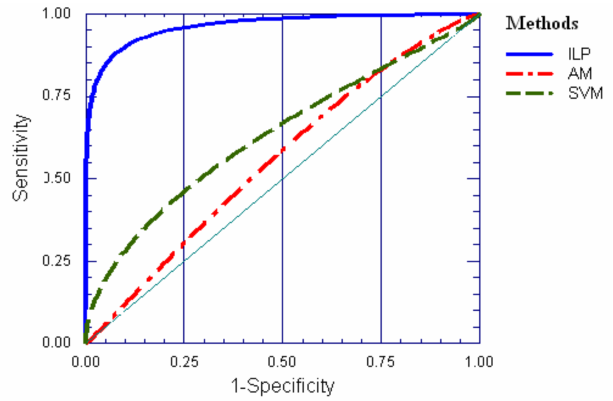


Figure 1: Comparative ROC curves of ILP, SVMs and AM method with 1000 negative examples.

The ROC curves of ILP, AM and SVMs methods with 1000 negative examples are shown in Figure 1. ROC curve (Receiver Operating Characteristic curve) shows the tradeoff between sensitivity and specificity (any increase in sensitivity will be accompanied by a decrease in specificity). Sensitivity refers to the ability of the test to detect individuals who actually have the disorder. On the other hand, the term specificity means that the test is specific to the disorder being assessed and that it does not give a positive result because of other conditions.

The ROC curve of our method is close to the left-hand border and then the top border of the ROC space. On the other hand, ROC curves of AM method and SVMs method are close to the 45-degree diagonal of the ROC space. The ROC curve demonstrates that our method has a consid-

[3] IST hit means how many times the corresponding interaction was observed. The higher the IST number, the more reliable the corresponding interaction is.

Table 1: Evaluation of our proposed method using EPR index with Ito, Uetz, Ito+Uetz, MIPS, and DIP PPI datasets.

Data	Number of interactions		EPR index	
	Original	Our proposed	Original	Proposed
Ito	4549	2000	0.191 ± .0306	**0.349** ± .0491
Uetz	1474	799	0.445 ± .0588	**0.539** ± .0831
Ito+Uetz	5827	2699	0.238 ± .0287	**0.363** ± .0437
MIPS	14146	6810	0.595 ± .0337	**0.685** ± .0422
DIP	15409	9047	0.418 ± .0260	**0.541** ± .0371

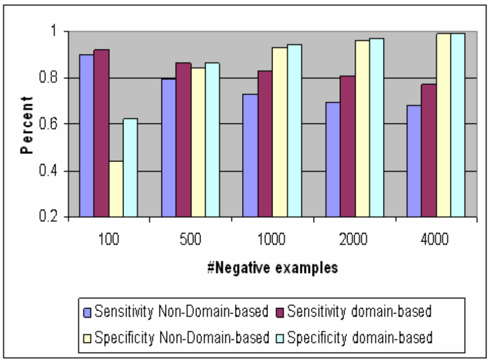


Figure 2: The sensitivity and specificity (denote by Sensitivity1 and Specificity1) of non-domain based approach are compared with those (denote by Sensitivity2 and Specificity2) of our proposed method with various sets of negative examples by 10-fold cross-validation tests.

erably better performance than those of AM and SVMs method.

Conducting 10-fold cross-validation with various tested numbers of negative examples, the results (in Figure 2) show that our method achieved higher sensitivity, and higher or equal specificity, than the non-domain based approach [25].

To show how reliable our method is, we compared the EPR indexes of original PPI datasets and datasets predicted by our method. The EPR index estimates the biologically relevant fraction of protein interactions detected in a high throughput screen. For each given dataset, we first excluded all protein pairs which overlap with those in the training dataset. All retrieved protein pairs that classified as positives are then estimated in terms of their EPR indexes. Table 1 shows the higher ERP index of our method compared with original ones.

4. DISCUSSION

The experimental results have shown that ILP approach potentially predicts PPI and DDI with high sensitivity and specificity. Furthermore, the inductive rules of ILP encouraged us to discover many interesting biological reciprocal relationships among protein-protein interactions and protein domains, and other genomic/proteomic features related to protein-protein interactions. Analysing our results in comparison with information in biological literatures and books, we found that ILP induced rules could be applied to further related studies in biology.

Studying the rules of PPI prediction related to domain-domain interaction information, we found many interesting rules. For example, the following rule shows that if two proteins have domains belonging to domain databases like PROSITE database or InterPro database and these domains interact with each other, they may interact.

has_int (A,B) :- dr_prosite (B, C), dr_prosite (A, C), ddi (A, B, yes) with 43 positives covered

*has_int(A,B) :- dr_interpro(B,C), dr_interpro(A,C), ddi (A, B, yes)*with 90 positives covered.

A large number of positives, which indicates these rules, confirms why domain-domain interactions are considered as key factors to predict PPI.

Considering the group of proteins which may be required for the production of *pyridoxine* (vitamin B6) sno1_yeast, snz3_yeast snz1_yeast, and snz2_yeast, we found that each pair in this group has an interaction which satisfies the following rule:

has_int(A,B) :- ig (A, B, C), C = 1, ddi (A, B, yes), function_cat (B, cell rescue defense and virulence).

This rule means interaction of protein A and protein B may occur if the proteins satisfy three conditions. First is that they interact with the same protein. Second is that they have at least one DDI. Third is that one of them is categorized to function catalogue *cell rescue defense and virulence*. We knows that PPI play an important role in drug design, so such rules and their evidence, are expected to help us to discover interesting relationships between PPI, DDI and protein function in pharmaceuticals.

Two most popular rules related to domain fusion information are:

has_int(A,B) :- dr_go(B,C), part_of(C,D), domain_ fusion(A,B,yes)

has_int(A,B) :- dr_go(B,C), dr_go(A,C), domain_ fusion(A,B,yes)

The first one covers 199 positives and the second one covers 217 positives. Both of these rules consist of GO terms and domain fusion information. According to the second rule, if two proteins have GO terms and their domains are fused in another protein, there may occur an interaction.

Our induced rules with large number of positives prove that if a pair of proteins, A and B, are located in the same subcellular compartment, protein A potentially interacts with protein B. In case of *nucleus compartment*, there are 216 covered positives, 284 for *cytoplasm compartment* and 15 for *mitochondria compartment*. However, surprisingly among induced rules, we found a rule with 37 positives that showed the phenomenon of two proteins being in different subcellular locations but interacting.

has_int(A,B) :- subcell_cat(B,nucleus), subcell_cat(A, cytoplasm), function_cat(A,transcription).

This phenomenon could occur when there is a certain translocation or post-translation modification of proteins in different subcellular compartments.

Since protein-protein interactions have close biological associations with domain-domain interactions, discovering DDI from PPI data is an area of much ongoing research. Ng *et al.* proposed to an integrative approach to infer putative domain-domain interactions from three data sources, including experimentally-derived protein interactions, protein complexes and Rosetta stone sequences [17]. To predict DDI, the maximum likelihood estimation (MLE) is applied by Deng *et al.* [6]. Riley et al. proposed a domain pair exclusion analysis (DPEA) for predicting DDI from databases of protein interactions [21]. These works showed that DDI can be efficiently predicted from PPI data. In the future, as more DDI are predicted and validated, our work is potential to reliably predict PPI. From PPI networks, we can build up more complex protein complexes and pathways in cell, such as signal transduction pathways or metabolic pathways.

5. CONCLUSION

We have presented an integrative domain-based approach using ILP and multiple genome databases to predict protein-protein interactions. The experimental results demonstrated that our proposed method could produce comprehensible rules, and at the same time, performed well in comparison with other work on protein-protein interaction prediction. In future work, we would like to investigate induced rules to study further the biological relationships among PPI, DDI, domain fusion and other genomic/proteomic features. Integrating more biological features may achieve better results. We also would like to apply the ILP approach to other important tasks, such as determining protein functions, and determining the sites, and interfaces of PPI using DDI data.

6. REFERENCES

[1] A. Bauer and B. Kuster. Affinity purification-mass spectrometry: Powerful tools for the characterization of protein complexes. *Eur. J. Biochem.*, 270(4):570–578, 2003.

[2] A. Ben-Hur and W. S. Noble. Kernel methods for predicting protein-protein interactions. *Bioinformatics*, 21(suppl1):i38–46, 2005.

[3] J. R. Bock and D. A. Gough. Predicting protein-protein interactions from primary structure. *Bioinformatics*, 17(5):455–460, 2001.

[4] X. Chen and M. Liu. Prediction of protein-protein interactions using random decision forest framework. *Bioinformatics*, 21(24):4394–4400, 2005.

[5] C. M. Deane, L. Salwinski, I. Xenarios, and D. Eisenberg. Protein interactions: Two methods for assessment of the reliability of high-throughput observations. *Mol Cell Proteomics*, pages M100037–MCP200, 2002.

[6] M. Deng, S. Mehta, F. Sun, and T. Chen. Inferring domain-domain interactions from protein-protein interactions. *Genome Res.*, 12(10):1540–1548, 2002.

[7] S. Dzeroski and N. Lavrac, editors. *Relational Data Mining.* Springer, 2001.

[8] D. Han, H.S.Kim, J.Seo, and W.Jang. A domain combination based probabilistic framework for protein-protein interaction prediction. In *Genome Informatics*, pages 250–259, 2003.

[9] T. Ito, T. Chiba, R. Ozawa, M. Yoshida, M. Hattori, and Y. Sakaki. A comprehensive two-hybrid analysis to explore the yeast protein interactome. In *Proc. Natl. Acad. Sci. USA 98*, pages 4569–4574, 2001.

[10] R. Jansen, H. Yu, D. Greenbaum, Y. Kluger, N. J. Krogan, S. Chung, A. Emili, M. Snyder, J. F. Greenblatt, and M. Gerstein. A Bayesian Networks Approach for Predicting Protein-Protein Interactions from Genomic Data. *Science*, 302(5644):449–453, 2003.

[11] T. Joachims. Making large-scale support vector machine learning practical. In B. Scholköpf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods: Support Vector Machines.* MIT Press, Cambridge, MA, 1998.

[12] R. Kim, J. Park, and J. Suh. Large scale statistical prediction of protein - protein interaction by potentially interacting domain (PID) pair. In *Genome Inform. Ser. Workshop Genome Inform*, pages 48–50, 2002.

[13] R. King, S. Muggleton, R. Lewis, and M. Sternberg. Drug design by machine learning: The use of inductive logic programming to model the structure activity relationships of trimethoprim analogues binding to dihydrofolate reductase. *Proc. of the National Academy of Sciences of the USA*, 89(23):11322–11326, 1992.

[14] L. R. Matthews, P. Vaglio, and J. R. et al. Identification of potential interaction networks using sequence-based searches for conserved protein-protein interactions or 'interologs'. *Genome Res.*, 11(12):2120–2126, 2001.

[15] S. Muggleton. *Inductive Logic Programming.* Academic Press, 1992.

[16] S. Muggleton, R. King, and M. Sternberg. Protein secondary structure prediction using logic-based machine learning. *Protein Eng.*, 6(5):549–, 1993.

[17] S. Ng, Z. Zhang, and S. Tan. Integrative approach for computationally inferring protein domain interactions. *Bioinformatics*, 19(8):923–929, 2003.

[18] T. Pawson, M. Raina, and N. Nash. Interaction domains: from simple binding events to complex cellular behavior. *FEBS Letters*, 513(1):2–10, 2002.

[19] M. Pellegrini, E. M. Marcotte, and M. J. T. et al. Assigning protein functions by comparative genome analysis: Protein phylogenetic profiles. In *Proc. Natl. Acad. Sci. USA*, volume 96, pages 4285–4288, 1999.

[20] D. R. Rhodes, S. A. Tomlins, S. Varambally, V. Mahavisno, T. Barrette, S. Kalyana-Sundaram, D. Ghosh, A. Pandey, and A. M. Chinnaiyan. Probabilistic model of the human protein-protein interaction network. *Nat Biotech*, 23(8):1087–0156, 2005.

[21] R. Riley, C. Lee, C. Sabatti, and D. Eisenberg. Inferring protein domain interactions from databases of interacting proteins . *Genome Biology*, 6(10):R89, 2005.

[22] G. P. Smith. Filamentous fusion phage: Novel expression vectors that display cloned antigens on the virion surface. *Science*, 228(4705):1315–1317, 1985.

[23] E. Sprinzak and H. Margalit. Correlated

sequence-signatures as markers of protein-protein interaction. *Journal of Molecular Biology*, 311(4):681–692, 2001.

[24] A. Srinivasan, 1993. http://web.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/.

[25] T. Tran, K.Satou, and T.B.Ho. Using inductive logic programming for predicting protein-protein interactions from multiple genomic data. In *PKDD'05*, pages 321–330, 2005.

[26] K. Truong and M. Ikura. Domain fusion analysis by applying relational algebra to protein sequence and domain databases. *BMC Bioinformatics*, 4(16):1–10, 2003.

[27] M. Turcotte, S. Muggleton, and M. Sternberg. Protein fold recognition. In *Proc. of the 8th International Workshop on Inductive Logic Programming (ILP-98)*, pages 53–64, 1998.

[28] P. Uetz, L. Giot, G. Cagney, T. A. Mansfield, R. S. Judson, J. R. Knight, D. Lockshon, V. Narayan, M. Srinivasan, P. Pochart, A. Qureshi-Emili, Y. Li, B. Godwin, D. Conover, T. Kalbfleisch, G. Vijayadamodar, M. Yang, M. Johnston, S. Fields, and J. M. Rothberg. A comprehensive analysis of protein-protein interactions in saccharomyces cerevisiae. *Nature*, 403(6770):623–627, February 2000.

[29] J. Wojcik and V. Schachter. Protein-protein interaction map inference using interacting domain profile pairs. *Bioinformatics*, 17(suppl1):S296–305, 2001.

[30] L. Zhang, S. Wong, O. King, and F. Roth. Predicting co-complexed protein pairs using genomic and proteomic data integration. *BMC Bioinformatics*, 5(38), 2004.

A Linear-time Algorithm for Predicting Functional Annotations from Protein Protein Interaction Networks[*]

Yonghui Wu
Department of Computer Science & Engineering
University of California
Riverside, CA 92507
yonghui@cs.ucr.edu

Stefano Lonardi[*]
Department of Computer Science & Engineering
University of California
Riverside, CA 92507
stelo@cs.ucr.edu

ABSTRACT

Recent proteome-wide screening efforts have made available genome-wide, high-throughput protein-protein interaction (PPI) maps for several model organisms. This has enabled the systematic analysis of PPI networks, which has become one of the primary challenges for the system biology community. Here we address the problem of predicting the functional classes of proteins (i.e., GO annotations) based solely on the structure of the PPI network. We present a maximum likelihood formulation of the problem and the corresponding learning and inference algorithms. The time complexity of both algorithms is linear in the size of the PPI network and experimental results show that their accuracy in the functional prediction outperforms current existing methods.

1. INTRODUCTION

High-throughput protein-protein interaction (PPI) networks with various levels of proteome coverage are currently available for several model organisms, namely *S. cerevisiae* [19], *D. melanogaster* [7, 6], *C.elegans* [12], *H. sapiens* [15] and *H. pylori* [14]. PPI data can be obtained through a variety of sophisticated assays, like co-immunoprecipitation, yeast two hybrid, tandem affinity purification and mass spectrometry. A PPI network is usually represented by a node-labeled undirected graph where vertices correspond to proteins and edges denote physical interactions.

Since the main mechanism by which cells are able to process information is through protein-protein interactions, PPI data has been essential to obtain new knowledge and insights in a wide spectrum of biological processes. In this paper, we focus on the problem of predicting the functional category of proteins *solely* based on the topological structure of the PPI network. The rationale of this approach is based on the observation that a protein is much more likely to interact with another protein in the same functional class than with a protein with a different function (see, e.g., [10, 21, 18, 13]). The prediction of functional classes can be useful either for proteins for which there is little or non-existing functional information (e.g., for predicting the involvement of a protein in specific pathway), or to confirm existing annotations provided by other methods. Motivated by the expectation that in the near future massive PPI networks will be available, here we propose a *computationally efficient* method that accurately determines the functional categories and will be capable to scale gracefully with the size of the network.

A variety of algorithmic techniques have been proposed in the literature to solve the problem of functional prediction with a wide range of computational complexity. Perhaps the most computationally efficient algorithm is based on the *majority rule* where the function of an unknown protein is simply determined by the most common function among its interacting partners [17]. A slightly more sophisticated majority-based method is the χ^2-method proposed in [8]. At the other end of the computational complexity spectrum, the authors of [21, 10] propose to assign proteins to functional classes so that the number of protein interactions among different functional categories is minimized. The optimization problem, known as *generalized multicut*, is NP complete.

The *functional flow* algorithm introduced in [13] lays somewhere in the middle of the complexity spectrum. The idea is to treat proteins with known function as infinite sources of (functional) flow. The flow is propagated through the network in a series of discrete steps. At the end, the function of unknown proteins is assigned based on the largest amount of flow received. The authors of [13] show that functional flow algorithm outperforms the generalized multicut algorithm, the majority rule-based algorithm and also its generalization to more distant neighbors [13]. The authors of [2] show that functional flow also outperforms the χ^2-method. Because of this, the performance of functional flow is the reference for our algorithm. Experimental results will show that our method achieves a better prediction accuracy than functional flow.

Perhaps the most similar method to the one we propose here is described in [4, 5], where the authors propose a probabilistic model based on the theory of Markov random fields. In their follow-up papers [3], Deng *et al* show how to integrate in their Markov random field additional information, namely gene expression data, protein complex information, domain structures to increase the prediction accuracy. The relationship between this work and [4, 5] will be discussed in greater detail later in paper. Here, however, we want to emphasize that the method presented in this manuscript is

[*]This project was supported in part by NSF CAREER IIS-0447773, and NSF DBI-0321756.

[*]Corresponding author

BIOKDD'07 , August 12, 2007, San Jose, California, USA
Copyright 2007 ACM 978-1-59593-839-8/07/0008 ...$5.00.

computationally more efficient than Deng *et al.* Unfortunately, the accuracy of their prediction cannot be directly compared with ours because these methods predict multiple functional classes for each protein. The approach in [11] is essentially similar to [5].

More recent papers tackle slightly different albeit related problems. In [18] the authors predict functional linkages between proteins based on the integration of four kinds of evidence, namely gene co-expression, gene co-inheritance, gene co-location and gene co-evolution. In [9], the authors predict protein interactions based on the cellular localization of proteins.

2. PROBLEM DEFINITION AND MODEL FORMULATION

We denote by $G(V,E)$ the PPI network under analysis, where V represents the set of proteins and E is the set of edges (interactions). For reason that will be clear later in the paper, we assume G to be directed (i.e., each undirected edge in the original PPI is represented by two directed edges, except for self-loops). We denote the set of k given functional classes as $\mathcal{F} = \{C_1, C_2, \ldots, C_k\}$. Each functional class can be thought as one of k possible colors that can be used to color the graph. Function $f : V \to \mathcal{F}$ captures the notion of functional class for all the proteins in V. When the function of a protein $v \in V$ is known, say C_i, then we will have $f(v) = C_i$. If the function of v is unknown, then $f(v) = \emptyset$. We define $W = \{v \in V : f(v) \in \mathcal{F}\}$ to be the set of proteins whose function is known and $U = V \setminus W$ to be the set of the proteins whose function is unknown. The functional annotation problem can be informally stated as follows. Given a PPI network $G(W \cup U, E)$ where W is annotated with functional classes, find the correct functional classes for the vertices in U.

The model used here to tackle the problem is entirely probabilistic and it is based on two simple observations. First, a simple statistical analysis on the available PPI data [16] and the associated GO functional annotations [1] reveals that the distribution associated with the functional classes is highly skewed. For example, in the *S. cerevisiae* network, the function "catalytic activity" is assigned to 1,514 proteins, whereas the function "protein tag" is only assigned to 5 proteins. This observation constitutes our prior knowledge on the probability of a randomly chosen protein to perform a certain function and can be captured by the notion of *prior distribution.* We denote the prior distribution by $\mathcal{P} : \mathcal{F} \to [0,1]$, where $\mathcal{P}(C_i)$ is the probability of a randomly chosen protein to have function C_i.

Second, our model has to incorporate the connectivity structure of the PPI networks. It is well-known that a protein is more likely to interact with another protein performing the same function [10, 21, 18, 13]. We model this preference using conditional probability distributions. If protein $t \in W$ has function C_i and protein $s \in U$ interacts with t, then the probability that s performs function C_j is given by $\mathbf{P}(C_j|C_i)$. We expect $\mathbf{P}(C_i|C_i)$ to be higher than $\mathbf{P}(C_j|C_i), \forall j \neq i$, because s is more likely to perform the same function of t. This can be easily generalized to multiple interacting partners. Suppose we want to predict the function of protein $s \in U$ and that we know that $t_1, t_2, t_3, \ldots, t_m \in W$ interact with s, as well as their functions $f(t_1), f(t_2), f(t_3), \ldots, f(t_m)$. If we assume that $f(t_1), f(t_2), f(t_3), \ldots, f(t_m)$ are independent and distributed according to the conditional multinomial distribution $[\mathbf{P}(C_1|f(s)), \mathbf{P}(C_2|f(s)), \mathbf{P}(C_3|f(s)), \ldots, \mathbf{P}(C_K|f(s))]$, then the most likely function for s is the one that maximizes

$$\begin{aligned} L(s) &= \mathcal{P}(f(s)) \prod_{t \in \{t_1, t_2, \ldots, t_m\}} \mathbf{P}(f(t)|f(s)) \\ &= \mathcal{P}(f(s)) \prod_{t \in V : (s,t) \in E} \mathbf{P}(f(t)|f(s)) \end{aligned}$$

We call $L(s)$ the *local likelihood* of protein s.

Note that a necessary condition to predict the functional class for $s \in U$ is to know the functional classes of the neighbors of s. Very often, however, the functions of the neighbors turns out to be unknown. Clearly, the assignment of a function to protein s may affect the prediction of the functions for the neighbors of s, and vice versa. Because of this, a purely local strategy is insufficient. To address this problem, we need to introduce the concept *global likelihood* of a PPI Network as $L(G) = \prod_{v \in V} L(v)$.

The free variables in the global likelihood function $L(\cdot)$ are $f(u_i)$, for all proteins $u_i \in U$ with unknown function. We seek the assignment to $f(u_i)$ such that the global likelihood $L(G)$ is maximized, which is equivalent to maximizing

$$l(G) = \sum_{v \in V} \log(\mathcal{P}(f(v))) + \sum_{(v,w) \in E} \log(\mathbf{P}(f(w)|f(v)))$$

Now we are ready to give a formal summary of the optimization problem associated with our model. We are given a directed PPI network $G(W \cup U, E)$ where U is the set of proteins with unknown functions and W is the set of proteins with known functions, a set of functions $\mathcal{F}$, a prior distribution $\mathcal{P}$ with $\sum_{C_i \in \mathcal{F}} \mathcal{P}(C_i) = 1$, and the conditional distributions $\mathbf{P}(C_i|C_j)$ such that $\sum_{C_i \in \mathcal{F}} \mathbf{P}(C_i|C_j) = 1, \forall C_j \in \mathcal{F}$. The problem is to predict the functional class $f(u)$ for each protein in set U, such that the global log likelihood $l(G)$ is maximized.

3. RELATION TO PREVIOUS WORK

Our model implicitly defines a Markov random field (MRF), a probabilistic model which is also used in [4, 5]. In Deng *et al.*'s works [4, 5], a distinct MRF is built for each functional class in $\mathcal{F}$. Each protein in the PPI network is associated to an indicator random variable for that function of interest. More specifically, each protein is associated with a unary potential $e^{\phi(X_i)}$, which has value $e^{\phi(1)}$ if the protein has that function and $e^{\phi(0)}$ otherwise. Each edge of the PPI graph is associated with a binary potential $e^{\psi(X_i,X_j)}$, which can take three possible values, namely $e^{\psi(1,1)}$ if both of the proteins have the function, $e^{\psi(0,1)}$ if one of the proteins has the function, and $e^{\psi(0,0)}$ if neither of the proteins has the function. Given the parameters $\theta = \{\phi(0), \phi(1), \psi(1,1), \psi(0,1), \psi(0,0)\}$, the global Gibbs distribution of the entire network is simply the product of the unary potentials and the binary potentials normalized by a constant factor depending on the parameters, as follows.

$$P\{X_1, X_2, X_3, \ldots, X_n|\theta\} = e^{\sum_{i=1}^{n} \phi(X_i) + \sum_{(i,j) \in E} \psi(X_i,X_j)}/Z(\theta)$$

Note that in our model, the prior probability $\mathcal{P}(f(v_i))$ corresponds to the unary potential in Deng's model, whereas the product $\mathbf{P}(f(v_i)|f(v_j))\mathbf{P}(f(v_j)|f(v_i))$ corresponds to the binary potential.

Despite the similarities, there are significant differences between Deng *et al.*'s model and ours. First, instead of building a distinct MRF for each function, we only have one unified probabilistic model for all the functions in $\mathcal{F}$ which allows us to capture the correlations between the functions. Second, the use of conditional distributions dramatically simplifies the process of estimating the parameters, which boils down to a simple count of relevant statistics (details to be explained in Section 4). The semantics of the conditional distributions also naturally gives rise to the efficient iterative algorithm that we will develop later. Finally, since we are modeling from the conditional distributions, the normalization factor of the global Gibbs distribution in our model is always one irrespective of the parameters we use.

A less obvious connection can be established between our model and the generalized multi cut approach by Vazquez *et al.* [21]. Recall that in this latter approach, the objective is to assign functional annotations to unknown proteins in such a way that one minimizes the number of times neighboring proteins have different annotations. A formal description of the generalized multi cut problem follows. Let I be the standard indicator function which is equal to 1 if the boolean expression is true and 0 otherwise. Given a PPI network $G(U \cup W, E)$ we seek annotations to the proteins in U such that $\sum_{(u,v)\in E} I(f(u) \neq f(v))$ is minimized.

FACT 1. *The generalized multi cut problem is a special case of our optimization problem when the prior distribution is uniform and most of the mass of the conditional probabilities is concentrated around* $\mathbf{P}(C_i|C_i)$.

Proof. Let us consider the following prior distribution and conditional distributions.

$$\begin{aligned}
\mathcal{P}(C_i) &= 1/|\mathcal{F}| && \forall C_i \in \mathcal{F}\\
\mathbf{P}(C_j|C_i) &= \epsilon && \forall C_i, C_j \in \mathcal{F}, C_i \neq C_j\\
\mathbf{P}(C_i|C_i) &= 1-(|\mathcal{F}|-1)\epsilon && \forall C_i \in \mathcal{F}
\end{aligned}$$

where $0 < \epsilon < 1$ is an arbitrarily small number. Then, the global log likelihood for the graph can be written as

$$\begin{aligned}
&l(G(V,E))\\
&= \sum_{v\in V} \log(\mathcal{P}(f(v))) + \sum_{(v,w)\in E} \log(\mathbf{P}(f(w)|f(v)))\\
&= \sum_{v\in V} \log(1/|\mathcal{F}|) + \sum_{(v,w)\in E f(w)\neq f(v)} \log(\mathbf{P}(f(w)|f(v)))\\
&\quad + \sum_{(v,w)\in E f(w)=f(v)} \log(\mathbf{P}(f(w)|f(v)))\\
&= |V|\log(1/|\mathcal{F}|) + \sum_{(v,w)\in E f(w)\neq f(v)} \log(\epsilon)\\
&\quad + \sum_{(v,w)\in E f(w)=f(v)} \log(1-(|\mathcal{F}|-1)\epsilon)\\
&= |V|\log(1/|\mathcal{F}|) + |E|\log(1-(|\mathcal{F}|-1)\epsilon) \qquad (1)\\
&\quad +(\log(\epsilon) - \log(1-(|\mathcal{F}|-1)\epsilon)) \sum_{(v,w)\in E} I(f(v) \neq f(w))
\end{aligned}$$

Note that the first two terms of (2) are constant and that the third term increases as the quantity $\sum_{(v,w)\in E} I(f(v) \neq f(w))$ decreases because $\log(\epsilon) - \log(1-(|\mathcal{F}|-1)\epsilon)$ is negative for a sufficiently small ϵ. Therefore, under this particular prior distribution and conditional distributions, maximizing the global log likelihood in our problem is equivalent to minimizing the objective function in the generalized multicut problem. $\blacksquare$

The generalized multicut problem is NP complete [13] because it is a generalization of the multi-way cut problem [20], which is known to be NP complete. Since our problem is a generalization of the generalized multicut problem, it is NP complete as well.

4. PARAMETER LEARNING

The prior distribution and the conditional distributions are multinomial distributions whose parameters can be learned from the structure of the given PPI network and the functional annotations on W. We need to determine $k-1$ parameters for the prior and $k(k-1)$ parameters for the k conditional distributions. We obtain these parameters using the maximum likelihood estimation method.

Let $F(W,E')$ be the subgraph of $G(V,E)$ induced by the set W of known functions, where $E' = \{(u,v)|(u,v) \in E, u \in W, v \in W\}$. The global likelihood for the subgraph $F(W,E')$ is defined as follows.

$$\begin{aligned}
&L(F(W,E'))\\
&= \prod_{v\in W} \mathcal{P}(f(v)) \prod_{(u,v)\in E'} \mathbf{P}(f(v)|f(u))\\
&= \prod_{C_i\in\mathcal{F}} \mathcal{P}(C_i)^{\sum_{v\in W} I(f(v)=C_i)} \qquad (2)\\
&\quad \prod_{C_i\in\mathcal{F}} \prod_{C_j\in\mathcal{F}} \mathbf{P}(C_j|C_i)^{\sum_{(v_i,v_j)\in E'} I(f(v_i)=C_i, f(v_j)=C_j)}
\end{aligned}$$

The first term in (3) is maximized when $\mathcal{P}(C_i) = \sum_{v\in W} I(f(v) = C_i)/|W|$ for all $C_i \in \mathcal{F}$. The second term in equation (3) is maximized when $\mathbf{P}(C_j|C_i) = \frac{\sum_{(v_i,v_j)\in E'} I(f(v_i)=C_i, f(v_j)=C_j)}{\sum_{(v_i,v_j)\in E'} I(f(v_i)=C_i)}$ for all $C_j \in \mathcal{F}$. Therefore, the maximum likelihood estimates for the parameters are

$$\begin{aligned}
\mathcal{P}(C_i) &= \sum_{v\in W} I(f(v) = C_i)/|W| \quad C_i \in \mathcal{F}\\
\mathbf{P}(C_j|C_i) &= \frac{\displaystyle\sum_{(v_i,v_j)\in E'} I(f(v_i) = C_i, f(v_j) = C_j)}{\displaystyle\sum_{(v_i,v_j)\in E'} I(f(v_i) = C_i)} \quad C_i, C_j \in \mathcal{F}
\end{aligned}$$

As a common practice in Bayesian statistics, we apply (uniform) Dirichlet priors to our estimators. This prevents the problem of handling zero probabilities. The time complexity of the learning phase is $O(|E| + |W|)$, whereas the space complexity is $O(k^2)$.

5. INFERENCE OF FUNCTIONAL CLASSES

Since we determined that our problem is NP complete, it is rather unlikely that we will find a polynomial time algorithm that can solve the problem optimally. To this end, we designed a statistically based iterative algorithm (SBIA for short), which turns out to perform well in practice. Our algorithm consists of two phases, namely the initialization phase and the iterative phase. The initialization phase consists of two steps. In the first step, we estimate the parameters for the prior distribution and the conditional distributions as described in Section 4. In the second step, we assign an initial functional class to each protein in V, as follows.

For each unknown protein $v \in U$, we assign

$$f^0(v) = argmax_{C_i \in \mathcal{F}} \ \mathcal{P}(C_i) \prod_{(v,t)\in E, t\in W} \mathbf{P}(f^0(t)|C_i).$$

In other words, we predict the initial function for v to be the one that maximizes the local likelihood of v (ignoring neighbors with unknown functions). If $v \in W$, then we set $f^0(v)$ to be the function corresponding to annotation in the original data.

In the second phase, we iteratively re-evaluate our predictions. For clarity of exposition we use superscripts to denote the iteration number, i.e., $f^n(v)$ denotes the predicted functional class for v made in the n^{th} iteration. For each unknown protein $v \in U$, we set

$$f^n(v) = argmax_{C_i \in \mathcal{F}} \ \mathcal{P}(C_i) \prod_{(v,t)\in E} \mathbf{P}(f^{n-1}(t)|C_i).$$

That is, we adjust our prediction for protein v to be the function that maximizes the local likelihood with respect to the functions predicted for its neighbors in the previous step. Again, if $v \in W$, then $f^n(v) = f^{n-1}(v)$.

We stop the iterative process as soon as the difference between the value of the global likelihood in two consecutive steps drops below a given threshold. The pseudo-code in Figure 1 summarizes the algorithm. The time complexity of the algorithm is $O(d|E|)$, where d represents the number of iterations (usually $d \leq 5$ in our experiments).

6. EXPERIMENTAL RESULTS

The dataset used in our experimental studies is the most well-characterized PPI network available at the time of writing, namely the network for *S. cerevisiae*, which is composed of 4,959 proteins and 17,511 interactions. The network was obtained from the DIP database [16]. We also extracted a *high confidence* yeast PPI network, which is a subset of the yeast PPI network in which interactions that are confirmed by only a single experiment have been removed. This latter network has 1,735 proteins and 2354 interactions. The functional annotations were obtained from the Gene Ontology (GO) hierarchy [1].

We used cross validation to quantitatively evaluate the prediction accuracy of our algorithm and to compare its performance with other methods. In each experiment, we randomly removed the functional annotation to a percentage p of known proteins, where p ranges from 5% to 95%. This new set of "unknown" proteins served as the test set, called hereafter T. We use $W \setminus T$ to denote the set of known proteins after p% of them have been "un-labelled" and U to denote the set of the remaining unknown proteins. Clearly, the SBIA's learning phase (i.e., the computation of the prior and the conditional probabilities) is carried out only on the proteins in $W \setminus T$. Learning on the original set W would constitute "cheating".

So far, in our model we assumed that each protein can perform only one function. This is, however, not true for some proteins. A protein may participate in multiple biological processes and as a result, it will carry out multiple functions. In the yeast network, 488 proteins out of 3,022 are annotated with two or more top level functions. To handle this issue, the nodes in $W \setminus T$ that are associated with multiple functions are replicated, so that each copy carries out exactly one of the annotated functions. Each copy has the same interaction partners of the original protein.

As said, the goal is to predict a function for each of the proteins in set $T \cup U$, based on the functional classes in $W \setminus T$ and the topology of the graph. For each protein in T, we declare a prediction to be correct if the predicted function is one of the functions the protein was originally assigned. The prediction accuracy is calculated as the ratio between the number of correct predictions and the total number of proteins in the set T. Since the prediction accuracy varies slightly every time we randomly select T, we replicate the same experiments ten times and compute the average accuracy. We also record the standard deviation, represented by the error bars in the figures.

We compared the accuracy of our method against that of functional flow [13] and against that of the *naive* approach. We chose to compare SBIA against the functional flow method because papers [2, 13] report that functional flow outperforms both majority-rule based methods [17, 8] as well as methods based on the generalized multicut [21, 10]. As said, a direct comparison between our method and MRF-based methods [4, 5, 11] is not feasible because these latter approaches predict more than one functional class for each protein. The naive method simply predicts the function of a protein to be the most probable functional class according to the prior, i.e., $argmax_{C_i \in \mathcal{F}} \mathcal{P}(C_i)$. Clearly, the expected prediction accuracy of the naive approach is equal to the ratio between the number of proteins annotated with the most probable function and the total number $|W|$ of known proteins.

We carried out two sets of experiments. In the first set, we considered the seventeen top level molecular functions defined in GO. In the yeast PPI network, 3,022 proteins out of 4,959 are annotated with one or more top level functions. The most frequent function is "catalytic activity", which occurs 1,514 times. Thus, the expected prediction accuracy for the naive approach is 0.501 or 50%. In the high confidence yeast PPI network 1,325 proteins are annotated. The most frequent function in this network is again "catalytic activity", which is assigned to 568 proteins. The statistics of the networks constituting the dataset are summarized in Table 1.

Figure 2-left and 3-left summarize the results of the first set of experiments on the seventeen functional classes in the top level of the GO hierarchy. The figures show that SBIA always outperforms functional flow, especially when p is large. In the yeast network, the prediction accuracies of the functional flow algorithm even falls below that of the naive approach when p is greater than 55%. SBIA, however, still retains good prediction accuracy until p becomes higher than 70%, and then asymptotically converges to that of the naive approach. Notice that the initialization phase of SBIA already achieves a good prediction accuracy. When p is less than 80%, the iterative phase improves the prediction accuracy even more, along with the global likelihood of the graph. The number of iterations executed is usually rather small, less than 5. When p is greater than 80%, the information left in the network is highly incomplete, and as expected the performance of our algorithm falls back to

```
SBIA:
  • Input:
      1. $G(V, E)$, where $V = U \cup W$. $W$ is the set of known problems and $U$ is the set of unknown proteins.
      2. $\mathcal{F}$, the set of functions.
      3. $f : W \to \mathcal{F}$, the annotations on the proteins in $W$.
  • Output:
      1. $f : U \to \mathcal{F}$, the predicted function for the proteins in $U$.
  • Initialization phase
      1. Estimate $Pri(C), P(C_i|C_j), C, C_i, C_j \in \mathcal{F}$ as suggested in section 4.
      2. For $v$ in $V$:
             IF $(v \in U)$ $f(v) = argmax_{f(v) \in \mathcal{F}} Pri(f(v)) \prod_{(v,t) \in E, t \in W} P(f(t)|f(v))$ ;
  • Iterative phase
      1. DO:
             FOR $v$ in $W$: $f'(v) = f(v)$
             FOR $v$ in $U$: $f'(v) = argmax_{f'(v) \in \mathcal{F}} Pri(f'(v)) \prod_{(v,t) \in E} P(f'(t)|f'(v))$
             $L(G) = (\prod_{v \in V} Pri(f(v))) \cdot (\prod_{(v,w) \in E} P(f(w)|f(v)))$
             $L'(G) = (\prod_{v \in V} Pri(f'(v))) \cdot (\prod_{(v,w) \in E} P(f'(w)|f'(v)))$
             IF $L'(G) >= L(G)$:
                 FOR $v$ in $V$: $f(v) = f'(v)$
         WHILE $(L'(G) > L(G))$
      2. RETERN $f : U \to \mathcal{F}$
```

Figure 1: Pseudo code of our Statistically Based Iterative Algorithm(SBIA)

Table 1: The statistics of the PPI networks used in the experiments. $|V|$ is the number of proteins in the network, $|E|$ is the number of interactions, $|W|$ is the number of known proteins, and *naive expected* is the expected prediction accuracy of the naive approach (see text).

			17 functional classes		190 functional classes	
organism	$\|V\|$	$\|E\|$	$\|W\|$	*naive expected*	$\|W\|$	*naive expected*
yeast	4,959	17,511	3,022	0.5010	2930	0.1939
yeast high confidence	1,735	2,354	1,325	0.4286	1278	0.1979

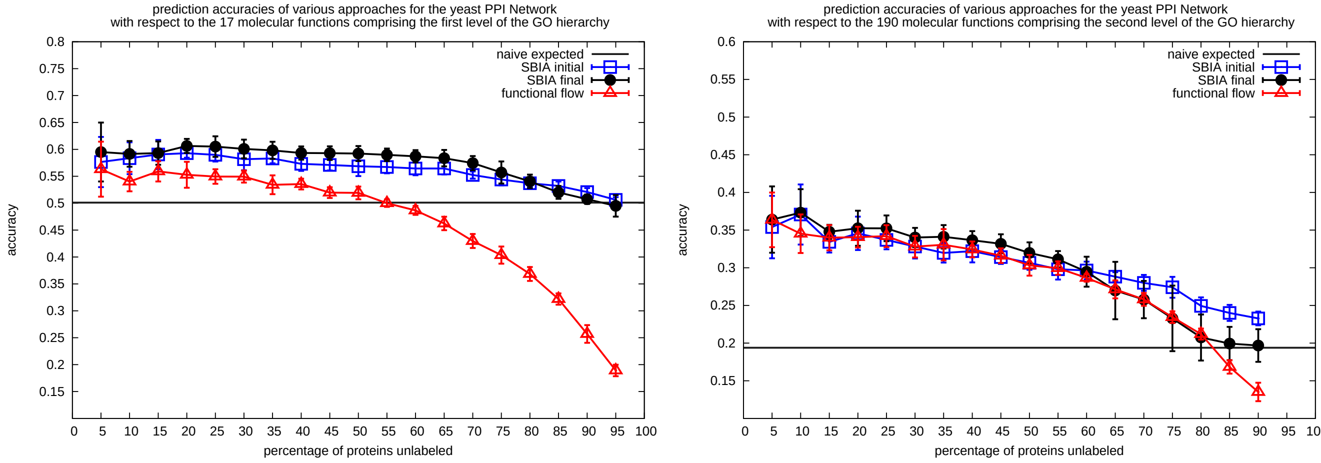


Figure 2: Prediction accuracies on the yeast PPI network with respect to the 17 functional classes at the first level of the GO hierarchy (right) and 190 functional classes at the second level of the GO hierarchy (left). The x-axis represents the percentage of known proteins on which the algorithms are tested. The "naive expected" line indicates the expected prediction accuracy of the naive approach. "SBIA initial" refers to the accuracy of SBIA after the initialization phase, whereas "SBIA final" shows the final accuracy of SBIA. "Functional flow" denotes the prediction accuracy of the functional flow algorithm

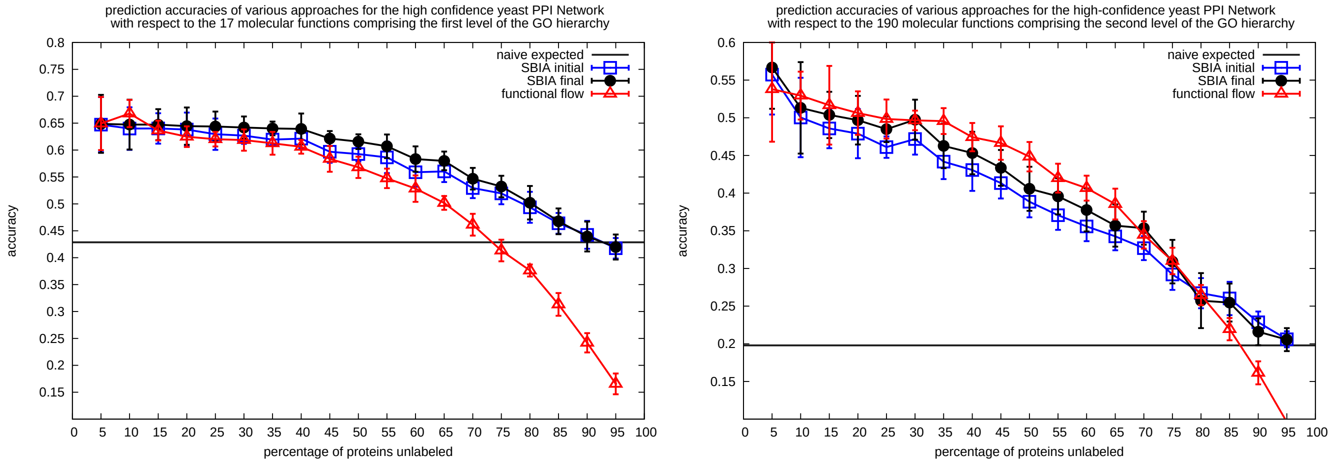


Figure 3: Prediction accuracy on the yeast high confidence PPI network (see caption of Figure 2 for more details). LEFT: 17 functional classes, RIGHT: 190 functional classes.

that of the naive approach. Due to the higher quality of the data in the yeast high confidence network, the improvement in accuracy of our algorithm and functional flow relative to the naive approach is almost doubled.

In the second set of experiments, we considered all the 190 molecular functions comprising the second level of the GO hierarchy. In the yeast network, 2,930 proteins out of 4,959 yeast proteins are annotated with one or more second level molecular functions. The most prevalent function is "hydrolase activity", which appears 568 times. Hence the expected prediction accuracy for the naive approach is 0.1939. In the high confidence yeast network, 1,278 out of 1,735 proteins are annotated. The most prevalent function is "protein binding", which is annotated to 253 proteins. The statistics are summarized in Table 1.

Figure 2-right and 3-right summarize the second set of experimental results. In Figure 3-right, the functional flow algorithm outperforms SBIA by 2-3% on average. We suspect that this is due to the relatively small size of the network (containing about 1,300 characterized proteins) under consideration and the large number of functions ($k = 190$). Recall that the number of parameters of our model is $\Theta(k^2)$. In this case, we believe that there is not enough data for the accurate estimation of the parameters for the prior distribution and the conditional distributions. For the yeast PPI network, the result is similar to that in the previous set of experiments. SBIA still outperforms functional flow, but the difference between the two approaches is not as strong as in the previous case.

7. CONCLUSIONS

We developed an efficient algorithm to assign functional GO terms to uncharacterized proteins on a PPI network based solely on the topology of the graph and the functional labels of known proteins. The statistical model proposed in this paper is a generalization of the GenMultiCut model and resemble the MRF-based model by Deng *et.al.* The similarity with the work of Deng *et.al.* is, however, superficial as we discussed in details in the paper. In particular, the structure of our model allows one to obtain easily and efficiently the maximum likelihood estimation of the underlying parameters, which is tipically not possible for a general MRF. Based on our statistical model, we presented efficient learning and inference algorithms. Our inference algorithm is an iterative algorithm, where each iteration runs in time linear in the size of the input. According to our experimental results, our algorithm converges very quickly to a local optimum. More importantly, our method gives consistently better predictions when compared with previous known algorithms.

8. REFERENCES

[1] Ashburner, M., Ball, C. A., and *et al.* Gene ontology: tool for the unification of biology. *Nature Genetics 25* (2000), 25–29.

[2] Chua, H. N., Sung, W.-K., and Wong, L. Exploiting indirect neighbours and topological weight to predict protein function from protein-protein interactions. *Bioinformatics 22* (2006), 1623 – 1630.

[3] Deng, M., Chen, T., and Sun, F. An integrated probabilistic model for functional prediction of proteins. *Journal of Computational Biology 11*, 2/3 (2004), 463–475.

[4] Deng, M., Tu, Z., Sun, F., and Chen, T. Mapping gene ontology to proteins based on protein-protein interaction data. *Bioinformatics 20*, 6 (2004), 895–902.

[5] Deng, M., Zhang, K., Mehta, S., Chen, T., and Sun, F. Prediction of protein function using protein-protein interaction data. *Journal of Computational Biology 10*, 6 (2003), 947–960.

[6] Formstecher, E., Aresta, S., and *et al.* Protein interaction mapping: A drosophila case study. *Genome Res. 15*, 3 (2005), 376–384.

[7] Giot, L., Bader, J. S., and *et al.* A protein interaction map of *Drosophila melanogaster. Science 302*, 5651 (2003), 1727–1736.

[8] Hishigaki, H., Nakai, K., Ono, T., Tanigami, A., and Takagi, T. Assessment of prediction accuracy of protein function from protein–protein interaction data. *Yeast 18*, 6 (2001), 523–531.

[9] Jaimovich, A., Elidan, G., Margalit, H., and Friedman, N. Towards an integrated protein-protein interaction network. In *Proceedings of ACM RECOMB* (2005), pp. 14–30.

[10] Karaoz, U., Murali, T. M., Letovsky, S., Zheng, Y., Ding, C., Cantor, C. R., and Kasif, S. Whole-genome annotation by using evidence integration in functional-linkage networks. *Proc Natl Acad Sci U S A 101*, 9 (2004), 2888–2893.

[11] Letovsky, S., and Kasif, S. Predicting protein function from protein/protein interaction data: a probabilistic approach. *Bioinformatics 19*, 1 (2003), i197–i204.

[12] Li, S., Armstrong, C., and *et al.* A map of the interactome network of the metazoan *C. elegans. Science 303* (2004), 540–543.

[13] Nabieva, E., Jim, K., Agarwal, A., Chazelle, B., and Singh, M. Whole-proteome prediction of protein function via graph-theoretic analysis of interaction maps. In *Proceedings of ISMB* (2005), pp. 302–310.

[14] Rain, J., Selig, L., and *et al.* The protein-protein interaction map of *Helicobacter pylori. Nature 409* (2001), 211–215.

[15] Rual, J.-F., Venkatesan, K., and *et al.* Towards a proteome-scale map of the human protein-protein interaction network. *Nature 437* (2005), 1173–1178.

[16] Salwinski, L., Miller, C. S., Smith, A. J., Pettit, F. K., Bowie, J. U., and Eisenberg, D. The database of interacting proteins: 2004 update. *Nucleic Acids Research 32* (2004), D449.

[17] Schwikowski, B., Uetz, P., and Fields, S. A network of protein-protein interactions in yeast. *Nature Biotechnology 18* (2000), 1257 – 1261.

[18] Srinivasan, B. S., Novak, A. F., Flannick, J. A., Batzoglou, S., and McAdams, H. H. Integrated protein interaction networks for 11 microbes. In *Proceedings of ACM RECOMB* (2006), pp. 1–14.

[19] Uetz, P., Giot, L., and *et al.* A comprehensive analysis of protein-protein interactions in *Saccharomyces cerevisiae. Nature 403*, 6770 (2000), 623–627.

[20] Vazirani, V. V. *Approximation algorithms.* Springer-Verlag New York, Inc., New York, NY, USA, 2001.

[21] Vazquez, A., Flammini, A., Maritan, A., and Vespignani, A. Global protein function prediction in protein-protein interaction networks. *Nature Biotechnology 21* (2003), 697.

Profile-feature Based Protein Interaction Extraction from Full-Text Articles

Shilin Ding[1] Minlie Huang[1] Hongning Wang[1] Xiaoyan Zhu[1,*]

[1]State Key Laboratory of Intelligent Technology and Systems (LITS),
Department of Computer Science and Technology, Tsinghua University,
Beijing, 100084, China

Email: dingsl@gmail.com, {aihuang,zxy-dcs}@tsinghua.edu.cn, whn03@mails.tsinghua.edu.cn

ABSTRACT

Various methods have been proposed to extract genetic protein-protein interactions from abstracts. These methods are unable to specify the interactions in which molecules are physically related and fail to explore the abundant evidence all over the articles. In this paper, we present a method of mining physical protein-protein interactions by exploiting profile feature from full-text articles during our participation in the second task of BioCreAtIvE Challenge 2006. This method synthesizes the features from the whole article as the protein pair's profile to extract the physical interactions, and specifies the SwissProt AC of the molecules involved in the interaction to help biologists make use of the information of the molecules, such as the sequence and cross reference. Compared with the other methods' performance released in BioCreAtIvE 2006, our method has shown very promising results.

Categories and Subject Descriptors

J.3 [**Life and Medical Sciences**]: Biology and Genetics; I.5.4 [**Pattern Recognition**]: Applications – *Text Processing*

General Terms

Algorithms, Experimentation

Keywords

Protein-Protein Interaction, Text Mining, Information Extraction

1. INTRODUCTION

The study of Protein-Protein Interaction (PPI) is one of the most pressing problems. Characterizing protein interaction partners is crucial to understanding not only the functional role of individual proteins but also the organization of entire biological processes. In the past years, the high throughput technologies have generated large amount of information. However, the information is buried in millions of peer-reviewed literatures. Without efficient management, the biological knowledge in the literatures is of little use to the researchers. A lot of knowledge databases, such as BIND [1], IntAct [11], and MINT [28] have been constructed to this end, but it costs a lot of time and expense to manually review and extract the important information from the literatures. So, automatically mining protein-protein interactions from bioscience literature is crucial and challenging [16].

There are two types of protein interactions: *Genetic Interaction* which is functional relationship among genes revealed by phenotype of cell, and *Physical Interaction* which is interaction among molecules. The task we participated in BioCreAtIvE 2006 is focused on mining physical interactions from the text because the genetic interactions are 1) not direct (the interaction may be through signaling cascades), thus, 2) not always trustworthy for biologists [30]. The abstracts with concentrated and limited information from MEDLINE are not capable to provide enough information to accomplish this task, while the full-text articles are more comprehensive to provide the evidence, such as the biological experiment which verifies the existence of the physical interaction. So the major problem here is how to exploit the physical interactions from the evidence synthesized from the full-text articles.

Various methods have been proposed to extract protein-protein interaction. But most of them are focused on abstract and fail to differentiate the physical interaction from the genetic interaction. In this paper, we describe a profile-feature based method to mine physical protein-protein interactions by exploiting abundant features from full-text articles.

The paper is organized as follows: The related works are discussed in Section 2. Section 3 presents the method to recognize the protein molecule names in text and normalize to them to entries in SwissProt. The profile-feature based method to extract the physical interactions from the evidence of the whole article is discussed in Section 4. In Section 5, we show the experiment and evaluation. And we draw our conclusions and discuss the future work in Section 6.

2. RELATED WORK

The researches of exploiting the information from the full-text articles are limited due to full texts' availability and complexity. SGPE [27] used abstracts and full-text articles to extract gene and protein synonyms, and Yu reported that the system performs

BIOKDD'07, August 12, 2007, San Jose, California, USA.

[*] Corresponding author: zxy-dcs@tsinghua.edu.cn
Tel: 86-10-62796831 Fax: 86-10-62782266

better on full-text articles because the names are more frequently listed in full-text articles. Schuemie [24] in their study of information content in abstracts versus that in full-text articles argued that the information density is higher in abstracts but the information coverage is much greater in full-text articles which indicates that the IE tools will perform better with the various information resources in the full-text articles. And Natarajan [20] used text mining of full-text articles to help generate novel hypothesis for the guide of gene-relation detection experiment and argued that the full-text articles are more comprehensive than the abstracts. So, the previous studies showed that the full-text articles are more effective for the extraction of physically interacted protein pairs.

Various methods and systems [3, 5, 7, 9, 13, 14, 19, 21] have been proposed for protein interaction extraction, but few of them are focused on physical interactions by exploring the evidence synthesized from the full-text articles. One class of these approaches is based on machine learning models. For example, Craven [4] employed a Naïve Bayes Classifier to predict relations from sentences.

Another class of methods for relation extraction is rule-based or pattern-based. The simplest method of this category is to extract relations from co-occurrence of entities in sentences [6, 15]. This method generates high sensitivity but low specificity.

Pattern based methods adopt hand-coded or automated patterns and then use pattern matching techniques to capture relations. Ono [21] manually constructed lexical patterns to match linguistic structures of sentences for extracting protein interactions. Similar hand-coded pattern based systems were also proposed by Rindflesch [23] and Pustejovsky [22]. Such methods contribute high accuracy but low coverage, and moreover, the construction of patterns is time-consuming and requires much domain expertise. Methods which can learn patterns automatically for general relation extraction include SPIES [14], ONBIRES [13, 7], Chiang [3], and Daraselia [5]. Most of them take annotated texts as input, and then learn patterns semi-automatically (starting from some pattern seeds) or automatically. Most of these methods focus on extracting one specific type of relations and can only explore the information confined in one sentence.

The third class of methods analyzes the syntax structures and semantics of the sentences to extract the relations [9]. This method strongly rely on the Natural Language Processing techniques, such as dependence parse trees [18], to get the structure of a particular sentence. This method has promising performance and is able to extract deeper semantic relations from the text. But it is also focused on single sentence and fails to explore the evidence from the whole articles.

In this paper, we describe a method to mine physical protein-protein interactions by exploiting abundant features. A profile-feature based method is adopted to extract the physical interactions from the full-text articles. Every sentence where the candidate molecule pairs co-occur is considered as a piece of evidence. And the profile, which is defined as the representation of the pair's features all over the article, is constructed based on all of the evidence. Thus, the method is able to exploits the document-level information instead of focusing on the features on sentence level. Here, we use SVM for training and classifying.

Although the information from the whole article is exploited, another difficulty facing physical interaction extraction is how to recognize the molecules in the articles. Since the physical interaction is the interaction between molecules, the identified names should be normalized to entries in a standard database, such as SwissProt. Thus, the biologists can easily get the whole information of the molecules, such as the sequence and taxonomy information, or other abundant cross-reference information.

Previous Named Entity Recognition methods [8, 25, 26, 29] can find out the protein names, but fail to specify what exact molecules these names refer to. The statistical based method is the most prevalent method to recognize named entities in the text. It exploits abundant word form features and context features to train a model [29, 25]. It has promising performance and flexibility but needs a large scale of annotated corpus. The rule based method is fast and highly accurate in a specific domain, but costs a lot of efforts to construct the rules [8]. These two methods are unable to normalize the names to database entries because the lack of reference to protein database. And the dictionary based method has the potential to map the names to the database entries, but the previous ones are only focused on find out the names.

The difficulty is due to extensive ambiguity in names and overlap of names with common English terms [12]. The use of phenotypic description, the conventional abbreviations lead to various synonyms that are difficult to differentiate. Our Named Entity Recognition and Normalization (NER/N) method is a dictionary matching method based on the organism information from the full-text article. We curated the SwissProt database to boost coverage and accuracy of the terms in the database. Then various rules are applied to solve naming convention related problem. The organism information is used to improve the NER/N process in terms of both time and accuracy.

Our contributions in this paper include 1) the novel NER/N method based on the organism information from the full-text article to recognize the protein name and specify the corresponding entry in SwissPort; and 2) the profile-feature based method which exploits the evidence all over the article to extract the physical interaction. In comparison to the average performance of all the submitted runs in BioCreAtIvE 2006, our method shows promising results and is ranked top in the official evaluation.

3. NAMED ENTITY RECOGNITION AND NORMALIZATION (NER/N)

Different from traditional NER, this task requires the protein names be normalized to primary Access Numbers (AC) of SwissProt entries, not just find the original names in the text. The motivation of this task is to help biologists identify the exact molecule of the mentioned protein, so they can use other information of the molecule, such as the sequence and taxonomy, and cross-reference information like protein structure. The major problem here is how to associate the name in the article with the entry in SwissProt.

- First, the inconsistent naming conventions and various usages in text cause a lot of ambiguous terms. For example, *TCF*, *PAL*, and *PKB* may refer to different entities.

- Second, abbreviated terms, such as *p53*, may cause difficulty for normalization, although domain experts can infer from the context what molecules the author is discussing.
- Third, the same protein name is used to identify different molecules that are from the same or related gene but different organisms. For example, *PI3K* may refer to different molecules in mouse (**P42337**), human (**P42336**), bovine (**P32871**), produced by the same gene *PIK3CA*.
- Fourth, the same protein name is used to identify different molecules of different isoforms. For example, PI3K is referred to **Q8BTI9** which is the beta isoform of the protein in mouse, and **O35904** which is the delta isoform.

As shown in Figure 1, there are mainly four processes in this module: 1) database curation; 2) organism detection; 3) dictionary-matching based name recognition; and 4) normalized names disambiguation. The process is as follows:

1) The SwissProt is curated to incorporate gene names/synonyms and unify the written form;
2) Find all the organisms that are mentioned in the article, mark their positions as an index;
3) The organism list is used to filter out irrelevant SwissProt entries for the matching of current article;
4) The article is processed by the same unification rules and matched by the filtered entries;
5) Disambiguate the multi-mapped names by the organisms in the context

3.1 Database Curation

During database curation, two main procedures below are done to improve the quality and coverage of the terms in SwissProt database:

- Curate entry terms in the SwissProt entries. The gene names/synonyms, gene product names/synonyms of the same entry are included. Addition of gene names may cause ambiguity since a gene may encode several proteins.
- Unify the written form of the entry terms based on rules. The same rules are applied to articles to maintain consistency.
  1) Prefixes and suffixes which are not critical for entity identification are removed. For example, prefix *c*, *n* and *a* of PKC, known as Protein Kinase C, which mean *conventional*, *novel* and *atypical* respectively, are removed.
  2) Terms with digits or Roman/Greek numbers are transformed into a unified format: Alphabet + white space + digits. This rule implies such normalization: IL-2, IL2, IL 2→IL 2; CNTFR alpha, CNTFR A, CNTFR I→CNTFR 1.
  3) Terms not in abbreviated forms are converted to lowercases.

The curation helps to improve the coverage because the official SwissProt names are descriptive and too long to use in articles. And it also helps to solve the nonstandard writing habits due to the rule-based unification.

3.2 Dictionary Matching

After curation, there are totally 230,000 entries, and more than 1 million terms. Obviously, it is not feasible for all the terms to be used during dictionary matching with the articles. To improve computation efficiency, we first detect the organisms in an article, and then use the information to rule out irrelevant entries. Our assumption here is that physical interactions described in one article would belong to a limited number of organisms. The organism database used as the controlled vocabulary is NCBI taxonomy [31]. A dictionary matching method is used to detect organisms, and five most frequent organisms are left, marked with their positions in the article. When matching the articles with SwissProt to find the ACs of the protein names mentioned, only the entries belonging to these organisms are used.

In experiment, the matching process saves about three quarters

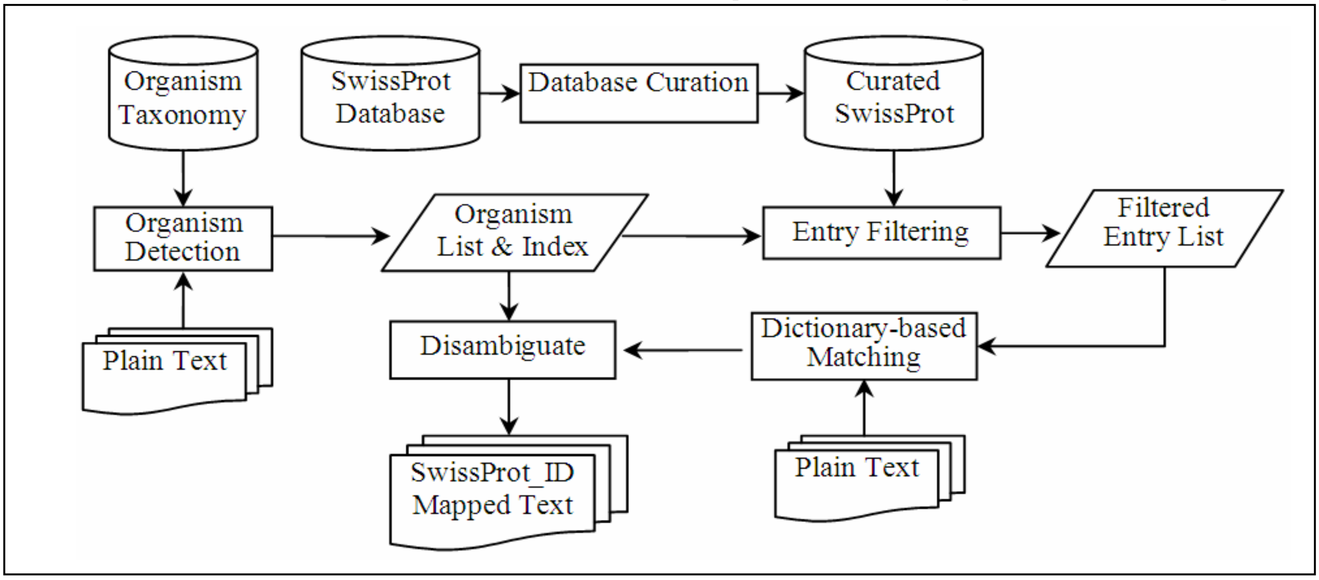


Figure 1: The Flowchart of NER. First, curate the terms in the SwissProt database; second, find the names and map them to SwissProt entries; and third, disambiguate the multi-mapped names by zone of control information from the organism contexts.

of the time due to the filtering. The time consumed by matching 740 articles with all entries is 460 minutes on a normal Pentium 4 2.0G processor. Through the filtering process before dictionary matching, the time is reduced to 125 minutes in the same condition.

3.3 Disambiguation

One protein name, particularly in abbreviated form, may correspond to multiple SwissProt entries. This is common in cases when the gene products in different organisms are similar (refer to the 3rd and 4th NER problems in Section 2). To solve the disambiguation, the principle of nearest neighbor is used, based on the organism's zone of control. The presumption here is that every protein name belongs to a particular organism's context. This context can be determined by the organism's zone of control (ZOC): beginning from the sentence that mentions the organism till the sentence that mentions another organism. When a multi-mapped name is met, we calculate which organism's zone the name belongs to based on the nearest neighbor rule, and filter out other maps to SwissProt entries with different organisms.

The disambiguation can't solve the isoform problems because the name is mapped to different isoforms that belong to the same organism. However the method is efficient because the isoform problems are not prevalent. We will see later in the experiment that this disambiguation method improves the precision greatly with only a little loss in recall.

From the discussion above, it can be inferred that our NER/N method outperforms other methods because: 1) carefully designed curation greatly improves the database's coverage and eliminates lots of naming inconsistency due to writing habit; 2) the dictionary matching method efficiently maps the name to the SwissProt entries based on the organism information from the full-text article.

4. PROFILE-FEATURE BASED EXTRACTION

Previous methods to extract protein interactions are based on sentence level, thus fail to synthesize the information from the whole articles. However, the topic-level interactions will be discussed at several places across the article, and these places will provide different sources of evidence, such as the experiment support and cross-reference evidence. The basic idea here is to extract interactions by using profile features derived from the whole document. The classifier is trained to make the decision based on the features all over the article. The profile-feature based extraction is more robust than pattern based extraction and other methods focused on the evidence from single sentence.

First, the goal is to extract physical interactions, so the single description as "*PTN1* binds to *PTN2*" does not necessarily indicate the existence of a physical interaction between *PTN1* and *PTN2*. However, if there is other evidence in the document, such as "The bind of *PTN1* to *PTN2* is determined by two hybrid screen", then the interaction is more probably to be true. So, different evidence will strengthen the validation of the physical interaction.

Second, the profile-feature based extraction is more robust when NER performance is far from satisfactory. The false positive protein names will falsely pair with other recognized names. But the pairs of the false positive proteins will be less statistically significant all over the document. Their profile features will be more random and less significant. For example, "The Y2H experiment proved the interaction between *PTN1* and *PTN2*, *CGA* ... …". The underlined term "*CGA*" that is the sequence of *PTN2* will be recognized as a protein, because *CGA* is the synonym of *Chromogranin A precursor*, which is P05059 in SwissProt. This false positive protein will be falsely paired with *PTN1* and *PTN2*. The previous method is hard to filter out the pair even though the pair only appears once in the article. However, the profile-feature based method is able to solve the problem by incorporate the evidence from the whole article.

4.1 Profile Feature

Profile features are selected to represent the evidence of a physical interaction. There are 3 types of profile features:

- 168 Unigram/Bi-gram Features
  100 of these features are selected by chi-square statistics of distinctiveness [18], and the rest 68 features are selected from Molecular Interaction (MI) ontology's [30] definition of Physical Interaction and Detection Method.
- 91 Pattern Features
  These features are generated in a semi-supervised manner [7]. These features have a form as "PTN * bind to * PTN", where PTN indicates a protein entity, and * means any word that can be skipped. The pattern feature is matched against the sentences as a regular expression.
- 2 Position Features
  One is whether the two proteins co-occur within the title; the other is whether they co-occur within the abstract.

These features eventually comprise a 261-dimensional feature vector, where each dimension is 1 or 0 indicating the presence or absence of a feature. Examples of these features are shown in Table 1.

Table1: Feature examples

Unigram/Bigram	Pattern
aggregation	activation of **PTN1* *by **PTN2*
crystallography	*PTN1* bind **PTN2*
elongation	*PTN1* *interact with **PTN2*
circular dichroism	*PTN1* *form complex with **PTN2*

4.2 Feature Construction

Every protein pair occurred within a sentence is viewed as a candidate. These sentences are considered as evidence. For each pair, *profile features* are extracted from all the sentences in which the pair appears. The corresponding bit is set as 1 if the feature is found in these sentences, see Figure 2. Through such a representation with abundant features, information from the whole document has been incorporated.

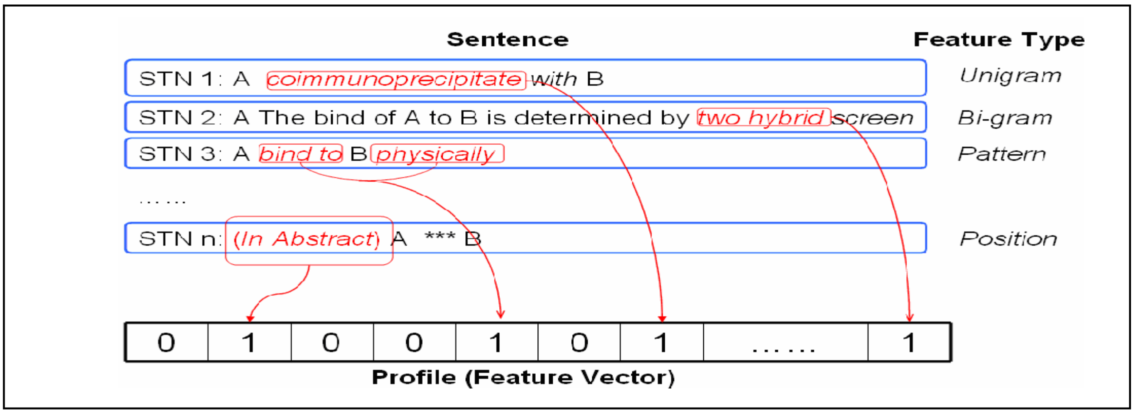

Sentence
Feature Type
STN 1: A coimmunoprecipitate with B
Unigram
STN 2: A The bind of A to B is determined by two hybrid screen
Bi-gram
STN 3: A bind to B physically
Pattern
……
STN n: (In Abstract) A *** B
Position
0
1
0
0
1
0
1
……
1
Profile (Feature Vector)

Figure 2: Feature Construction

4.3 Training

We use SVM-Light as our classifier [17]. In this part, we will discuss the construction of the training set.

The problem of the training corpus is that the supervised information is not given at the sentence level but only at the document level. The annotations from MINT and IntAct only specify the database ID (mainly SwissProt AC) of the interactors in the article, which means they do not provide the evidence texts that support the existence of the physical interaction, neither do we know where the interactors appear in the texts. So the annotation of the training corpus can not be used directly.

To establish the training set that the classifier can make use of, the protein names are first extracted and mapped to primary Access Number of SwissProt entries by our NER module. The protein pairs[1] which are annotated by domain experts are considered as positive samples. The other protein pairs in the text are treated as negative samples. Since lots of proteins are not part of a physical interaction, the number of negative samples overwhelms that of positive samples, which will lead to a biased distribution of training set. So from 740 training articles we randomly choose the negative samples twice as many as the positive samples and finally get 701 positive samples and 1402 negative samples as the training set for SVM.

5. EXPERIMENT AND EVALUATION

Data used in the experiments are introduced in Section 4.1. Evaluation methods are presented in detail in Section 4.2. The experiments of NER/N and Physical Interaction Extraction are discussed in Section 4.3 and 4.4. The evaluation results are officially published by BioCreAtIvE 2006.

5.1 Data Setup

BioCreAtIvE 2006 provided 740 full-text articles for training and 358 articles for testing from MINT and IntAct (The annotations of the testing articles are not released until the end of BioCreAtIvE 2006). These articles are manually annotated by database curators. The interaction pairs are only annotated from the full text articles in case there was an experimental confirmation for this interaction mentioned in the article.

5.2 Evaluation

Due to the annotation methods applied by MINT and IntAct, the evaluation in BioCreAtIvE 2006 is different from previous evaluation of PPI extraction tools. Traditionally, the annotation will focus on one sentence and provide the *position* of the interactors and their *relations* (such as “induce” or “bind”). Thus the evaluation requires the exact match of these criteria to mark the result as true positive [13]. However, the current annotation in MINT and IntAct is focused on document level and provide the normalized database ID of the physically interacted proteins. So, the evaluation requires the detection of normalized interaction pairs of the document.

The evaluation for NER/N provided by BioCreAtIvE 2006 is also different from that of traditional NER task, because it only considers the physically interacted protein ACs as reference. So a lot of correctly recognized and normalized proteins are evaluated as false positive because they are not annotated as part of a physical interaction. Thus, the data of the evaluation can’t represent the absolute performance of a NER/N module, but the comparison can reveal the difference of these NER/N methods.

5.3 Named Entity Recognition And Normalization (NER/N)

The performance of our NER/N module is shown in Table 2. The average results are calculated on 45 runs from 16 teams. Our performance is much better than the mean/median performance. From the comparison, it’s obvious that our contributions to NER/N are database curation and organism-based disambiguation.

The curation will improve the database entries’ accuracy and coverage, because the official names of the SwissProt entries are very long, descriptive and formal. The addition of synonyms and gene names will significantly increase the coverage. The unification of the various writing habits helps a lot to improve the matching accuracy. The F-score after database curation is improved by 77.3% compared to the naïve match.

The disambiguation based on organism information collected from the whole article greatly improves the NER/N’s precision with slight loss in recall. The F-score is improved by 14.6% after disambiguation. Thus, the disambiguation by organism is efficient.

[1] Protein Pair is defined as two proteins which co-occur in at least one sentence in the name-mapped text.

Although our method outperforms other methods (Our > Mean + Dev), the result is far from satisfaction. One problem is the wide spread synonyms which are hard to differentiate, such as PKB, Akt, and CGA. Another problem lies in the disambiguation. One protein name may refer to multiple entries in SwissProt, such as protein isoforms, which make the disambiguation method hard to handle.

Table 2: Overall performance vs. our overall results of NER/N

Score	Proteins normalized to SwissProt entries			
	Precision	Recall	F-score	
Mean	0.1495	0.2828	0.1707	
Std. Dev	0.0963	0.1294	0.0764	
Median	0.1337	0.2723	0.1683	Improv.
Naïve Match	0.2223	0.1024	0.1402	N/A
Prev. +Curation	0.2345	0.2648	0.2487	**+77.3%**
Prev. +Disambiguation	**0.3483**	**0.2410**	**0.2849**	**+14.6%**

5.4 Physical Interaction Extraction

To illustrate the effectiveness of profile-feature based method, we compare our methods with other methods submitted by other 45 runs from 15 teams in BioCreAtIvE 2006. Moreover, we adopt the results of pattern based method derived from ONBIRES [13, 7] as the baseline. The pattern based method learns lexicon-syntactic patterns describing interactions in a semi-supervised way: it first learns the patterns from large amount of unlabeled texts and then uses relatively small amount of labeled texts to select the candidate patterns. After that, the patterns are aligned against the sentences to extract interactions, where the matching score must exceed a pre-specified threshold. In this model, interactions are extracted at the sentence level. Thus, the approach is sensitive to the performance of NER which is far from satisfactory.

Table 3 shows the overall performance for both average results of all runs and our submitted results (two results by pattern based method, ONBIRE, and one result by profile-feature based method). It is worth noting that our results are much better than mean performance across all runs from all teams. And our system based on profile-feature excels others significantly (Our > Mean + 2*Dev) and is ranked top in the evaluation.

One reason for the whole system achieving higher performance is our effective NER/N module. To illustrate the contribution of profile-feature based method alone, we compare it with our pattern based method.

Profile-feature based model achieves the best results compared to the other two runs submitted by pattern based system, ONBIRES. These three results are achieved by the same NER/N module, so the NER/N does not impact the comparison of different extraction methods. It is obvious that the profile-feature based model contributes a much better precision than others. This is mainly because the model is more rational by synthesizing the evidence from the whole article, thus causes less false positive results.

So, the conclusion can be made from the evaluation that the profile-feature based method outperforms the traditional extraction methods, such as the pattern based method. The main advantage is that profile-feature is able to encode various features from the whole article. Because the task is focused on physical interactions, extraction methods which only exploit single evidence is prone to generating false positive results, while profile based method can incorporate lots of evidence and extract the semantic relations more rational.

6. DISCUSSION

To extract physically interacted protein pairs from the full-text articles has two major challenges: 1) recognizing protein named entities and mapping each entity to a unique entry in the SwissProt database; 2) identifying protein pairs which have been experimentally confirmed to have physical interactions. These challenges can lead to *Biologically Meaningful Knowledge*, which requires deeper understanding of semantic relations in the text.

First, NER/N is a most challenging task, and is obvious the bottleneck of the system. The difficulty to recognize and normalize the names to SwissProt entries is due to various synonyms and ambiguity in names. Database curation and organism based disambiguation are exploited as solutions. However, since the conventional naming of biomedical entities is far from standardized, the curation procedure lacks unified guides and fails to help the database to cover all the terms. Moreover, the normalization of the protein names to the unique entries in SwissProt database requires deeper understanding of the semantics buried in natural language. Future work will be focused on exploiting semantic information of the article for NER/N. The third problem is that the processing speed is not suitable for real-time application. We will try to speed up the NER/N process in the future by 1) indexing the protein terms in SwissProt and 2) dictionary matching by suffix tree.

Second, the profile based method is superior to previous ones because it incorporates evidence all over the article. However, one problem is that the model considers the article as a linear structure and misses a lot of useful information such as the positioning feature. The future work will focus on using more information from different regions of the full texts, such as the table/figure captions and cross-reference information to extract the interactions. Another problem is the lack of understanding of the syntactic structure and semantics of the sentence. This is a common problem because of the immature of Natural Language Understanding. We will try to develop novel method to capture the deeper semantics of the document by NLP techniques, such as the semantic lexicon/role defined in FramNet [2].

We believe that the text mining in biomedical area is to extract and manage the biological meaningful knowledge from the literatures. This knowledge can be used to integrate with the high-throughput experimental data for validation, hypothesis generation and biological discovery, and finally make the text mining really helpful to biologists.

Table 3: Physical interaction extraction performance averaged on 45 runs from 16 teams vs. our overall results. "Whole collection" means all the articles have been considered. "SwissProt only article collection" means articles containing exclusively interaction pairs which can be normalized to SwissProt entries have been scored.

Score	Whole collection			SwissProt only article collection		
	Precision	Recall	F-score	Precision	Recall	F-score
Mean	0.1062	0.1858	0.1035	0.1160	0.2000	0.1127
Std. Dev	0.0945	0.1001	0.0761	0.1035	0.1062	0.0836
Median	0.0755	0.1961	0.0788	0.0808	0.2156	0.0842
ONBIRES (th=0.0)	0.1373	0.2905	0.1579	0.1566	0.3189	0.1784
ONBIRES (th=80.0)	0.2177	0.2651	0.2039	0.2434	0.2828	0.2247
Profile-feature	**0.3096**	**0.2935**	**0.2623**	**0.3695**	**0.3268**	**0.3042**
Rank (in 45 runs)	**2**	**4**	**2**	**2**	**3**	**1**

7. ACKNOWLEDGEMENT

The work is supported by Chinese Natural Science Foundation under grant No. 60572084 and 60621062, National High Technology Research and Development Program of China (863 Program) under No. 2006AA02Z321, as well as Tsinghua Basic Research Foundation under grant No. 052220205.

8. REFERENCE

[1] Bader, G.D., Donaldson, I., Wolting, C., Quellette, B.F., Pawson, T. and Hogue, C.W. BIND –The Biomolecular Interaction Network Database. *Nucleic Acids Research*, 29(1), 2001, pp. 242–245.

[2] Baker, C., Fillmore, C., and Lowe, J. The Berkeley FrameNet project. In *Proceedings of the Thirty-Sixth Annual Meeting of the Association for Computational Linguistics and Seventeenth International Conference on Computational Linguistics*, pp. 86-90, 1998

[3] Chiang, J.H., and Yu, C.Y. Literature extraction of protein functions using sentence pattern mining. *IEEE. Trans. On Knowledge and Data Enginerring*, 17 (8), 2005, pp. 1088-1098.

[4] Craven, M. Learning to extract relations from Medline. *AAAI-99 Workshop on Machine Learning for Information Extraction*, 1999.

[5] Daraselia, N., Yuryev, A., Egorov, S., Novichkova, S., Nikitin, A., and Mazo, I. Extracting Human Protein Interactions from MEDLINE Using a Full-Sentence Parser. *Bioinformatics*, vol. 20(5), 2004, pp. 604-611.

[6] Ding, J., Berleant, D., Nettleton, D., and Wurtele, E. Mining medline: abstracts, sentences, or phrases? In *Proceedings of the 7th Pacific Symoisium of Bio-computing*, pp. 326–337, 2002

[7] Ding, S.L., Huang, M.L., and Zhu, X.Y. Semi-supervised Pattern Learning for Extracting Relations from Bioscience Texts. In *Proceedings of the 5th Asia-Pacific Bioinformatics Conference*, pp. 307-316, 2007.

[8] Fukuda, K., Tsunoda, T., Tamura, A., and Takagi, T. Toward information extraction: identifying protein names from biological papers. In *Proceedings of the 3rd Pacific Symposium on Biocomputing*, pp. 707-718, 1998.

[9] Fundel K., Küffner R., and Zimmer R. RelEx - Relation extraction using dependency parse trees. *Bioinformatics*, vol. 23, 2007, pp. 365-371

[10] Hanisch, D., Fundel, K., Mevissen, H.T., Zimmer, R., and Fluck, J. ProMiner: Rule-based Protein and Gene Entity Recognition. *BMC Bioinformatics* 2005, 6 (Suppl 1): S14.

[11] Hermjakob, H., Montecchi-Palazzi, L., Lewington, C., Mudali, S., Kerrien, S., Orchard, S., Vingron, M., Roechert, B., Roepstorff, P., Valencia, A., Margalit, H., Armstrong, J., Bairoch, A., Cesareni, G., Sherman, D., and Apweiler, R. IntAct: an open source molecular interaction database. *Nucleic Acids Research*, vol. 32, 2004, pp. D452-D455.

[12] Hirschman, L., Yeh1, A., Blaschke, Y., and Valencia, A. Overview of BioCreAtIvE: critical assessment of information extraction for biology. *BMC Bioinformatics*, 2005, 6 (Suppl): S1

[13] Huang, M.L., Zhu, X.Y., Ding, S.L., Yu, H., and Li, M. ONBIRES: ONtology-based BIological Relation Extraction System. In *Proceedings of the Fourth Asia Pacific Bioinformatics Conference*, pp. 327-336, 2006.

[14] Huang, M.L., Zhu, X.Y., Hao, Y., Payan, D.G., Qu, K., and Li, M. Discovering patterns to extract protein-protein interactions from full texts. *Bioinformatics*, vol. 20, 2004, pp. 3604-3612.

[15] Jelier,R., Jenster, G., Dorssers, L.C., van der Eijk, C.C., van Mulligen, E.M., Mons, B., Kors, J.A. Co-occurrence based meta-analysis of scientific texts: retrieving biological relationships between genes. *Bioinformatics*, vol. 21, 2005, pp. 2049–2058.

[16] Jensen, L.J., Saric, J., and Bork, P. Literature mining for the biologist: from information retrieval to biological discovery. *Nature Reviews Genetics*, Vol. 7(2), 2006, pp. 119-129.

[17] Joachims, T. Making large-Scale SVM Learning Practical. *Advances in Kernel Methods* - Support Vector Learning, B. Schölkopf and C. Burges and A. Smola (ed.), MIT-Press, 1999.

[18] Manning, C., and Schütze, H. *Foundations of statistical natural language processing*. Cambridge, MA: MIT Press. 1999.

[19] Marcotte, E.M., Xenarios, I., and Eisenberg, D. Mining literature for protein-protein interactions. *Bioinformatics*, vol. 17, pp. 259-363, 2001

[20] Natarajan, J., Berrar, D., Hack, C., and Dubitzky, W. Knowledge Discovery in Biology Texts: Applications, Evaluation Strategies, and Perspectives. *Critical Reviews in Biotechnology*, vol. 25(1-2), 2005, pp. 31-52.

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

[22] Pustejovsky, J., Castano, J., and Zhang, J. Robust relational parsing over biomedical literature: extracting inhibit relations. In *Proceedings of the seventh Pacific Symposium on Bio-computing*, pp 362-373, 2002.

[23] Rindflesch, T., Hunter, L., and Aronson, L. Mining molecular binding terminology from biomedical text. In *Proceedings of the AMIA Symposium*, Washington, D.C., pp. 127–131, 1999.

[24] Schuemie, M.J., Weeber, M., Schijvenaars, B.J., van Mulligen, E.M., van der Eijk C.C., Jelier, R., Mons, B., and Kors, J.A. Distribution of information in biomedical abstracts and full-text publications. *Bioinformatics*, vol. 20(16), 2004, pp. 2597-2604.

[25] Settles, B. Biomedical Named Entity Recognition Using Conditional Random Fields and Rich Feature Sets. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications* (*NLPBA*), Geneva, Switzerland, pp. 104-107, 2004.

[26] Yoshimasa, T., and Tsujii, J. Boosting Precision and Recall of Dictionary-Based Protein Name Recognition. In *Proceedings of the ACL 2003 workshop on Natural language processing in biomedicine,* pp. 41-48, 2003.

[27] Yu, H., Hatzvisaailoulou, V., Friedman, C., Rzhetsky, A., and Wilbur, W.J. Automatic Extraction of Gene and Protein Synonyms from Medline and Journal Articles. In *Proceedings of AMIA Symposium*, pp. 919-923, 2003

[28] Zanzoni, A., Montecchi-Palazzi, L., Quondam, M., Ausiello, G., Helmer-Citterich, M., Cesareni, G. (2002) MINT: a Molecular INTeraction database. *FEBS Letters*, vol. 513(1), 2002, pp. 135-140.

[29] Zhou, G.D., Zhang, J., Su, J., Shen, S., and Tan, C.L. Recognizing names in biomedical texts: a machine learning approach. *Bioinformatics*, Vol. 20 (7), 2004, pp. 1178-1190.

[30] http://www.ebi.ac.uk/ontology-lookup/browse.do?ontName=MI

[31] http://www.ncbi.nlm.nih.gov/entrez/query.fcgi?db=Taxonomy

A Decomposition Approach for Discovering Network Building Blocks[*]

Qiaofeng Yang
Physical Biosciences Division
Lawrence Berkeley National Laboratory
Berkeley, CA 94720, USA
qyang@lbl.gov

Stefano Lonardi
Department of Computer Science & Engineering
University of California
Riverside, CA 92521, USA
stelo@cs.ucr.edu

ABSTRACT

The increasing availability of biological networks (protein-protein interaction graphs, metabolic and transcriptional networks, etc.) is offering new opportunities to analyze their topological properties and possibly gain new insights in their design principles. Here we concentrate on the problem of *de novo* identification of the building modules of networks, which we refer to as *network modules*.

We propose a novel graph decomposition algorithm based on the notion of edge betweenness that discovers network modules without assuming any *a priori* knowledge. We claim that the knowledge of the distribution of network modules carries more information than the distribution of subgraphs which is commonly-used in the literature. To demonstrate the effectiveness of the statistics based on network modules, we show that our method is capable of clustering more accurately networks known to have distinct topologies, and that the number of informative components in our feature vector is significantly higher. We also show that our approach is very robust to structural perturbations (i.e., edge rewiring) to the network. When we apply our algorithm to protein-protein interaction (PPI) networks, our decomposition method identifies highly connected network modules that occur significantly more frequently than those found in the corresponding random networks. Detailed inspection of the functions of the over-represented network modules in *S. cerevisiae* PPI network shows that the proteins involved in the modules either belong to the same cellular complex or share biological functions with high similarity. A comparative analysis of PPI networks against AS-level Internet graphs shows that in AS-level networks highly connected network modules are less frequent but more tightly connected with each other.

Categories and Subject Descriptors

J.3 [**Life and Medical Sciences**]: Biology and Genetics; D.2.8 [**Software Engineering**]: Design—*Methodologies*

General Terms

Graph theory

1. INTRODUCTION

Many real world systems can be modeled as network graphs, and their formal analysis can help us understand the underlying design principles behind each corresponding system. For example, identifying highly connected subgraphs in protein-protein interaction graphs can potentially enable life scientists to discover new protein complexes or speculate about the functions of unknown proteins [3, 23, 6]. In addition, the topological analysis can offer new insights in the roles of structural elements on the network performance, such as, traffic flow or diffusion of computer viruses over the Internet, epidemic diseases or ideas spreading in social networks, error and attack tolerance of various communication networks, etc.

In the past few years, a significant research activity has been focused on studying global and local properties of the network graphs (see, e.g., [7, 4, 27]) and significant breakthroughs have been achieved. For instance, the concept of scale-free networks, and the small world phenomenon have changed the way we model and analyze graphs across many different disciplines, from biological networks, to social networks all the way to communication networks.

In an attempt to understand the design principles of networks, the concept of *network motif* [18] has been recently proposed to represent the subgraphs in the network that occur significantly more often than the number of times they occur in the corresponding random networks. By using the concept of network motif, the authors of [18] were able to show that similar motifs were found in several information processing networks irrespective of their origin. They argued that these motifs may define universal classes of networks. The concept of network motif has been widely adopted to study local properties of various biological networks. For example, the network motifs in the transcriptional regulation network of *E. coli* were studied by Shen-Orr *et al.* [24]. The authors found that three highly significant motifs, namely, the *feed-forward loop*, the *single input module* and the *dense overlapping regulons*, are the main building blocks of the network. They also discovered that each motif is associated with a specific function in determining gene expression. A large collection of metabolic pathway networks were analyzed by Koyuturk *et al.* in [13]. The authors designed

[*]This project was supported in part by NSF CAREER IIS-0447773, and NSF DBI-0321756.

BIOKDD '07 San Jose, California USA
Copyright 2007 ACM 0-12345-67-8/90/01 ...$5.00.

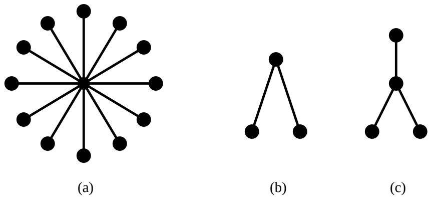


Figure 1: Illustrating the bias introduced by the occurrences of *hubs* (a) on the counts of subgraphs (b) and (c)

an efficient algorithm based on the *frequent itemsets* algorithm [1, 10] to find frequent subgraphs in the metabolic networks of over 150 organisms. Wuchty *et al.* [28] studied the conservation of 678 yeast proteins with the corresponding ortholog proteins in five higher eukaryotic organisms. The authors discovered that the orthologs are not randomly distributed in the yeast protein interaction network but are the building blocks of larger cohesive motifs, which tend to be evolutionarily conserved. They also observed that larger motifs tend to be conserved as a whole, with each of their components having an ortholog. Yeger-Lotem *et al.* [31] proposed the concept of *composite network motifs*, which consist of patterns from both transcription-regulation and protein-protein interaction networks that appear significantly more often than in random networks. They detected two-protein, three-protein, and four-protein motifs that occur in both networks.

Recently, the concept of network motif has been used to classify graphs. Milo *et al.* [17] introduced the concept of *significance profile* which is computed over the small subgraphs of the network and is used to cluster different networks. The profile is a normalized z-score for each subgraph obtained by comparing the number of occurrences of the subgraph to the number of occurrences in corresponding random networks. The authors were able to show that all networks having similar functionality share similar profiles. Surprisingly a few super-families of unrelated networks also share very similar significance profiles. Along the same line, Middendorf *et al.* [16] proposed a discriminative approach to understand the design of complex networks. The authors built a classifier based on alternating decision tree and trained the classifier using raw subgraph counts of 148 subgraphs obtained from seven random graph models. The protein-protein interaction graph (PPI) of *D. melanogaster* was classified as duplication-mutation-complementation network [26].

While this paper was under review, a work by Luo *et al.* [14] appeared in the scientific literature. The authors present an agglomerative algorithm to identify biological modules in PPI based on the concept of betweenness and modularity [9, 19, 21].

```
Input: Graph $G$, integer $k$ and a list $L$ of all subgraphs $g_i$
of size smaller or equal to $k$
Output: Number of occurrences of each subgraph $g_i$ in $L$
    $C \leftarrow$ CONNECTED_COMPONENTS($G$)
    for each connected component $G_d \in C$ do
        ENQUEUE($Q, G_d$)
    while $Q \neq \emptyset$ do
        $n, G_c \leftarrow$ 1,DEQUEUE($Q$)
        if NUM_VERTICES($G_c$) $\leq k$ do
                UPDATE_COUNTS($L, G_c$)
        else
            while $n = 1$ do
                $e \leftarrow$ EDGE_BETWEENNESS($G_c$)
                REMOVE_EDGE($G_c, e$)
                $C \leftarrow$ CONNECTED_COMPONENTS($G_c$)
                $n \leftarrow$ SIZE($C$)
            for each connected component $G_d \in C$ do
                ENQUEUE($Q, G_d$)
    return $L$
```

Figure 2: Sketch of the edge betweenness decomposition algorithm

We observe that the majority of the approaches mentioned above share two common features, namely (1) they are designed to operate on directed graphs and (2) they are based on the *exhaustive* enumeration of all the subgraphs (up to a given size) in the network. From here on, we refer to exhaustive subgraph enumeration approaches as *Subgraph Counting Network Motif (SCNM)* approaches. We observed that using the raw subgraph counts as an indicator of over-representation has an inherent shortcoming. This arise from the fact that some subgraphs substantially overlap with each other, which in turn creates strong biases in the absolute counts. For example, *hubs* (nodes with high degree) are quite common in PPI networks [11]. As illustrated in the example of Figure 1, if one hub of degree twelve (a) is present in the network, then we will observe 66 subgraphs of type (b) and 220 subgraphs of type (c). If the network under study has several hubs, then type (b) and type (c) subgraphs will be highly over-represented when compared to random networks and they will dominate the analysis. However, such subgraphs may well be totally irrelevant from a statistical or biological viewpoint.

Here we address this limitation of SCNM approaches by introducing a novel graph decomposition method based on the concept of edge betweenness [9, 19, 21]. Our method decomposes the network into a collection of small subgraphs (called *network modules*), and thereby creates a disjoint partitioning of the nodes. The fact that a node can belong to only one network module solves the problem of counting overlapping subgraphs, and potentially allows us to assign putative biological functions to the nodes involved in the same network module. In order to evaluate objectively the effectiveness of our method to extract important features from the graph, we compare it to SCNM approaches on the problem of graph classification (along the lines of [17]). Results show that our approach is more accurate in distinguishing networks known to have distinct topologies. Our method is also tested for robustness against random perturbations to the network (i.e., edge rewiring), and our findings suggest low sensitivity to small changes in the graph. Finally, we report on preliminary results on the analysis of several protein-protein interaction networks (PPI). We show that highly connected network modules are more over-represented in PPI networks than those found in their random counterparts, and that the proteins involved either belong to the same cellular complex or share highly similar functions.

2. AN EDGE BETWEENNESS DECOMPOSITION ALGORITHM

It is well-known that proteins that are involved in the

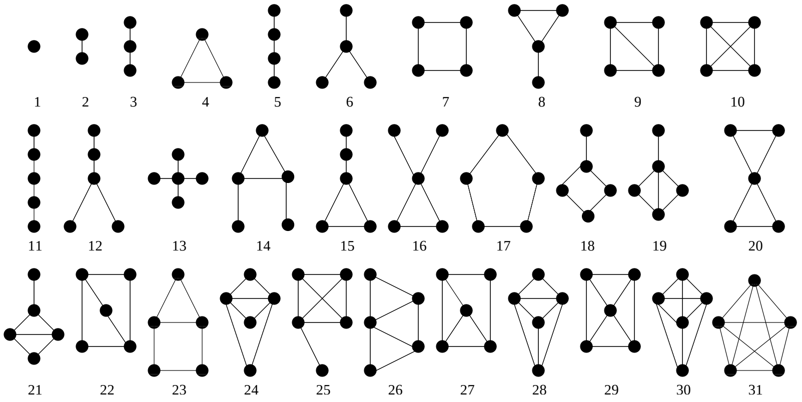


Figure 3: Non-isomorphic subgraphs of size ranging from one to five nodes

same cellular process or reside in the same protein complex are expected to have strong interactions with their partners. At the same time, interactions between distinct functional modules are expected to be suppressed in order to increase the overall robustness of the network by localizing effects of deleterious perturbations [15]. Biological networks are believed to consist of different modules with distinct functions [11, 22]. Here we are interested in identifying the building blocks of these functional modules without any *a priori* biological knowledge.

In this study, the detection of the building modules is based solely on the concept of edge betweenness. Consider the shortest paths between all pairs of vertices in a graph. The *betweenness* of an edge [9] is defined as the number of these shortest paths running through it[1]. When two different functional modules are loosely connected with each other, all shortest paths between vertices in those two modules have to traverse the few links between them. By removing those edges, the functional modules are separated from one another. The effectiveness of the betweenness approach on PPI graph in decomposing the network to find functional modules has been recently reported in [6]. In order to find the basic building modules of the network, we proceed as follows. First, we compute the edge betweenness of all the edges. Then, we start removing the edges with the highest betweenness until the largest connected component of the graph becomes smaller than or equal to some predefined threshold (k). Each time we remove an edge, the betweenness is recomputed from scratch. All the "small" connected components are then classified and counted. We refer to all the classified small subgraphs as *network modules*.

The outline of the algorithm is sketched in Figure 2. The function EDGE_BETWEENNESS computes and returns the edge with the largest edge betweenness. Evaluating the betweenness value for all edges of graph $G = (V, E)$ requires $O(|V||E|)$ time, by running a BFS from each node of the graph. The iterative removal of all $|E|$ edges leads an overall worst-case time complexity of $O(|V||E|^2)$ for our approach. Because of its computational cost, a distributed implementation of EDGE_BETWEENNESS was used [30].

When comparing our approach to Newman and Girvan method [19, 21], several major differences emerge. Although both algorithms employ betweenness to determine the order in which edges have to be removed, Newman and Girvan's relies on a metric that evaluate the quality of the decomposition, called *modularity*. In their method, the final decomposition is obtained by "cutting" the dendrogram of the decomposition at the point in which the value of the modularity peaks. In our method, we keep removing edges until the graph disconnects; only if the component is small enough, we stop the process and classify the module in one of 31 non-isomorphic subgraphs (shown in Figure 3).

Table 1: The set of graphs used in the experiments

ID	*name*	$\vert V\vert$	$\vert E\vert$
1	*H. pylori* PPI	702	1359
2	*H. sapiens* PPI	1059	1318
3	*C. elegans* PPI	2629	3970
4	*S. cerevisiae* PPI	4770	15181
5	*D. melanogaster* PPI	7057	20815
6	*E.coli.* Transcription	418	519
7	*S. cerevisiae* Transcription	688	1078
8	*C. elegans* Neuron Connectivity	202	1952
9	AS1	3522	6324
10	AS2	4885	9276
11	AS3	7246	14629
12	AS4	10515	21455
13	AS5	4686	8772
14	AS6	9200	28957
15	Circuits1	122	189
16	Circuits2	252	399
17	Circuits3	512	819
18	Protein Structure1	95	213
19	Protein Structure2	53	123
20	Protein Structure3	97	212
21	Social1	67	142
22	Social2	32	80
23	Japanese	2704	7998
24	English	7381	44207
25	French	8325	23841
26	Spanish	11586	43065

Note that in our approach each vertex can only belong to one network module, in contrast to the *network motifs* widely used in the literature [18, 24, 28, 31, 16], which are based on exhaustive subgraph counting (SCNM) approach. To make a distinction between our approach and SCNM approach, we refer to our method as *Graph Decomposition Network Module* (GDNM) approach.

[1] If multiple shortest paths between a pair of nodes exists, each shortest path contributes an equal fraction to the edge-betweenness of their edges [5].

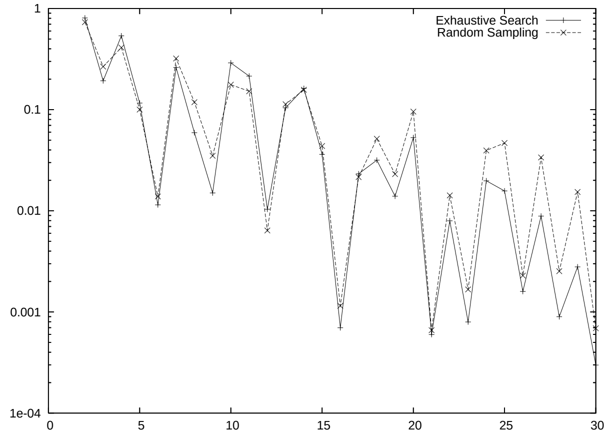


Figure 4: Comparing the exhaustive subgraph enumeration and random sampling on the graph Protein Structure3. The x-axis represents the subgraph index (according to Figure 3), whereas the y-axis represents the subgraph concentration. Subgraph of size 3, 4 and 5 were sampled 100,000 times

3. REPRESENTATION OF GRAPH FEATURES

Since the number of possible subgraphs grows exponentially with the number of nodes, in this study we only consider the number of occurrences of network modules of size up to five nodes (as in papers [20, 28]). As illustrated in Figure 3, there are 31 non-isomorphic subgraphs of size up to $k = 5$. Each subgraph g_i is indexed by an integer $i = 1, \ldots, 31$.

When a graph G is processed by the algorithm in Figure 2 where $k = 5$ and $L = \{g_1, \ldots, g_{31}\}$, a feature vector of 31 components is returned. Note that the number of occurrences of subgraphs of size one and two in the SCNM approaches it is somewhat meaningless, since they correspond respectively to the number of nodes and the number of edges in the graph. As a consequence, the feature vector for the exhaustive subgraph counting is 29-dimensional for $k = 5$. In our approach it is meaningful to keep track of all those 31 counts because when the network is broken down into connected components, some of those components may just have one or two nodes.

Before we can use these feature vectors to classify graphs, we need to normalize the components to remove the dependency on the absolute size of the graph. This will allow us to compare graphs of different sizes. We consider two normalizations, as explained below.

3.1 Subgraph Proportion Normalization

The first normalization tries to capture what proportion of nodes belongs to each subgraph class g_i. Given a graph $G = (V, E)$ and the vector $[n_i]$ of network module counts, the i-th component of the *subgraph proportion* vector is defined as $n_i|g_i|/|V|$ where n_i is the number of occurrences for subgraph class g_i. In the following we will use this normalization for the feature vectors associated with network building modules computed by our GDNM decomposition. Note, that since $\sum_{i=1}^{31} n_i|g_i| = |V|$, the sum of all the components of the subgraph proportion vector is always 1.

3.2 Subgraph Concentration Normalization

The second (alternative) normalization denotes how frequent is one subgraph class with respect to all the other classes with the same number of nodes. Given the vector $[n_i]$ of subgraph counts, the i-th component of the *subgraph concentration* [12] vector is defined as $n_i / \sum_{j:|g_j|=|g_i|} n_j$, where n_i is the number of occurrences of subgraph g_i. It is easy to realize that the sum of all the components of the subgraph concentration vector is always k. In the following we will use this normalization for the vector associated with the exhaustive SCNM approaches, since the subgraph proportion vector is not feasible for it. If we used the subgraph concentration normalization for the GDNM approach, we would loose the information carried by the network modules of size one and two (both components will be one).

4. RESULTS AND DISCUSSION

To test the effectiveness of our GDNM approach, we conducted several experiments and compared the results with the SCNM method. The first set of experiments is about graph classification, both on simulated data and on real networks (see Table 1 for a summary of the dataset). Five PPI networks were obtained from DIP database [29] and the rest of the networks are from [2]. We also performed a robustness test of our technique and computed the over-represented modules in PPI networks. Then, we studied the biological functions associated with the over-represented network modules found by our algorithm on the yeast PPI network.

4.1 Graph classification

4.1.1 *Estimating the subgraph counts*

Due to the large size of some of the networks in our dataset, the exhaustive subgraph enumeration is not always possible. In order to obtain the network motifs based on subgraph counting, we adopted the sampling algorithm by Kashtan *et al.* [12] to compute the number of occurrences of each subgraph in the network. For completeness of presentation, we briefly review the sampling procedure for a subgraph of size k. (1) Pick an edge $e = (u, v) \in E$ uniformly at random; (2) Set $U = \{u, v\}$ (3) Compute the set F of vertices that are adjacent to the vertices in U; (4) Pick one vertex from F at random and add it to U; (5) Repeat steps (3) and (4), until the target number k of vertices is reached.

Figure 4 shows a comparison between the exhaustive subgraph enumeration and the sampling approach for the "Protein Structure 3" network. The figure shows that the sampling algorithm gives good approximations of the subgraph concentration. We compared the sampling approach to the exhaustive count on many other relatively small graphs and in all cases it was capable of producing good estimates.

4.1.2 *Classification of Real Networks*

The real-world networks summarized in Table 1 were processed along the same lines as the previous experiment. It is worth noting that we treated all networks as undirected graphs although some of them (i.e., transcription regulation networks, social networks and language networks) are directed. Figure 5 shows the two Pearson correlation coefficient matrices for the 26 networks for our decomposition algorithm (left) and the subgraph counting approach (right).

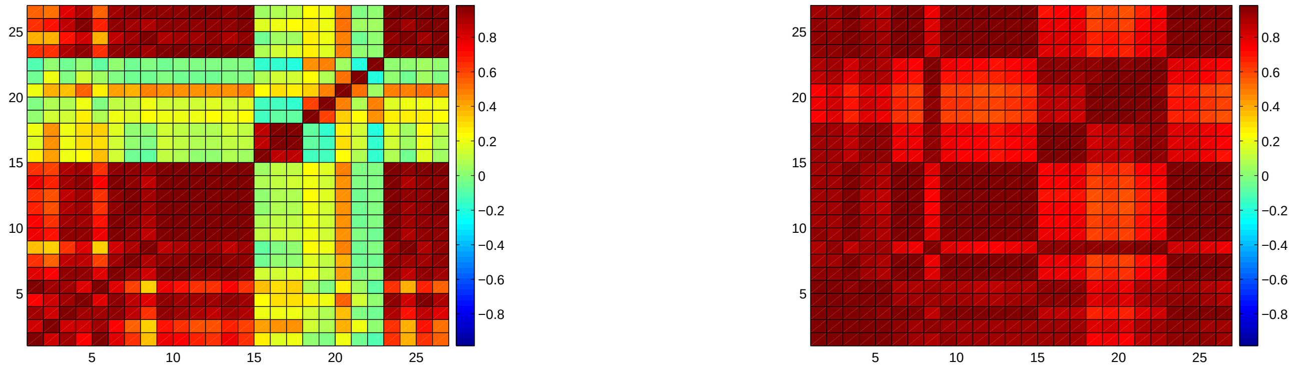


Figure 5: Pearson correlation coefficient matrix on the 26 real networks in Table 1 using decomposition network module approach (LEFT) and subgraph counting network motif approach (RIGHT)

Both pictures use the same scale. An inspection of the right matrix (corresponding to SCNM) shows that almost all networks are significantly correlated with one another. On the other hand, the feature vectors computed with our approach (left) show clearly that there are several distinct families of networks. The first is a big cluster composed of biological networks (PPI, transcriptional and neural), Internet AS-level networks, and languages networks, although the neural network does not share significant similarity with some members of this family. The second consists of circuit networks and the third consists of protein structure networks. The two social networks are not strongly correlated probably due to their small size. Note that circuit, protein structure and social networks are clustered together in the SCNM correlation matrix (right).

4.1.3 *Principal component analysis*

In order to establish an objective measure of the quality of the features extracted by the two approaches, we performed a principal component analysis (PCA) of the covariance matrices for both methods and both datasets (random and real data). The goal of this PCA analysis is to establish the effective dimensionality of the feature vectors obtained by the two methods. Figure 6 shows the distribution of the eigenvalues of the covariance matrix for random (left) and real networks (right). The value of the eigenvalues clearly illustrates that our decomposition method extracts more information from the graph. The analysis shows that our approach has a larger number of significant independent components in the feature vectors. For example on the random dataset, 11 principal components have significant eigenvalues whereas only three are obtained using the subgraph counting approach. On the real network dataset, our method extracts 21 significant components against 14 of the other approach. The fact that we have more “useful” components in our feature vectors can explain why our approach creates sharper and more accurate boundaries between different types of graphs.

4.2 Robustness

To test the sensitivity of the GDNM approach to random perturbation to the graph, we conducted a few experiments in which we swapped some of the edges of the network at random. This process is called *rewiring* [4], and works as follow.

Given a graph $G(V,E)$, randomly pick two edges $(u,v) \in E$ and $(x,y) \in E$. If $(u,x) \notin E$ and $(v,y) \notin E$, add (u,x) and (v,y) to E and delete (u,v) and (x,y) from E. Otherwise, if $(u,y) \notin E$ and $(v,x) \notin E$, add (u,y) and (v,x) to E and delete (u,v) an d(x,y) from E. If both choices are feasible, then whether we should connect (u,x) and (v,y) or (u,y) and (v,x) is arbitrarily chosen at random.

Figure 7 shows the profile of the vectors computed by our decomposition method before and after random perturbations up to 10% edge-rewiring on the PPI networks of yeast and fly. The figures indicate that our approach is quite robust to random perturbations.

4.3 Enrichment of Network Modules in PPI

We applied our GDNM algorithm to two large biological networks, namely, the protein-protein interaction (PPI) network for *S. cerevisiae* (yeast) and the PPI for *D. melanogaster* (fly). According to [8] the PPI of drosophila was obtained by high-throughput yeast two hybrid assays, whereas the source of the PPI data for yeast is a mix of mass spectrometry and yeast two hybrid assays. Our objective on PPIs is to identify network modules which are over-represented when they are compared to corresponding random networks, and possibly determine whether these over-represented modules are associated with important biological functions. We studied over-represented network modules both analytically and empirically. We performed an analytical analysis based on ER random graph model and an empirical analysis based on scale-free network model. We also report a preliminary comparative analysis of PPI and AS-level networks.

Consider an Erdos-Renyi (ER) random graph $G(V,E)$, which has $|V| = n$ labeled vertices and each pair of vertices is connected with probability p. Given G we want to calculate the expected number of occurrences of subgraphs $H_{r,l}$ with r vertices and l edges. Let $Z_{r,l}$ be the random variable associated with the number of subgraphs $H_{r,l}$ in G. The expected number of occurrences of $H_{r,l}$ can be obtained as follows

$$E(Z_{r,l}) = \binom{n}{r}\binom{r(r-1)/2}{l} p^l (1-p)^{(r(r-1)/2)-l}.$$

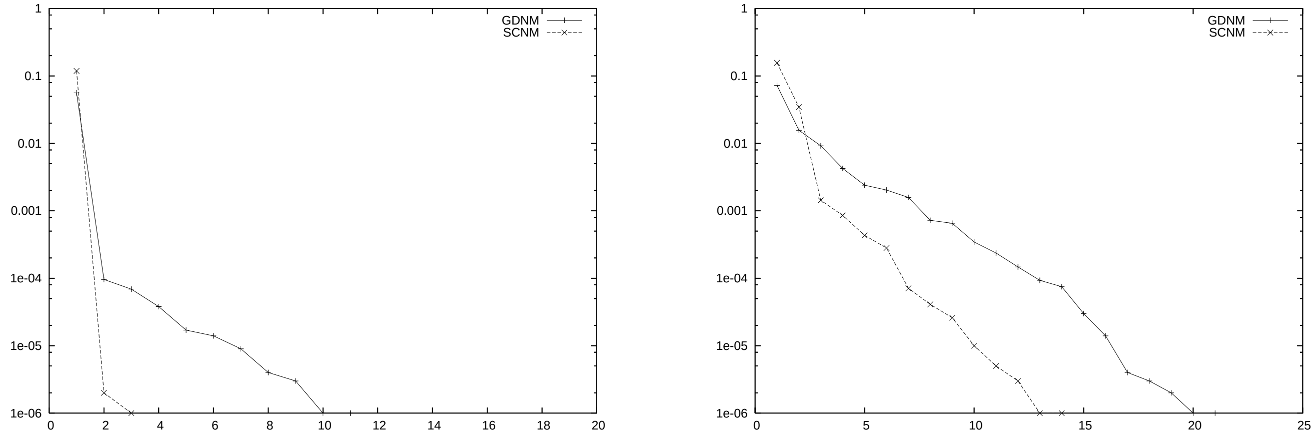


Figure 6: The eigenvalue distribution of the covariance matrix for 20 random networks (LEFT) and 26 real networks (RIGHT). The x-axis represents the ranks of the eigenvalues, the y-axis represent the absolute value of the eigenvalues

Indeed, there are $\binom{n}{r}$ ways of selecting r vertices from n vertices, and the maximum number of edges over r vertices is $\binom{r}{2} = r(r-1)/2$. The probability of observing l edges given r vertices is therefore $\binom{r(r-1)/2}{l}p^l(1-p)^{(r(r-1)/2)-l}$. The value of $E(Z_{r,l})$ is not a tight reference point when used to evaluate the significance of the subgraph counts obtained using our GDNM approach. The reason is that the count captured by $Z_{r,l}$ include overlapping and disconnected subgraphs, whereas our approach only considers non-overlapping and connected subgraphs.

Table 2 lists the observed and expected number of subgraphs $H_{r,l}$ in the yeast PPI network. It is obvious from Table 2 that densely connected subgraphs, such as $g_{28} - g_{31}$, are significantly over-represented when compared with the ER random graph model.

When comparing network module counts with the expected number of subgraphs in the ER random model, another fact need to be taken into account. Since our method removes edges with high betweenness first, it tends to favor highly connected subgraphs to sparser subgraphs. This observation has to be taken into account in the assessment of the statistical significance of these findings. In order to eliminate this bias, we also conducted an empirical analysis of the statistical significance, as described next.

To better understand the distribution of the number of subgraphs when the underlying random graph model has the same degree distribution as the original network, we performed an empirical study based on scale-free network model. The random networks were generated using the same method used to generate the scale-free networks above, but this time the degree distributions are that of the yeast and fly PPI networks. We made sure that the degree distributions are well preserved between real and random networks (statistics not shown). Our GDNM approach was subsequently applied on the scale-free random networks.

Figure 8 shows the profile of the subgraph proportion vectors for yeast (left) and fly (right) networks compared to the subgraph proportion vectors obtained from the random networks with the same degree distribution (averaged over 10 random networks). The comparison shows that large highly-connected subgraphs (i.e., those with high subgraph indices) occur significantly more often in PPI networks than in random networks. This indicates that the occurrences of densely connected modules in PPI networks cannot be explained by chance and may imply important biological roles in the cell. When interpreting these results, we should not forget how the PPI data is collected. For example, since co-immunoprecipitation detects multi-protein complexes, this in turn can possibly bias the number of occurrences of cliques or other highly connected modules. An open question is how to correct for this bias, since the technology used in the collection of protein interaction data is likely to stay with us, at least in the short term.

Table 2: The observed and expected number of subgraphs with r vertices and l edges.

Network module	r	l	Observed	Expected
g_3	3	2	214	97629
g_4	3	3	8	45
$g_5 - g_6$	4	3	118	1.05 e6
$g_7 - g_8$	4	4	20	1085
g_9	4	5	6	0.60
g_{10}	4	6	3	1.38 e−4
$g_{11} - g_{13}$	5	4	137	1.41 e7
$g_{14} - g_{18}$	5	5	18	23430
$g_{19} - g_{23}$	5	6	26	27
$g_{24} - g_{27}$	5	7	18	0.02
$g_{28} - g_{29}$	5	8	7	1.10 e−5
g_{30}	5	9	18	3.38 e−9
g_{31}	5	10	29	4.66 e−13

It is clear from both analytical and empirical approaches that densely connected modules are significantly over-represented. In order to gain some insights in the functions of these modules in PPI networks we concentrated on module g_{31} (5-clique), which is one of the statistically significant modules identified in the yeast network. The functional analysis of the 29 occurrences of module g_{31} obtained by our algorithm reveals two classes of modules. In the first we found cellular protein complexes, such as 26S protease, RNA polymerase II, spliceosome, origin recognition complex, nuclear pore complex, etc. In the second, we found proteins

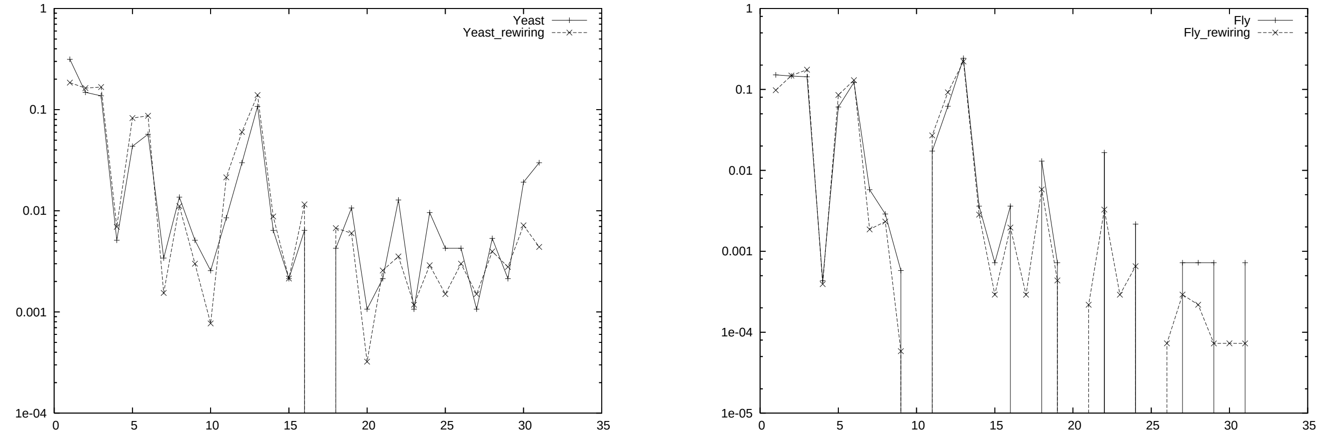


Figure 7: Testing the robustness of our decomposition approach before and after 10% edge rewiring in *S. cerevisiae* (LEFT) and *D. melanogaster* (RIGHT)

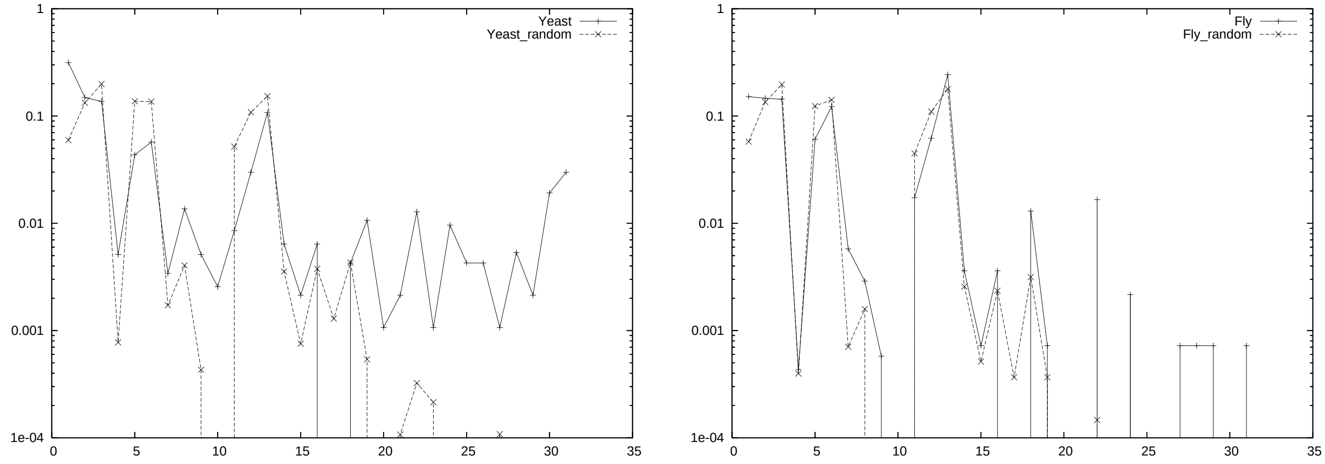


Figure 8: Comparing the occurrences of network modules in *S. cerevisiae* (LEFT) and *D. melanogaster* (RIGHT) against the corresponding random graphs (averaged over 10 random graphs)

that share highly similar functions, of which are involved in transcription regulation, translation initiation, cell cycle control, cellular transportation, mRNA processing, signal transduction cascades, etc. The functional categories of the 29 occurrences of module g_{31} are summarized in Table 3. Examples of the proteins involved in some of the modules g_{31} are given in Table 4. Due to lack of space, we refer the reader to `http://www.cs.ucr.edu/~qyang/` for the complete set of annotations.

Table 3: Distribution of the 5-cliques based on function annotation in *S. cerevisiae* PPI network

Function Category	Number of 5-cliques
Transcription	7
mRNA processing	5
Cell cycle	5
Cellular transportation	4
Metabolism	3
Translation	2
Cytoskeleton	1

We also performed a comparative analysis of the network modules in PPI networks against Internet AS-level networks. The goal of the analysis was to determine whether the over-represented modules in PPI are more or less interconnected than in the AS-level graphs AS4 and AS5. Both PPI and AS-level graphs have a skewed degree distribution. The "rich club connectivity" [32] analysis on the AS4 and AS5 reported one 10-clique among the vertices with the highest degree (data not shown), which is referred as the *core* of the Internet. Figure 9 shows that the yeast PPI has significantly more occurrences of large network modules (e.g., $g_{25}, g_{26}, \ldots, g_{31}$) than AS4 and AS5. Internet AS-level networks are known to have highly connected core structure, where the links inside the core carry higher amount of communication flow than rest of the links in the network. Therefore, links inside the core will have higher betweenness and will be removed first in the decomposition process. The consequence is that in AS-level networks the resulting decomposition will lack these large network modules. In contrast, the highly con-

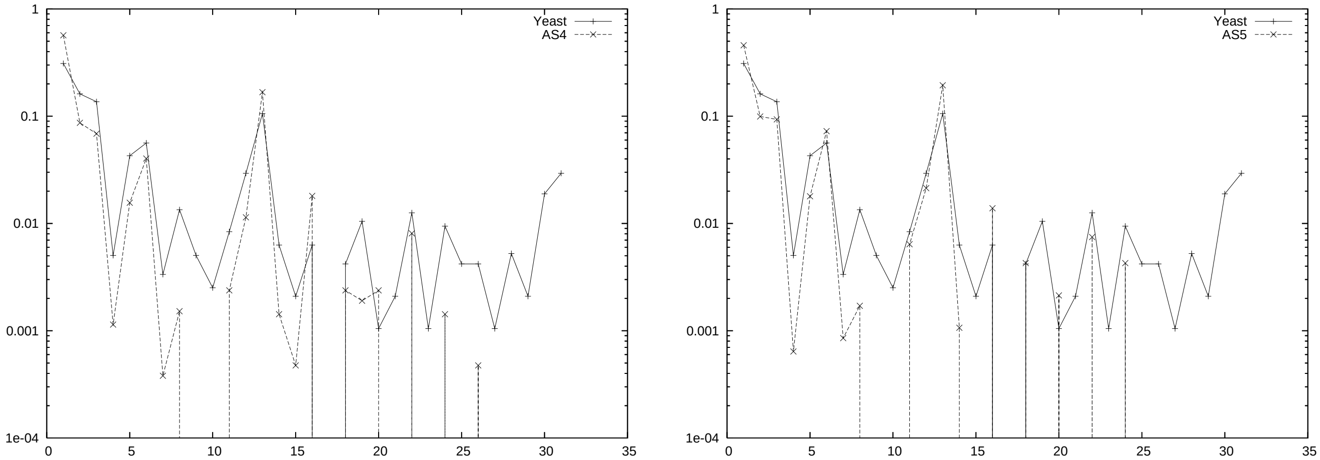


Figure 9: Comparing the occurrences of network modules between *S. cerevisiae* PPI network and the Internet AS-level network AS4 (LEFT) and AS5 (RIGHT)

nected large modules in PPI networks tend to be more frequent and more loosely connected with each other. This may indicate that PPI networks are organized in a decentralized manner across multiple functional domains, inside which strong connections among proteins may constitute the core facility for carrying out specific functions.

5. CONCLUSIONS

In this paper we proposed a new graph decomposition approach that is based on the concept of edge betweenness. The decomposition breaks the network into a set of small network modules, whose frequency of occurrence is then mapped to feature vectors and then normalized. The experiments show that our decomposition method produces normalized feature vectors that more clearly define classes of graphs than the ones produced by the subgraph counting (network motif) approach. More specifically, the analysis of the eigenvalues of the principal components of the covariance matrices shows that our approach extracts a larger number of independent informative features.

Our method turns out to be quite robust to edge rewiring and therefore not over-sensitive to small perturbations to the graph. The analysis of the PPI networks of yeast and fly has identified several over-represented modules when compared to random networks with the same degree distribution, and AS-level Internet graphs. A preliminary investigation on the proteins associated with the cliques found by our decomposition algorithm on the yeast PPI network shows that the proteins involved either belong to the same complex or share similar biological function.

We conclude by addressing some of the limitations of our method that could point to future research direction. The main advantage of a decomposition approach is that one node belongs to only one module, thereby solving the problem of over-counting overlapping subgraphs. However, on PPI graphs this is also a disadvantage because one protein can belong to only one network module, but it is well-known that proteins can be involved in multiple pathways or complexes. In order to capture the notion of "soft-partitioning" on graphs, a radically novel approach might be needed. For example, recent approaches [33] use the notion of information bottleneck [25] to obtain soft partitions of graphs. Also, although our method is not as expensive as the process of counting exhaustively all the subgraphs in a large network, it is still quite computationally intensive. The high computational cost of our method and other graph clustering methods remains an hindrance to their application on large networks.

6. REFERENCES

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules. *Proc. 20th Int. Conf. Very Large Data Bases*, pages 487–499, 1994.

[2] U. Alon. `http://www.weizmann.ac.il/mcb/UriAlon/`.

[3] G. D. Bader and C. W. V. Hogue. An automated method for finding molecular complexes in large protein interaction networks. *BMC Bioinformatics*, 4, 2003.

[4] A.-L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286:509–512, 2002.

[5] U. Brandes. A faster algorithm for betweenness centrality. *Journal of Mathematical Sociology*, 25:163–177, 2001.

[6] R. Dunn, F. Dudbridge, and C. M. Sanderson. The use of edge-betweenness clustering to investigate biological function in protein interaction networks. *BMC Bioinformatics*, 6(39), 2005.

[7] M. Faloutsos, P. Faloutsos, and C. Faloutsos. On power-law relationships of the internet topology. *ACM SIGCOMM'99 Comput. Commun. Rev.*, 29:251–263, 1999.

[8] L. Giot, J. S. Bader, C. Brouwer, and *et al.* A protein interaction map of *Drosophila melanogaster. Science*, 302:1727–1736, 2003.

[9] M. Girvan and M. E. J. Newman. Community structure in social and biological networks. *PNAS*, 99(12):7821–7826, 2002.

[10] K. Gouda and M. J. Zaki. Efficiently mining maximal frequent itemsets. *Proceedings of the 2001 IEEE International Conference on Data Mining*, pages 163–170, 2001.

Table 4: Annotations of some 5-cliques in *S. cerevisiae* PPI network (all the annotation can be found at http://www.cs.ucr.edu/~qyang/**)**

	DIP ID	Description of the proteins	Molecular function
1	DIP:1112N	Pre-mRNA splicing factor PRP19	Involved in pre-mRNA splicing and cell cycle control
	DIP:1682N	Pre-mRNA splicing factor ISY1	
	DIP:1681N	Pre-mRNA splicing factor SYF1	
	DIP:1684N	Pre-mRNA splicing factor SYF2	
	DIP:1685N	Pre-mRNA splicing factor CLF1	
2	DIP:2285N	Origin recognition complex subunit 2	Components of origin recognition complex (ORC)
	DIP:2286N	Origin recognition complex subunit 3	
	DIP:2287N	Origin recognition complex subunit 4	
	DIP:2288N	Origin recognition complex subunit 5	
	DIP:2289N	Origin recognition complex subunit 6	
3	DIP:1704N	Eukaryotic translation initiation factor 3 RNA-binding subunit	Eukaryotic translation initiation factors which bind to the 40S ribosome and promote the binding of methionyl-tRNAi and mRNA
	DIP:2519N	Eukaryotic translation initiation factor 3 90 kDa subunit	
	DIP:5870N	Eukaryotic translation initiation factor 3 110 kDa subunit	
	DIP:4532N	Possible eukaryotic translation initiation factor 3 30 kDa subunit	
	DIP:2303N	Eukaryotic translation initiation factor 5	
4	DIP:1587N	26S protease regulatory subunit 6B homolog	Components of 26S protease complex
	DIP:2883N	26S protease regulatory subunit 7 homolog	
	DIP:2100N	26S proteasome regulatory subunit RPN10	
	DIP:5261N	26S proteasome regulatory subunit RPN9	
	DIP:2808N	Proteasome component C11	
5	DIP:866N	Nucleoporin NUP57	Components of the nuclear pore complex (NPC)
	DIP:709N	Nucleoporin NUP49/NSP49	
	DIP:2074N	Nucleoporin NUP145 precursor	
	DIP:2430N	Nucleoporin NUP84	
	DIP:2721N	Nucleoporin NUP120	

[11] J.-D. J. Han, N. Bertin, T. Hao, D. S. Goldberg, G. F. Berriz, L. V. Zhang, D. Dupuy, A. J. M. Walhout, M. E. Cusick, F. P. Roth, and M. Vidal. Evidence for dynamically organized modularity in the yeast protein-protein interaction network. *Nature*, 430:88–93, 2004.

[12] N. Kashtan, S. Itzkovitz, R. Milo, and U. Alon. Efficient sampling algorithm for estimating subgraph concentrations and detecting network motifs. *Bioinformatics*, 20:1746–1758, 2004.

[13] M. Koyutürk, A. Grama, and W. Szpankowski. An efficient algorithm for detecting frequent subgraphs in biological networks. *Bioinformatics*, 20:i200–i207, 2004.

[14] F. Luo, Y. Yang, C.-F. Chen, R. Chang, J. Zhou, , and R. H. Scheuermann. Modular organization of protein interaction networks. *Bioinformatics*, 23:207–214, 2007.

[15] S. Maslov and K. Sneppen. Specificity and stability in topology of protein networks. *Science*, 296:910–913, 2002.

[16] M. Middendorf, E. Ziv, and C. H. Wiggins. Inferring network mechanisms: The drosophila melanogaster protein interaction network. *PNAS*, 102:3192–3197, 2005.

[17] R. Milo, S. Itzkovitz, N. Kashtan, R. Levitt, S. Shen-Orr, I. Ayzenshtat, M. Sheffer, and U. Alon. Superfamilies of evolved and designed networks. *Science*, 303:1538–1542, 2004.

[18] R. Milo, S. S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon. Network motifs: simple building blocks of complex networks. *Science*, 298:824–827, 2002.

[19] M. E. J. Newman and M. Girvan. Finding and evaluating community structure in newtorks. *Physical Review E*, 69, 026113, 2004.

[20] N. Przulj, D. G. Corneil, and I. Jurisica. Modeling interactome: Scale-free or geometric. *Bioinformatics*, 20:3508–3515, 2004.

[21] F. Radicchi, C. Castellano, F. Cecconi, V. Loreto, and D. Parisi. Defining and identifying communities in networks. *PNAS*, 101:2658–2663, 2004.

[22] A. W. Rives and T. Galitski. Modular organization of cellular networks. *PNAS*, 100(3):1128–1133, 2003.

[23] R. Sharan, T. Ideker, B. P.Kelley, R. Shamir, and R. M.Karp. Identification of protein complexes by comparative analysis of yeast and bacterial protein interaction data. *RECOMB*, pages 282–289, 2004.

[24] S. S. Shen-Orr, R. Milo, S. Mangan, and U. Alon. Network motifs in the transcriptional regulation network of escherichia coli. *Nature Genetics*, 31:64–68, 2002.

[25] N. Tishby, F. Pereira, and W. Bialek. The information

bottleneck method. In *Proceedings of the 37-th Annual Allerton Conference on Communication, Control and Computing*, pages 368–377, 1999.

[26] A. Vazquez, A. Flammini, A. Maritan, and A. Vespignani. Modelling of protein interaction networks. *ComPlexUs*, 1:38–44, 2003.

[27] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:440–442, 1998.

[28] S. Wuchty, Z. N. Oltvai, and A.-L. Barabási. Evolutionary conservation of motif constituents in the yeast protein interaction network. *Nature Genetics*, 35:176–179, 2003.

[29] L. Xenarios, L. Salwinski, X. J. Duan, P. Higney, S.-M. Kim, and D. Eisenberg. DIP, the database of interacting proteins: a research tool for studying cellular networks of protein interactions. *Nucleic Acids Research*, 30:303–305, 2002.

[30] Q. Yang and S. Lonardi. A parallel algorithm for clustering protein-protein interaction networks. *International Journal of Data Mining and Bioinformatics*, 1(3):241–247, 2007.

[31] E. Yeger-Lotem, S. Sattath, N. Kashtan, S. Itzkovitz, R. Milo, R. Y. Pinter, U. Alon, and H. Margalit. Network motifs in integrated cellular networks of transcription-regulation and protein-protein interaction. *PNAS*, 101:5934–5939, 2004.

[32] S. Zhou and R. J. Mondragon. Accurately modeling the internet topology. *Physical Review E Phys. Rev. E*, 70, 2004.

[33] E. Ziv, M. Middendorf, and C. H. Wiggins. Information-theoretic approach to network modularity. *Phys. Rev. E*, 71, 046117, 2005.

Use of Gene Ontology as a Tool for Assessment of Analytical Algorithms with Real Data Sets: Impact of Revised Affymetrix CDF Annotation

Megan Kong[1], Zhongxue Chen[2,3], Yu Qian[1], Jennifer Cai[1], Jamie Lee[1], Eva Rab[1], Monnie McGee[2], Richard H. Scheuermann[1,2]

Department of Pathology[1] and Department of Clinical Sciences[2], University of Texas Southwestern Medical Center, Dallas, TX and Department of Statistical Sciences[3], Southern Methodist University, Dallas, TX

Email: richard.scheuermann@utsouthwestern.edu

ABSTRACT

The Gene Ontology™ (GO) of biological process, molecular function and cellular component terms is the predominant source for functional annotation of gene products. An important use of GO-based annotation has been in the interpretation of gene expression microarray results. One of the challenges to gene expression microarray data analysis and interpretation is that cross-hybridization of probes to the related transcripts can contribute to the signal measured. Several recent studies have reported revised microarray probe annotations designed to circumvent this problem by ensuring that the probe annotation matches the current version of the relevant genome sequence and by eliminating probes with sequence similarity to multiple gene, but the impact of these revised annotations remains to be assessed. Here we describe a general approach of using GO annotation co-clustering characteristics to compare the performance of alternative data mining methods, and apply this approach to assess the impact of improved probe annotation on the results of gene expression microarray data interpretation. Using this approach, we found that revised Affymetrix GeneChip® probe annotation gives rise to improved interpretation of microarray gene expression experiments related to the development, function and transformation of human B lymphocytes.

Keywords

Bioinformatics, gene ontology, microarray data analysis, microarray annotation.

BIOKDD'07, August 12, 2007, San Jose, California, USA.
Copyright 2007 ACM 978-1-59593-839-8/07/0008...$5.00.

1. INTRODUCTION

An ontology is a formal structured vocabulary that captures the semantic relationships between terms. The standardization of terms and their definitions supports data management and exchange in and between bioinformatics systems. In addition, the formal specification of semantic relationships between the vocabulary terms in the ontological structure supports inference and reasoning that can be used to enhance computational data mining.

The Gene Ontology™ (GO) is one of the most successful biomedical ontologies, and includes biological process, molecular function and cellular component terms linked together in a directed acyclic graph with "is_a" and "part_of" relationships [13]. The GO has been used extensively to annotate prokaryotic and eukaryotic gene products based on information described in the scientific literature [2]. An important use of GO-based gene annotation has been to assist in the interpretation of gene expression microarray results [1; 4; 7; 10; 19; 21]. For example, the CLASSIFI algorithm uses GO annotation to classify groups of genes defined by gene cluster analysis using the statistical analysis of GO annotation co-clustering [19].

Gene expression microarrays [12; 24] have fueled a paradigm shift in biomedical research in which reductionistic molecular biology research on individual gene products is augmented by system-level analysis of how the entire transcriptome of a cell population is altered under different normal and pathological conditions. Of the several types of gene expression microarrays that have been developed, the Affymetrix GeneChip® is the most widely used [16]. An Affymetrix GeneChip® can contain from six thousand to more than fifty thousand 25-mer perfect match (PM) oligonucleotide probes with sequences designed to match specific target genes, depending on the organism and platform. Usually the number of PM probes within a probe set is between 11 and 20.

The nature of the microarray technique has brought with it significant challenges in data analysis because of the number of genes being interrogated, the difficulty in controlling and removing the experimental noise, and the need for data normalization to control for inter-experiment variability [11].

Several analytical algorithms, such as MAS5.0 (http://www.affymetrix.com/ products/software/specific/mas.affx), MBEI [20], RMA [16], FARMS [15], and DFW [6] have been developed to deal with these challenges. To assess the performance of these algorithms, a series of data sets were produced in which a group of known transcripts were mixed at known quantities and their levels in the mixture measured using standard microarray methodologies [8; 18]. These so-called "spike-in" data sets provide the ability to assess sensitivity and specificity performance because the true positive and true negative results are known [5; 18; 22].

One of the challenges to gene expression microarray data analysis and interpretation is that cross-hybridization to transcripts related to the target gene of interest can contribute to the fluorescent signal measured. This is especially problematic for the Affymetrix platform because of the relatively short length of each of the oligonucleotide probes. Although the original probe sequences were selected to avoid sequence similarity to related genes, our knowledge of gene and genome sequences has continued to evolve since the current chips were designed. Several recent papers have investigated the quality of Affymetrix GeneChip® probe sets based on current sequence information and found that as much as 30% of the PM probes may be problematic due to potential cross-hybridization and mis-annotation [9; 14; 25]. The Molecular and Behavioral Neuroscience Institute at the University of Michigan (BRAINARRAY, http://brainarray. mbni.med.umich.edu/Brainarray/) has developed new .cdf annotation files for the purposes of annotating Affymetrix chips based on the latest available knowledge of sequences. However, it has been difficult to determine how much this improved annotation will improve the interpretation of Affymetrix GeneChip® using spike-in data sets due to their limited genome coverage.

Here we describe a general approach for using GO annotation information to compare the performance of alternative methods for data mining. The approach is based on the postulate that an improvement in any step in the microarray data analysis pipeline should be reflected in improved co-clustering of related genes in real biomedical data sets. This approach was applied to assess the impact of improved Affymetrix GeneChip® probe annotation on the interpretation microarray gene expression experiments related to the development, function and transformation of human B lymphocytes.

2. METHODS

2.1 *Data Sets*

We used several Affymetrix gene expression data sets selected from the GSE2350 series [3] downloaded from the NCBI GEO database (http://www.ncbi.nlm.nih.gov /projects/geo/) in this study. The "Myc" data set consists of 6 microarray chip measurements from cells that conditionally overexpress the c-Myc proto-oncogene (GSM44096 to GSM44101) and 6 measurements from similar cells that do not (GSM44102 to GSM44107). The "Normal B cell Development" data set consists of 24 measurements of naïve B cell (GSM44133 to GSM44137), centroblast (GSM44143 to GSM44147), centrocyte (GSM44148 to GSM44152), and memory B cell (GSM44138 to GSM44142). The "B cell Response" data set has 18 measurements of Burkitt's lymphoma B cells stimulated with anti-IgM (GSM44063 to GSM44068) or anti-IgM and anti-CD40L (GSM44069 to GSM44074), and unstimulated controls (GSM44051 to GSM44056). For detailed descriptions of each of the data set, please refer to http://www.ncbi.nlm.nih.gov/projects/geo/.

2.2 *Software to Generate the Revised .chp File*

The revised .cdf annotation file was obtained from the University of Michigan website: http://brainarray.mbni.med.umich.edu/ CustomCDF. We have used the file HS95Av2_HS_3REFSEQ_6 (ACSII version) for generating the revised .chp files. The revised annotations found in this file are based on the use of RefSeq sequence records from the RefSeq database of non-redundant and curated sequences. The original annotation package hgu95av2cdf was obtained from Bioconductor (http://www.bioconductor.org /packages/1.9/AnnotationData.html).

For each .cel file, which contains probe-level intensities, .chp file, which contains summarized expression values, were derived following three steps (Figure 1A). This approach to probe set summarization is identical to the default approach supported in the MAS5.0 Affymetrix software. First, the detection p-value was calculated for each probe set using one-sided Wilcoxon signed rank test coded in perl. The default value of $\tau = 0.015$ was used. Then the R value, where R = (PM-MM)/(PM +MM) was calculated for each probe pair. The difference between R and τ was used to calculate the detection p-value for a one-sided Wilcoxon signed rank test. The detection call, present (P), absent (A) or marginal (M), was assigned based on the detection p-value. P-values that were less than 0.04 were assigned a present call, between 0.04 and 0.06 a marginal call, and p-values more than 0.06 received an absent call. The original .chp files and the revised .chp files were generated using both the original .cdf annotation file (HG_U95Av2.cdf) and the revised .cdf annotation file (HS95Av2_HS_3REFSEQ_6.cdf) for each data set, respectively.

Second, the summarized expression value for each probe set was calculated in R using MAS5.0 from the Bioconductor affy package (http://www.bioconductor.org/). For the original .chp file calculation, the hgu95av2cdf_1.10.0.zip package was loaded into R, followed by the MAS 5.0 calculation. For the revised .chp file calculation, we used the following R commands to calculate the summarized expression values for the GSM44096 .chp file from the .cel and .cdf files:

```
data<- ReadAffy('GSM44096.CEL')
data@cdfName <- "HS95Av2_HS_3REFSEQ_6"
s1 = mas5(data)
write.exprs(s1, file="revised_mas5_GSM44096.txt")
```

Third, the final .chp files were generated using Perl code by merging the probe set ID, the number of probe pairs in the probe set, the summarized expression value, the detection call, the detection p-value, and the probe set description.

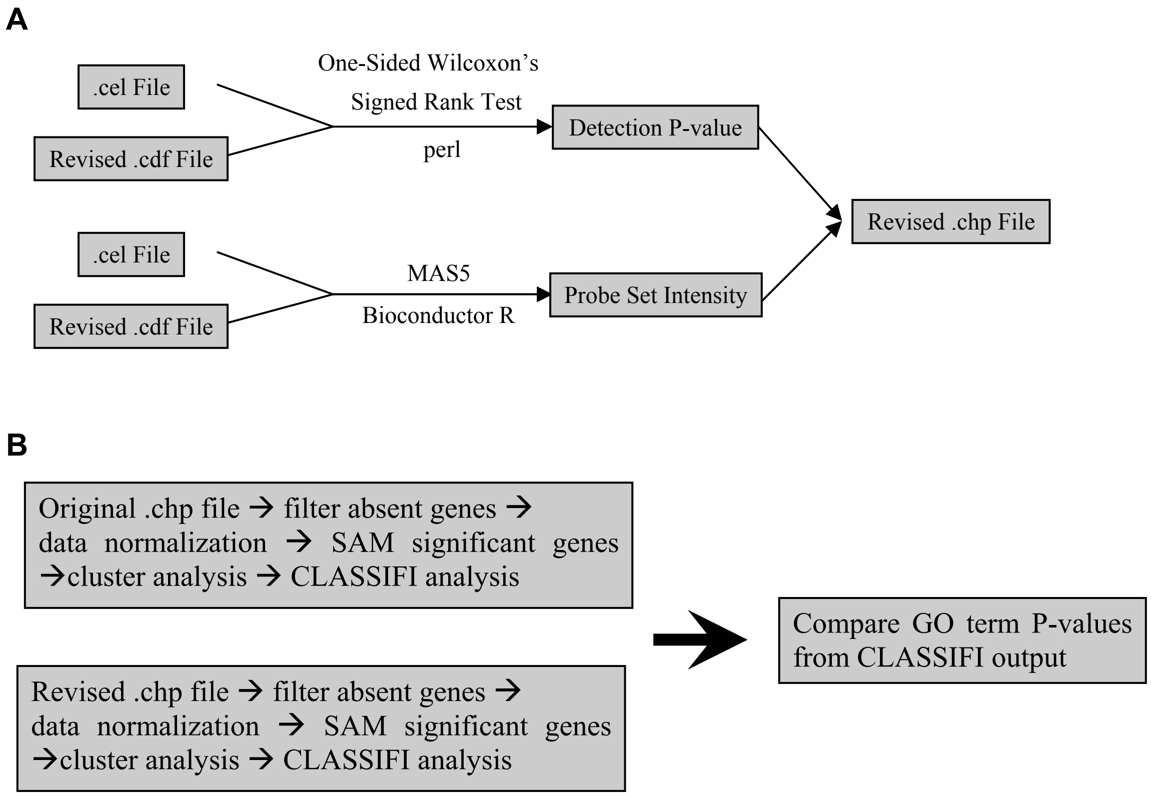

A
.cel File
Revised .cdf File
One-Sided Wilcoxon's
Signed Rank Test
perl
Detection P-value
.cel File
Revised .cdf File
MAS5
Bioconductor R
Probe Set Intensity
Revised .chp File
B
Original .chp file → filter absent genes → data normalization → SAM significant genes →cluster analysis → CLASSIFI analysis
Revised .chp file → filter absent genes → data normalization → SAM significant genes →cluster analysis → CLASSIFI analysis
Compare GO term P-values from CLASSIFI output

Figure 1. Data processing approaches. A. Generation of revised .chp files using revised .cdf file probe set annotations. B. Filtering, normalization, clustering and cluster classification approaches used to compare effects of revised probe set annotation on Affymetrix microarray data analysis.

2.3 Data Analysis

The data analysis approach used is illustrated in Figure 1B. Data filtering, normalization, Significant Analysis of Microarray (SAM) selection of differentially-expressed genes [26] and k-means clustering were performed using TIGR Multiexperiment Viewer (MeV) version 4.0 [23] (http://www.tm4.org/mev.html) as follows. Each data analysis was performed with two categories of samples, with 6 experiment measurements each. For data filtering, only probe sets with at least four present (P) detection calls in either category were selected for further analysis.

Data was normalized by columns; specifically, the signal of each probe set in one measurement was adjusted by the mean and the standard deviation (STD) of the signals for this measurement. The normalized signal value equals (x - mean)/STD, where x is the original summarized expression value.

SAM was used for the selection of differentially-expressed genes. Initial SAM analysis was performed using all the default settings. Several FDR cutoffs (1%FDR, 5%FDR, or 10%FDR) were used.

The combined list of positive and negative differentially-expressed genes from SAM was used to perform k-means clustering analysis. Euclidean distance metric was used in k-means clustering; different numbers of cluster (7, 9, 16, or 20) were used for data set analysis.

The list of all the clusters was saved and formatted to conform to the web-based implementation of CLASSIFI found at: http://pathcuric1.swmed.edu/pathdb/classifi.html. The input format for CLASSIFI is a tab-delimited text file that contains probe set ID (e.g., 13635_at for original annotation, NM_000025_NCBI_refseq for revised annotation), probe set description, and cluster ID. One of the CLASSIFI output files, *classifi_topfile*, displays the GO term that has the lowest co-clustering p-value for each cluster (see Table 1). Another output file of CLASSIFI, *classifi_outputfile*, lists all the GO terms in all the clusters with their co-clustering p-values.

2.4 P-value Distribution Assessment

The p-values of the co-clustering of GO terms were compared between the CLASSIFI results using the original .cdf annotation and the revised .cdf annotation. To compare the lowest GO term p-values for the clusters obtained, we calculated the mean, the median and the range of the log10 transformed the p-values. To compare the whole distribution of all of the co-clustering GO term p-values for all of the clusters, the Wilcoxon rank sum test was utilized to test whether or not there is a statistically significant difference in the distributions. Briefly, for original p-value list1 of size n1 and revised p-value list2 of size n2, p-values from list1

Cluster ID	GO ID	GO Term	GO Type	g	f	c	n	P Value
O1	GO:0006365	35S primary transcript processing	BP	1659	4	62	2	7.86E-03
O2	GO:0005575	cellular component	CC	1659	1425	191	178	7.26E-04
O3	GO:0016763	transferase activity, transferring pentosyl groups	MF	1659	6	22	2	2.44E-03
O4	GO:0006397	mRNA processing	BP	1659	73	130	13	3.35E-03
O5	GO:0006796	phosphate metabolism	BP	1659	124	68	13	1.09E-03
O6	GO:0016879	ligase activity, forming carbon-nitrogen bonds	MF	1659	27	81	6	1.46E-03
O7	GO:0003774	motor activity	MF	1659	16	76	5	5.20E-04
O8	GO:0016814	hydrolase activity, acting on carbon-nitrogen (but not peptide) bonds, in cyclic amidines	MF	1659	4	73	3	3.17E-04
O9	GO:0000165	MAPKKK cascade	BP	1659	17	116	8	6.44E-06
O10	GO:0030333	antigen processing	BP	1659	8	72	4	2.00E-04
O11	GO:0005643	nuclear pore	MF	1659	14	231	9	1.81E-05
O12	GO:0031301	integral to organelle membrane	CC	1659	8	52	3	1.46E-03
O13	GO:0051082	unfolded protein binding	MF	1659	47	39	6	6.00E-04
O14	GO:0019933	cAMP-mediated signaling	BP	1659	3	27	2	7.58E-04
O15	GO:0004871	signal transducer activity	MF	1659	186	47	13	1.30E-03
O16	GO:0042625	ATPase activity, coupled to transmembrane movement of ions	MF	1659	12	44	4	1.83E-04
O17	GO:0003676	nucleic acid binding	MF	1659	396	20	14	1.47E-05
O18	GO:0007165	signal transduction	BP	1659	289	109	38	4.18E-06
O19	GO:0044237	cellular metabolism	BP	1659	899	31	28	1.49E-05
O20	GO:0001568	blood vessel development	BP	1659	7	168	5	1.79E-04
R1	GO:0006968	cellular defense response	BP	1497	9	16	4	1.07E-06
R2	GO:0005625	soluble fraction	CC	1497	25	56	8	1.53E-06
R3	GO:0000062	acyl-CoA binding	MF	1497	4	37	3	5.47E-05
R4	GO:0008624	induction of apoptosis by extracellular signals	BP	1497	9	36	4	3.27E-05
R5	GO:0008204	ergosterol metabolism	BP	1497	3	90	3	2.11E-04
R6	GO:0005635	nuclear envelope	CC	1497	26	194	12	2.99E-05
R7	GO:0005663	DNA replication factor C complex	CC	1497	4	104	4	2.21E-05
R8	GO:0005537	mannose binding	MF	1497	4	61	4	2.50E-06
R9	GO:0000119	mediator complex	CC	1497	6	51	4	1.71E-05
R10	GO:0004556	alpha-amylase activity	MF	1497	6	57	6	2.34E-09
R11	GO:0006809	nitric oxide biosynthesis	BP	1497	4	85	4	9.72E-06
R12	GO:0030529	ribonucleoprotein complex	CC	1497	83	32	12	3.49E-08
R13	GO:0019992	diacylglycerol binding	MF	1497	18	82	12	4.65E-12
R14	GO:0016755	transferase activity, transferring amino-acyl groups	MF	1497	7	69	7	3.27E-10
R15	GO:0019722	calcium-mediated signaling	BP	1497	5	54	5	5.08E-08
R16	GO:0009066	aspartate family amino acid metabolism	BP	1497	2	58	2	1.48E-03
R17	GO:0015980	energy derivation by oxidation of organic compounds	BP	1497	31	44	6	1.93E-04
R18	GO:0004883	glucocorticoid receptor activity	MF	1497	7	69	7	3.27E-10
R19	GO:0005794	Golgi apparatus	CC	1497	50	88	16	5.11E-09
R20	GO:0000079	regulation of cyclin dependent protein kinase activity	BP	1497	5	214	5	5.73E-05

Table 1. Comparison of data analysis results from the original and revised annotation files. The "Myc" data set (see Methods) was used in this analysis. Cluster IDs started with letter "O" or "R" represents the clusters obtained from the original or the revised annotation file respectively. In GO type, "BP", "MF", or "CC" stands for "biological process", "molecular function", or "cellular component" respectively. g, number of probes in data set; f, number of probes with a given ontology in data set; c, number of probes in the gene cluster; n, number of probes with a given ontology (Lee at al., 2006). The GO terms with the lowest p-value in each cluster are displayed.

and list2 were combined and sorted in ascending order. Ranks were assigned to each of the p-value with the smallest p-value getting the rank of 1. The lists of p-values were then separated again and the rank sums were calculated for list1 (sum1) and list2 (sum2). Then, the z-score was calculated based on the approximation:

Average: m= n1*(n1+n2+1)/2
Standard Deviation: d=square root (n1*n2*(n1+n2+1)/12)
Z score: z= (sum1-m)/d

Lastly, the p-value for the distributional comparison was obtained in R using Z-score as input for the function pnorm().

3. RESULTS

Several groups have examined the quality of Affymetrix probe set gene annotation using updated gene/genome sequence information and have found that a substantial number of probes sets are affected [9; 14]. In theory, revisions to the gene annotation of Affymetrix probe sets based on updated genome sequence information would be expected to improve the interpretation of gene expression microarray data. Unfortunately, it has been difficult to assess the impact of revisions to probe set annotation using classical approaches based on the processing of artificial spike-in data sets because a relatively small number of spiked-in transcripts have been used in these data sets and their selection is highly biased toward well-characterized genes. We hypothesized that the co-clustering of genes involved in related biological processes using gene expression microarray data sets from real biological samples could be used to address this limitation, based on the postulate that any improvements made in the pre-processing of gene expression data should result in better co-clustering of related genes.

To test the hypothesis that improvements in annotation translate into improvements in interpretation, we compared the extent of co-clustering of related genes using a series of publicly-available Affymetrix gene expression microarray data sets related to human B lymphocyte development and function - GSE2350 [3]. Initially, a microarray data set generated to assess the impact of c-myc overexpression on gene expression patterns in Burkitt's lymphoma cell lines was evaluated. Details of the data pre-processing procedure employed are described in the Methods section. Briefly (Figure 1), .chp files containing summarized probe set expression values were generated using Affymetrix's original .cdf annotation files provided in Bioconductor (http://www.bioconductor.org/packages/1.9/AnnotationData.html) (original .chp files) and revised .cdf annotation files developed by the University of Michigan group (http://brainarray.mbni.med.umich.edu/CustomCDF) based on updated probe sequence analysis (revised .chp files). The original and revised .chp files were then processed to remove genes that appeared not to be expressed in the samples used (absent calls), to normalize the summarized expression values to give similar distributions, to select genes that are differentially expressed in the data set using the SAM algorithm [26], to group genes together based on their expression patterns using k-means clustering, and to examine the co-clustering of related genes using the CLASSIFI algorithm [19].

Table 1 lists the GO terms showing the most significant co-clustering characteristics for each of the gene clusters using the original .chp files (Clusters #O1 - #O20) and the revised .chp files (Clusters #R1 - #R20) for the Myc data set. As an example, Cluster #O10 contained a total of 72 probe sets (c) with similar expression characteristics. Four of these probe sets recognized genes that were annotated with the GO term "antigen processing" (n). In this data set, 1659 total probe sets were found to be differentially expressed (g), and 8 of these differentially-expressed genes were annotated with the GO term "antigen processing" (f). Based on the hypergeometric distribution, the probability that 4 of the 8 "antigen processing" genes would co-cluster in a gene cluster of size 72 given that there were 1659 genes evaluated in the data set is 2.00E-04. Out of all of the GO terms that were annotated to genes found in Cluster #O10, the GO term "antigen processing" showed the most significant co-clustering (i.e. lowest p-value, the least likely to have co-clustered based on chance alone).

It is difficult to directly compare the co-clustering results derived using the two annotation files based on cluster membership because the numbers and identities of genes that pass the filtering and normalization pre-processing steps differ. However, the extent of co-clustering of related genes can be estimated by assessing the lowest GO term p-values for the clusters obtained. For the Myc data set, using the original .cdf annotation, the mean and median of the log$_{10}$-transformed lowest p-values for the 20 gene clusters were -3.56 and -3.25, respectively, whereas the mean and median of the log$_{10}$-transformed lowest p-values using the revised .cdf annotation were -6.08 and -5.31, respectively. In addition, 10 clusters contained all representative genes for a particular gene ontology term from the entire data set when analysis was performed using the revised .cdf annotation (e.g. all 7 glucocorticoid receptor activity genes were found in gene cluster #R18). Whereas no such case of complete co-clustering was found when analysis was performed using the original .cdf annotation file. These data suggest that the use of the revised .cdf annotation leads to more significant co-clustering of related genes in this data set.

FDR	cdf	Mean	Median	Range
1%	Original	-3.56	-3.25	-2.10 to -5.38
	Revised	-6.08	-5.31	-2.84 to -11.1
5%	Original	-3.50	-3.58	-2.38 to -5.31
	Revised	-6.98	-6.88	-3.56 to -12.1
10%	Original	-4.05	-4.10	-2.12 to -5.39
	Revised	-8.66	-8.80	-3.82 to -13.3

Table 2. Comparison of the co-clustering p-value for most significant GO term using different FDR cut off in SAM analysis. The "Myc" data set (see Methods) was used in this analysis. k-means cluster generated 20 clusters for each FDR cut off. False discovery rate (FDR) cut off of 1%, 5%, or 10% was applied. Values are log10 transformed of the lowest GO term p-value in all the clusters, which represent the mean, the median or the range of the all clusters.

K	cdf	Mean	Median	Range
7	Original	-3.23	-3.12	-2.37 to -4.30
	Revised	-6.62	-5.48	-4.55 to -6.79
9	Original	-3.35	-3.12	-2.26 to -5.84
	Revised	-6.00	-5.94	-3.55 to -9.75
16	Original	-3.54	-3.47	-2.48 to -5.75
	Revised	-6.31	-5.57	-3.84 to -11.4
20	Original	-3.56	-3.25	-2.10 to -5.38
	Revised	-6.08	-5.31	-2.84 to -11.1

Table 3. Comparison of the co-clustering p-value for most significant GO term using different number of clusters in k-means analysis. "Revised" annotation or "Original" annotation files were used in analyzing the "Myc" data set (see Methods). 1% FDR was used for SAM analysis. 7, 9, 16 or 20 clusters were generated from k-means clustering. Values are log10 transformed of the lowest GO term p-value in all the clusters, which represent the mean, the median or the range of the all clusters.

The use of the revised .cdf annotation also appeared to yield GO terms that reach deeper in the GO hierarchy (i.e. more specific process, function and component terms). For example, using the original .cdf annotation, the most significant GO terms for 6 of the 20 gene clusters were represented more than 100 times (f > 100) in the entire data set indicating a relatively common, high-level annotation term, as compared with zero gene clusters with f > 100 using the revised .cdf annotation (Table 1). The mean value of *f* dropped from 178 to 15 using the revised .cdf annotation. The assumption is that a greater number of genes would be annotated with GO terms that describe more general functions (e.g. "signal transduction") than with GO terms that describe more specific functions (e.g. "cell defense response").

Because the numbers and identities of genes that pass the filtering and normalization pre-processing steps differed when using the two different annotation files, it was important to determine if the improved performance of the revised .cdf annotation file was robust to variations in parameters used during data pre-processing steps. Thus, the effects of different false discovery rate (FDR) cutoffs used in the SAM algorithm for the selection of differentially expressed genes were evaluated (Table 2). For all three FDR cutoffs evaluated, the mean, median and range of lowest GO term p-values were all substantially lower when the revised annotation file was used.

The effects of different numbers of clusters used in the k-means algorithm were evaluated next (Table 3). Again, for all four values of *k* evaluated, the mean, median and range of lowest GO term p-values were all substantially lower when the revised annotation file was used.

To determine if the improved performance of the revised .cdf annotation might be dependent on the data set used, we evaluated four addition data sets derived from the GSE2350 series (Table 4). For the first three additional data sets, the revised .cdf annotation again out-performed the original .cdf annotation based on the lowest p-values observed. However, for the fourth additional data set (naïve and memory B cells), the p-value characteristics were much more similar between the two results than was seen with all the other data sets. One possible explanation for this is that the cells used for comparison in this last data set are likely to be much more similar to each other than the cells used in the other data sets. Both naïve and memory B cells are relatively quiescent, and probably only differ from each other by a small subset of genes that change during the relatively small number of differentiation steps between these two cell types. Indeed, the number of differentially expressed genes selected in this data set was much smaller than in the other data sets (611 compared to 1497 from the Myc data set). In the other data sets, many of the comparisons relate to the differences between resting and activated cells of various types, which might be expected to show much larger differences in gene expression patterns.

The previous analyses focused on using the GO terms with the single lowest p-values in each gene cluster for comparison. In order to obtain a more complete picture of related gene co-clustering, the entire distribution of p-values for all GO terms in all gene clusters using the two .cdf annotations was compared (Figure 2). Throughout the distribution, the number of GO terms with relatively low p-values (p < 0.1) was much higher when using the revised .cdf annotation (Figure 2A). For example, the

Data set	cdf	Mean	Median	Range
B cell anti-IgM	Original	-3.33	-3.39	-1.94 to -6.95
	Revised	-6.15	-5.33	-3.48 to -11.4
B cell anti-IgM + anti-CD40	Original	-4.93	-4.01	-2.81 to -15.3
	Revised	-7.54	-6.83	-3.74 to -15.1
Centoblasts& centrocytes	Original	-3.25	-2.89	-1.86 to -10.26
	Revised	-5.67	-4.8	-2.21 to -18.46
Naïve and memory B cells	Original	-4.04	-3.17	-1.98 to -18.02
	Revised	-4.60	-3.96	-1.02 to -7.64

Table 4. Comparison of lowest GO term p-values with different microarray data sets. Values are log10 transformed of the lowest GO term p-value in all the clusters, which represent the mean, the median or the range of the all clusters in each data set. The "B-cell anti-IgM" represents data for B cell stimulated with anti-IgM (GSM44063 to GSM44068) and B cell unstimulated controls (GSM44051 to GSM44056) from the "B cell Response" data set. The "B-cell anti-IgM+anti-CD40L" represents data for B cell stimulated with both anti-IgM and anti-CD40L (GSM44069 to GSM44074), and B cell unstimulated controls (GSM44051 to GSM44056) from the "B cell Response" data set. The "Centroblasts&centrocytes" represents data for centroblasts (GSM44143 to GSM 44147) and centrocytes (GSM44148 to GSM44152) from the "Normal B cell Development" data set. The "Naïve and memory B-cell" represents data for naïve B cell (GSM44133 to GSM44137) and memory B cell (GSM44138 to GSM44142) from the "Normal B cell Development" data set. 5%FDR was applied to all the data sets except "Centroblasts &centrocytes", where 30% FDR was used. Twenty clusters were generated using k-means clustering for all data sets.

number (Figure 2B) and percent (Figure 2C) of GO terms giving p-values below 10^{-3} was three to four times higher using the revised .cdf annotation. The Wilcoxon rank sum test was used to compare the entire distribution of GO term p-values, in two different data sets at two different FDR cutoffs and two different values for k. The differences in the entire p-value distributions in every case were highly statistically significant (Table 5).

4. DISCUSSION

At the present time, there are various tools to analyze microarray data; however, the quality and the validity of these analytical tools need to be assessed fairly. One way to evaluate analytical tools is to analyze spike-in data using these tools and compare the receiver operating characteristic (ROC) curves produced [15; 22]. While this approach is useful, there is some concern that analytical approaches that work well with spike-in data may not work as well with data derived from real, complex biological samples. In addition, ROC analysis is not feasible for real biological data since the true expression values for target mRNA's are rarely known, and so methods of comparison other than ROC curves are needed.

In this study, we examined the possibility of using GO term co-clustering as a comparative tool to assess the impact of using revised annotation on Affymetrix gene expression microarray data analysis. This idea is based on the postulate that genes encoding proteins involved in the same biological process or protein complex will be coordinately expressed; that is, genes that have the same GO annotations are more likely to be in the same gene expression cluster. Thus, better analytical algorithm that gives rise to results that better reflect the underlying biology would be expected to give rise to more significant co-clustering of GO terms. Our analysis comparing the Affymetrix revised and original annotation is the first attempt to use this method to assess the impact the revised annotation has on data analysis. We have analyzed several data sets utilizing different analysis parameters (different FDR, different number of clusters) and calculated the GO term co-clustering probability using the CLASSIFI tool. Our results demonstrate that the p-values for the most significant GO terms in each cluster are significantly lower when using the revised annotation file. In addition, the whole distribution of all the co-clustering p-values for all the GO terms is substantially lower when the revised annotation is used. Thus, using revised annotation indeed produces much more significant co-clustering of related genes. The results showing significant improvement in related gene co-clustering not only suggests that the revised .cdf annotation is better, but also support the general approach of using GO co-clustering as an algorithm evaluation tools with real biological data. In the future, we plan to use the co-clustering method to compare the performance of various preprocessing algorithms on real data sets.

Use of gene ontology terms to help interpret systems-level biological data is a great addition to the data analysis arsenal [1; 4; 10; 21; 19]. Although this study has focused on microarray analysis, the same methodology can be applied to other large-scale data sets in which large sets of genes/proteins are analyzed (e.g. protein-protein and genetic interaction networks). Thus, in addition to aiding in the understanding of the functions played by genes and proteins in the cell, the Gene Ontology can also play a role in assisting in the development of improved data mining approaches that reveal underlying functional properties in complex biological systems.

Data set	1% FDR		5%FDR	
	k=9	k=16	k=9	k=16
B cell anti-IgM + anti-CD40	1.41E-25	5.18E-27	3.52E-19	4.96E-24
"Myc" data set	5.31E-18	3.01E-32	1.21E-19	2.98E-27

Table 5. Significant differences in the p-value distributions between the results from the original and the revised annotation. To compare the entire distribution of all GO term p-values for all gene clusters using the two annotation files, Wilcoxon rank sum test was employed to test whether or not there is a statistically significant difference in the distributions. 1% or 5%FDR was used in SAM. 9 or 16 clusters were used for k-means cluster. The B-cell stimulated with both anti-IgM and anti-CD40L (GSM44069 to GSM44074) and the B-cell unstimulated controls (GSM44051 to GSM44056) from the "B cell Response" data set and the "Myc" data set (see Methods) were used in this analysis.

5. ACKNOWLEDGEMENTS

The authors wish to acknowledge support for this work from Dr. Milton Packer, Director, Department of Clinical Sciences, University of Texas Southwestern Medical Center and from the National Institutes of Health contracts N01-AI40076, N01-AI40041 to RHS and 1 R15GM079742-01 to MM.

6. REFERENCES

[1] Al-Shahrour, F., Díaz-Uriarte R., and Dopazo, J. FatiGO: a web tool for finding significant associations of Gene Ontology terms with groups of genes. *Bioinformatics*, 20, 578-580, 2004.

[2] Ashburner, M., Ball, C. A., Blake, J. A., Botstein D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M., Rubin, G.M. and Sherlock, G. Gene Ontology: tool for the unification of biology. *Nature Genetics,* 25, 25 – 29, 2000.

[3] Basso, K., Margolin A.A., Stolovitzky, G., Klein, U., Dalla-Favera, R., Califano, A. Reverse engineering of regulatory networks in human B cells. *Nature Genetics* 37, 382-90, 2005.

[4] Bluthgen, N., Brand, .K, Cajavec, B., Swat, M., Herzel, H. and Beule, D. Biological profiling of gene groups utilizing Gene Ontology. *Genome Inform.,* 16, 106-15, 2005.

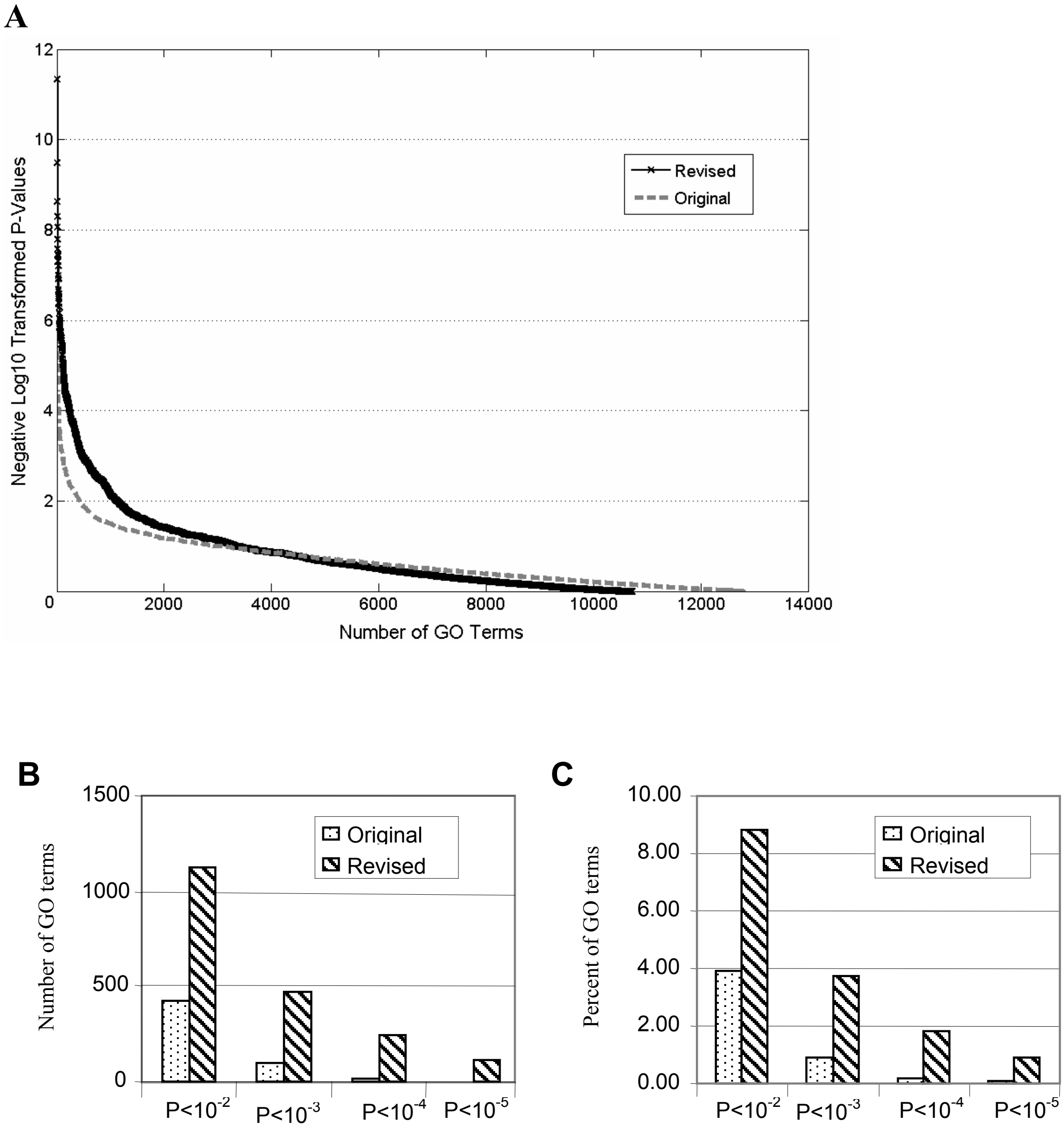


Figure 2. Comparison of GO term p-value distributions between original and revised Affymetrix probe set annotation. The "Myc" data set (see Methods) was used in this analysis. 1%FDR for SAM analysis and 20 clusters for k-means cluster were applied.
A. The distribution of all the p-values for all the GO terms in every cluster. The curve marked with "x" represents the co-clustering p-values using the revised annotation. The dashed line represents the co-clustering p-values using the original annotation. Number of GO terms (B) or the percent of GO terms (C) are plotted against different p-value cut off ($P < 10^{-2}$, 10^{-3}, 10^{-4}, or 10^{-5}).

[5] Bolstad, B.M., Irizarry, R.A., Åstrand, M. and Speed, T.P. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *Bioinformatics*, 19, 185-193, 2003.

[6] Chen, Z., McGee, M., Liu, Q., Scheuermann, R. H. A distribution free summarization method for Affymetrix GeneChip arrays. *Bioinformatics*, 23, 321-7, 2007.

[7] Conesa, A., Gotz, S., Garcia-Gomez, J.M., Terol, J., Talon, M. and Robles, M. Blast2GO: a universal tool for annotation, visualization and analysis in functional genomics research. *Bioinformatics*, 21, 3674-6, 2005.

[8] Cope, L. M., Irizarry, R.A., Jaffee, H.A., Wu, Z., and Speed, T.P. (2004) A benchmark for Affymetrix GeneChip expression measures. *Bioinformatics*, 20, 323-331, 2004.

[9] Dai, M., Wang, P., Boyd, A.D., Kostov, G., Athey, B., Jones, E.G., Bunney, W.E., Myers, R.M., Speed, T.P., Akil, H., Watson, S.J. and Meng, F. Evolving gene/transcript definitions significantly alter the interpretation of GeneChip data. *Nucleic Acids Res.,* 33, e175, 2005.

[10] Doniger, S.W., Salomonis, N., Dahlquist, K.D., Vranizan, K., Lawlor, S.C., and Conklin, B.R. MAPPFinder: using Gene Ontology and GenMAPP to create a global gene-expression profile from microarray data. *Genome Biol.*, 4, R7, 2003.

[11] Draghici, S., Khatri, P., Eklund A.C. and Zoltan S. Reliability and reproducibility issues in DNA microarray measurements. *Trends Genet.,* 22, 101-9, 2006.

[12] Eisen, M.B., Spellman, P.T., Brown, P. O., and Botstein, D. Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci. U.S.A.,* 95, 14863–14868. 1998.

[13] The Gene Ontology Consortium. Creating the Gene Ontology Resource: Design and Implementation. *Genome Res*., 11, 1425-1433, 2001.

[14] Harbig, J., Sprinkle, R., and Enkemann, S. A. A sequence-based identification of the genes detected by probesets on the Affymetrix U133 plus 2.0 array. *Nucleic Acids Res.,* 33, e31, 2005.

[15] Hochreiter, S., Clevert, D., and Obermayer, K. A new summarization method for affymetrix probe level data. *Bioinformatics*, 22, 943-949, 2006.

[16] Irizarry, R.A., Hobbs, B., Collin, F., Beazer-Barclay, Y.D., Antonellis, K.J., Scherf, U. and Speed, T.P. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics* 4, 249-264, 2003.

[17] Irizarry, R.A., Bolstad, B.M., Collin, F., Cope, L.M., Hobbs, B. and Speed, T.P. Summaries of Affymetrix GeneChip probe level data. *Nucleic Acids Res.*, 31, e15, 2003.

[18] Irizarry, R.A., Wu, Z., and Jaffee, H. A. Comparison of Affymetrix GeneChip expression measures. *Bioinformatics,* 22, 789-794, 2006.

[19] Lee, J.A., Sinkovits, R.S., Mock, D., Rab, E.L., Cai, J., Yang, P., Saunders, B., Hsueh, R.C., Choi, S., Subramaniam, S., Scheuermann, R.H. in collaboration with the Alliance for Cellular Signaling. Components of the antigen processing and presentation pathway revealed by gene expression microarray analysis following B cell antigen receptor (BCR) stimulation. *BMC Bioinformatics,* 7, 237, 2006.

[20] Li, C., and Wong, W.H. Model-based analysis of oligonucleotide arrays: Expression index computation and outlier detection. *Proc Natl Acad Sci U S A.*, 98, 31-36, 2001

[21] Martin, D.M., Berriman, M., and Barton, G.J. GOtcha: a new method for prediction of protein function assessed by the annotation of seven genomes. *BMC Bioinformatics*, 5, 178, 2004.

[22] McGee., M. and Chen, Z. New spike-in data sets for the Affymetrix HG-U133a Latin square experiement. COBRA Reprint Series, Article 5, 2006.

[23] Saeed, A. I., Bhagabati, N. K., Braisted, J. C., Liang, W., Sharov, V., Howe, E. A., Li, J., Thiagarajan, M., White, J. A. and Quackenbush, J. TM4 Microarray Software Suite, *Methods Enzymol.,* 411, 134-193, 2006.

[24] Schena, M., Shalon, D., Davis, R.W. and Brown, P.O. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science* 270, 467-70, 1995.

[25] Shi, L., Reid, L.H., Jones, W.D., Shippy, R., Warrington, J.A., Baker, SC., Collins, P.J., de Longueville, F., Kawasaki, E.S., Lee, K.Y., Luo, Y. et al. The MicroArray Quality Control (MAQC) project shows inter- and intraplatform reproducibility of gene expression measurements. *Nat Biotechnol.* 24, 1151-1161, 2006.

[26] Tusher, V.G., Tibshirani, R. and Chu, C. Significance analysis of microarrays applied to the ionizing radiation response. *Proc Natl Acad Sci U S A.*, 98, 5116–5121, 2001

Clustering of Non-Alignable Protein Sequences

Abdellali Kelil
Department of Computer Sciences
University of Sherbrooke
Sherbrooke, QC, Canada
1 (819) 823 8616
Abdellali.Kelil@USherbrooke.ca

Shengrui Wang
Department of Computer Sciences
University of Sherbrooke
Sherbrooke, QC, Canada
1 (819) 821 8000 ext 62022
Shengrui.Wang@USherbrooke.ca

Ryszard Brzezinski
Department of Biology
University of Sherbrooke
Sherbrooke, QC, Canada
1 (819) 821 8000 ext 61077
Ryszard.Brzezinski@USherbrooke.ca

ABSTRACT

We are interested in the problem of grouping families of non-alignable protein sequences, such as circular-permutation, multi-domain and tandem-repeat proteins, into clusters (classes) of related biological functions. For such sequences, whose numbers are constantly growing, the commonly used alignment-dependent approaches fail to yield biologically plausible results. To the best of our knowledge, no automatic process yet exists to carry out clustering on these proteins. Biologists often use more complex manual approaches based on secondary and tertiary structures, which require considerably more resources and time.

In this paper, we develop a new similarity measure SMS, applied directly on non-aligned sequences. It allows us to develop a new and original alignment-free algorithm, named CLUSS, for clustering protein families based on a spectral decomposition approach inspired by the latent semantic analysis (LSA) widely used in information retrieval. CLUSS, utilized jointly with SMS, is effective on both alignable and non-alignable protein sequences. To show this, we have extensively tested our algorithm on different benchmark protein databases and families; we have also compared its performance with many alignment-dependent mainstream algorithms. The source code, the application server, and all experimental results are available at CLUSS web site http://prospectus.usherbrooke.ca/CLUSS/.

Categories and Subject Descriptors

J.3 [**Life and Medical Sciences**]: Biology and Genetics; I.5.3 [**Pattern Recognition**]: Clustering

General Terms

Algorithms, Measurement, Experimentation

Keywords

Clustering, Phylogenetic, Biological Function, Protein Sequences, Matching, Similarity Measure, Alignable, Non-Alignable

BIOKDD'07, August 12, 2007, San Jose, California, USA.
Copyright 2007 ACM 978-1-59593-839-8/07/0008...$5.00.

1. INTRODUCTION

With the rapid burgeoning of protein sequence data, the number of proteins for which no experimental data are available greatly exceeds the number of functionally characterized proteins. To predict a function for an uncharacterized protein, it is necessary not only to detect its similarities to proteins of known biochemical properties (i.e., to assign the unknown protein to a family), but also to adequately assess the differences in cases where similar proteins have different functions (i.e., to distinguish among subfamilies). One solution is to cluster each family into distinct subfamilies composed of functionally related proteins. Subfamilies resulting from clustering are easier to analyze experimentally. A subfamily member that attracts particular interest need to be compared only with the members of the same subfamily. A biological function can be attributed with high confidence to an uncharacterized protein, if a well-characterized protein within the same cluster is already known. Conversely, a biological function discovered for a newly characterized protein can be extended over all members of the same subfamily.

Almost all automatic clustering approaches deal with only aligned protein sequences, which are performed via alignment algorithms such as the widely known MUSCLE [8], ClustalW [36], MAFFT [18] and T-Coffee [26], and many others. These algorithms often provide information on both conserved and mutated motifs, making it a good approach for measuring similarities between protein sequences. However, they have several serious limitations, including the following:

- **Dependence on the algorithm used.** The results depend heavily on the algorithm selected and the parameters set by the user for the alignment algorithm (e.g., gap penalties). As far as easily-alignable proteins are concerned, almost every existing alignment algorithm can yield good results. However, for protein sequences that are difficult to align, each alignment algorithm finds its own solution. Such variable results create ambiguities and can complicate the clustering task [25].

- **Problem of non-alignable sequences.** For the case of non-alignable protein sequences (i.e., not yet definitively aligned), alignment-based algorithms do not succeed in producing biologically plausible results. This is due to the nature of the alignment approaches, which are based on the matching of subsequences in equivalent positions, while non-alignable proteins often have similar and conserved domains in non-equivalent positions [25], such as circular-permutation, multi-domain and tandem-repeat proteins

There are other known difficulties that limit the reliability of alignment, especially for the case of hard-to-align protein sequences, such as "*repeat*", "*substitution*" and "*gap*" problems, which are well discussed by Higgins [15].

The number of protein sequences that are hard-to-align or not alignable at all is rapidly increasing. These proteins are frequently related to important biological phenomena, and their classification is of primary importance for the comprehension of these phenomena. One example is the group of 33 (α/β)8-barrel proteins belonging to the Glycoside Hydrolase (GH) family [35], which has an important role in the physiology of the alive cell, as discussed in [5,13]. A large number of these are still uncharacterized, since to date the process has been carried out manually with complicated approaches, such as those employed by Côté *et al.* [5] and Fukamizo *et al.* [13] for the characterization of the 33 (α/β)8-barrel proteins belonging to the GH [35] family. Most of the tools currently available are based on the alignment of protein sequences, making them inappropriate for this kind of proteins.

Our aim in this paper is to develop a new approach to the biological interpretation of protein sequences, especially those which cause problems for alignment-dependent algorithms. Our work is an attempt to build an algorithm to help biologists perform analyses of certain kinds of protein sequences, which are now carried out almost manually. In the rest of the paper, we use the terms subfamily and cluster interchangeably.

2. RELATED WORK

The literature reports a number of algorithms for clustering protein databases, such as the widely used algorithm BLAST [1] and its improved versions Gaped-Blast and PSI-Blast [2], and SYSTERS [23], ProtClust [29] and ProtoMap [40] (see [32] for a review). These algorithms have been designed to deal with large sets of proteins by using various techniques to accelerate examination of the relationships between proteins. However, they are not very sensitive to the subtle differences among similar proteins. Consequently, these algorithms are not effective for clustering protein sequences in closely related families. On the other hand, more specific algorithms have also been developed, for instance, the widely cited algorithms BlastClust [3]**,** which uses score-based single-linkage clustering, TRIBE-MCL [10]**,** based on a Markov clustering approach, and gSPC [34]**,** based on a method that is analogous to the treatment of an inhomogeneous ferromagnet in physics. Almost all of these algorithms are either based on sequence alignment or rely on alignment-dependent algorithms for computing pair-wise similarities.

3. APPROACH OVERVIEW

In this paper, we propose an efficient and original algorithm, CLUSS, for clustering protein families based on a new alignment-free measure we propose for protein similarity. The novelty of CLUSS resides essentially in two features. First, CLUSS is applied directly to non-aligned sequences, thus eliminating the need for aligned sequences. Second, it adopts a new measure of similarity, directly exploiting the substitution matrices generally used to align protein sequences and showing a great sensitivity to the relations among similar and divergent protein sequences. CLUSS can be summarized as follows:

Given *F***,** a family containing a given number of proteins:

1. Build a pairwise similarity matrix for the proteins in *F* using SMS our new similarity measure.
2. Create a phylogenetic tree of the protein family *F* using our new clustering approach.
3. Assign a co-similarity value to each node of the tree.
4. Calculate a critical threshold for identifying subfamily branches, by computing the interclass inertia [7].
5. Collect each leaf from its subfamily branch into a distinct subfamily.

4. SMS: SIMILARITY MEASURE

Many approaches to measuring the similarity between protein sequences have been developed. Prominent among these are alignment-dependent approaches, including the well-known algorithm BLAST [1] and its improved versions Gaped-Blast and PSI-Blast [2], whose programs are available at [3], as well as several others such as the one introduced by Varré *et al.* [37] based on movements of segments, and the recent algorithm Scoredist introduced by Sonnhammer *et al*. [33] based on the logarithmic correction of observed divergence. These approaches often suffer from accuracy problems, especially for multi-domain proteins (in general case hard-to-align protein sequences). The similarity measures used in these approaches depend heavily on the alignability of the protein sequences. In many cases, alignment-free approaches can greatly improve protein comparison, especially for non-alignable protein sequences. These approaches have been reviewed in detail by several authors [30,31,9,38]**.** Their major drawback, in our opinion, is that they consider only the frequencies and lengths of similar regions within proteins and do not take into account the biological relationships that exist between amino acids. To correct this problem, some authors [9] have suggested the use of the Kimura correction method [22] or other types of correction, such as that of Felsenstein [12]. However, to obtain an acceptable phylogenetic tree, the approach described in [9] performs an iterative refinement including a profile-profile alignment at each iteration, which significantly increases its complexity.

To overcome these difficulties of alignment-based approaches, we have developed SMS a new approach inspired by biological considerations and known observations related to protein structure and evolution. The goal is to make efficient use of the information contained in amino acid subsequences in the proteins, which leads to a better similarity measurement. The principal idea of our approach is to use a substitution matrix such as BLOSUM62 [14] or PAM250 [6] to measure the similarity between matched amino acids from the protein sequences being compared.

4.1 Matching score

In this section, we will use the symbol $|.|$ to express the length of a sequence. Let X and Y be two protein sequences belonging to the protein family F. Let x and y be two identical subsequences belonging respectively to X and Y; we use $\Gamma_{x,y}$ to represent the matched subsequence of x and y. We use l to represent the minimum length that $\Gamma_{x,y}$ should have (i.e., we will be interested only in $\Gamma_{x,y}$ whose length is at least l residues). We define E^l_{XY}, the key set of matched subsequences $\Gamma_{x,y}$ for the definition of our similarity function, as follows (see Figure 1 for an example):

$$E^l_{X,Y} = \left\{ \Gamma_{x,y} \middle| \begin{array}{l} \left|\Gamma_{x,y}\right| \geq l\ , \\ \forall \Gamma_{x',y'} \in E^l_{X,Y}, \Gamma_{x',y'} \neq \Gamma_{x,y} \Rightarrow \left(x' \not\subset x \vee y' \not\subset y\right) \end{array} \right\} \tag{1}$$

The expression ($x' \not\subset x$) means that x' is not included in x, either in terms of the composition of the subsequences or in terms of their respective positions in X. The matching set E^l_{XY} contains all the matched subsequences of maximal length between the sequences X and Y. It will be used to compute the matching score of the sequence pair.

The formula E^l_{XY} adequately describes some known properties of polypeptides and proteins. First, protein motifs (i.e., series of defined residues) determine the tendency of the primary structure to adopt a particular secondary structure, a property exploited by several secondary-structure prediction algorithms. Such motifs can be as short as four residues (for instance those found in β-turns), but the propensity to form an α-helix or a β-sheet is usually defined by longer motifs. Second, our proposal to take into account multiple (i.e., ≥2) occurrences of a particular motif reflects the fact that sequence duplication is one of the most powerful mechanisms of gene and protein evolution, and if a motif is found twice (or more) in a protein it is more probable that it was acquired by duplication of a segment from a common ancestor than by acquisition from a distant ancestor.

The construction of $E^l_{X,Y}$ requires a CPU time proportional to $|X|*|Y|$. In practice, however, several optimizations are possible in the implementation, using encoding techniques to speed up this process. In our implementation of SMS, we used a technique that improved considerably the speed of the algorithm; we can summarize it as follows:

By the property that all possible matched subsequences satisfy $|\Gamma x,y\ | \geq l$, we know that each $\Gamma x,y$ in $E^l_{X,Y}$ is an expansion of a matched subsequence of length l. Thus we first collect all the matched subsequences of length l, which takes linear time. Secondly, we expand each of the matched subsequences as much as possible on the both left and right sides. And finally, we select all the expanded matched sequences that are maximal according to the inclusion criterion. This technique is very efficient for reducing the execution time in practice. However, due to the variable lengths of the matched sequences, it may not be possible to reduce the worst-case complexity to a linear time. In the Results section, we provide a time comparison between our algorithm and several existing ones.

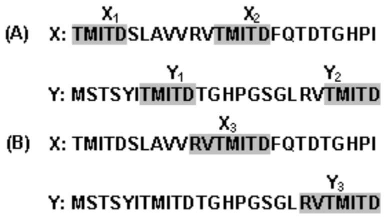

X1
X2
(A)
X: TMITDSLAVVRVTMITDFQTDTGHPI
Y1
Y2
Y: MSTSYITMITDTGHPGSGLRVTMITD
X3
(B)
X: TMITDSLAVVRVTMITDFQTDTGHPI
Y3
Y: MSTSYITMITDTGHPGSGLRVTMITD

Figure 1. Matching subsequences

Figure 1 shows an example of $E^l_{X,Y}$ construction, with l=4. Let X and Y be two protein sequences, as illustrated. Among the matches shown in Figures 1.A and 1.B, the matched subsequence Γ_1 of X_1 and Y_1, will be added to the matching set $E^4_{X,Y}$. Similarly, for Γ_2 the match of X_1 and Y_2, and Γ_3 the match of X_2 and Y_1 will also be added to the matching set E^4_{XY}. On the other hand, since $X_2 \subset X_3$ and $Y_2 \subset Y_3$, Γ_4 the matched subsequence of X_2 and Y_2, will not be added to E^4_{XY}. Instead, Γ_5 the match of X_3 and Y_3, will be added to the set E^4_{XY}, even though X_3 overlaps with X_2.

Let M be a substitution matrix, and Γ a matched subsequence belonging to the matching set E^l_{XY}. We define a weight $W(\Gamma)$ for the matched subsequence Γ, to quantify its importance compared to all the other subsequences of E^l_{XY}, as follows:

$$W\left(\Gamma\right) = \sum_{i=1}^{|\Gamma|} M\left[\Gamma\left[i\right], \Gamma\left[i\right]\right] \quad (2)$$

where $\Gamma[i]$ is the i^{th} amino acid of the matched subsequence Γ, and $W[\Gamma[i],\Gamma[i]]$ is the substitution score of this amino acid with itself. Here, in order to make our measure biologically plausible, we use the substitution concept to emphasize the relation which binds one amino acid with itself. The value of $M[\Gamma[i],\Gamma[i]]$ (i.e., entries on the diagonal of the substitution matrix) estimates the rate at which each possible amino acid in a sequence remains unchanged over time; in other words, $W(\Gamma)$ measures the conservability of the matched subsequence Γ in both X and Y, which is an important concept in biology that emphasizes the importance of each region of the protein sequence.

Now we define S the matrix of matching scores, such as $S_{X,Y}$ is the matching score between X and Y two protein sequences belonging to the family F. The matching score $S_{X,Y}$, understood as representing the substitution relation of the conserved regions in both sequences, is defined as follows:

$$S_{X,Y} = \frac{\sum_{\Gamma \in E^l_{X,Y}} W\left(\Gamma\right)}{MAX\left(|X|,|Y|\right)} \quad (3)$$

Finally, the pairwise similarity matrix SMS of the protein family F is calculated by applying the Pearson's correlation coefficient to the matrix S.

4.2 Minimum length *l*

Our aim is to detect and make use of the significant motifs best conserved during evolution and to minimize the influence of those motifs which occur by chance. This motivates one of the major biological features of our similarity measure, the inclusion of all long conserved subsequences (i.e., multiple occurrences) in the matching, since it is well known that the longer the subsequences, the smaller the chance of their being identical by chance, and vice versa. Here we make use of the theory developed by Karlin *et al.* in [21,19,20] to calculate, for each pair of sequences, the value of l, the minimum length of matched subsequences. According to theorem 1 in [19] we have:

$$K_{r,N} = \frac{\log n\left(|Seq_1|,\ldots,|Seq_N|\right) + \log\lambda\left(1-\lambda\right) + 0.577}{-\log\lambda} \quad (4)$$

$$n\left(|Seq_1|,\ldots,|Seq_N|\right) = \sum_{1 \leq i_1 \leq \ldots \leq i_r \leq N} \prod_{v=1}^{r} \left|Seq_{i_v}\right| \quad (5),$$

$$\lambda = \max_{1 \leq v_1 \leq \ldots \leq v_r \leq N} \left(\sum_{i=1}^{m} \prod_{j=1}^{r} p_i^{\left(v_j\right)} \right) \quad (6),$$

$$\sigma_{r,N} \approx \frac{1.283}{\left|\log\lambda\right|} \quad (7)$$

This formula calculates $K_{r,N}$, the *expected length of the longest common word present by chance at least r times out of N m-letter sequences* [19] (i.e., $Seq_1,\ldots,Seq_N$), where $p_i^{(v)}$ is generally specified as the i^{th} residue frequency of the observed v^{th} sequence, and $\sigma_{r,N}$ the asymptotic standard deviation of $K_{r,N}$.

According to the conservative criterion proposed by Karlin *et al.* .[19], to measure the similarity between two protein sequences, we take into account all subsequences present 2 times out of the 2 sequences which have a length that exceeds $K_{r,N}$ by at least two standard deviations. In other words, for each pair of sequences, matched subsequences shorter than $l=K_{2,2}+2.\sigma_{2,2}$ (i.e., by fixing $N=r=2$) have a real chance of being similar as a result of random phenomena, while those with lengths greater than $l=K_{2,2}+2.\sigma_{2,2}$ are more likely to be conserved motifs. So, for each pair of protein sequences X and Y, we calculate a specific and appropriate value of l to calculate $S_{X,Y}$ the similarity between X and Y.

5. CLUSS: CLUSTERING ALGORITHM

CLUSS is composed of three main stages. The first one consists in building SMS, a pair-wise similarity matrix; the second, in building a phylogenetic tree according to this matrix, using a new clustering approach based on spectral decomposition; and the third, in identifying subfamily nodes from which leaves are grouped into subfamilies.

5.1 Stage 1: Similarity matrix SMS

Using one of the known substitution score matrices, such as BLOSUM62 [14] or PAM250 [6], we compute *SMS*, the *NxN* similarity matrix, where *N* is the number of sequences of the protein family *F* to be clustered, and $SMS_{i,j}$ is the similarity between the i^{th} and the j^{th} protein sequences of *F*. The construction of *SMS* takes CPU time proportional to $N(N-1)T^2/2$, with *T* the typical sequence length of the *N* sequences.

5.2 Stage 2: Phylogenetic tree

To build the phylogenetic tree, we adopt a strategy inspired by the latent semantic analysis approach (LSA) [4], widely used in information retrieval, in which data are mapped to a vector space of reduced dimension (i.e., less than the number of data). By using a hierarchical strategy, and starting from the protein sequences, each of which is represented by a vector in a Euclidian space (i.e., step 1 of this stage), and considered as the root node of a (sub)tree containing only one node, we iteratively join a pair of root nodes in order to build a bigger subtree. At each iteration, a pair of root nodes is selected if they are the most similar root nodes (i.e., corresponding vectors have the largest cosine product). This process ends when there remains only one (sub)tree, which is the phylogenetic tree. The present stage is composed of three steps, as follows:

5.2.1 Step1: Spectral decomposition of SMS

The main idea is to perform a spectral decomposition of the similarity matrix *SMS*, to map the protein sequences onto a vector space, thereby making use of its advantages, of which the most important for us is the conservability of distances.

Spectral decomposition of the square symmetric matrix *SMS* is done through Eigen decomposition [39]. We obtain:

$$SMS = V * V^T \quad (8)$$

$$V = \begin{pmatrix} V_1 \\ \vdots \\ V_N \end{pmatrix} \approx \begin{pmatrix} u_1^1 & \cdots & u_p^1 \\ \vdots & \ddots & \vdots \\ u_1^N & \cdots & u_p^N \end{pmatrix} * \begin{pmatrix} \sqrt{\lambda_1} & & \\ & \ddots & \\ & & \sqrt{\lambda_p} \end{pmatrix} \quad (9)$$

where $\lambda_1,\ldots,\lambda_p$ are the *p* non-negative eigenvalues of *SMS* and $u_1,\ldots,u_p$ are the *p* eigenvectors corresponding to the *p* eigenvalues.

For two vectors V_X and V_Y, in Z^N, representing the protein sequences *X* and *Y*, respectively, the Euclidian inner product is defined as:

$$SMS_{X,Y} = \langle V_X, V_Y \rangle = \sum_{i=1}^{N} v_i^X * v_i^Y \quad (10)$$

When properly normalized (i.e., as proposed in section 4.1), the matrix *SMS* measures the correlation between protein sequences, which is similar to the role of the covariance matrix in principal component analysis (PCA). However, in the conventional PCA method, we must subtract the averages from the covariance matrix, which means that our method is not a PCA approach.

5.2.2 Step 2: Building the tree

The similarity between two root nodes referred to above is computed in the following way. At the beginning of the iteration, the similarity between any pair of nodes is initialized by the cosine product. We assign to each root node *L* (i.e., an individual leaf represents one protein sequence) a co-similarity c_L according to its importance in *F*.

By taking into account information about the neighborhood around each of the nodes *L* and *R*, the concept of co-similarity reflects the cluster compactness of all the sequences (leaf nodes) in the subtree. In fact, its value is inversely proportional to the within-cluster variance. As the subtree becomes larger, the co-similarity tends to become smaller, which means that the sequences within the subtree become less similar and the difference (separation) between sequences in different clusters becomes less significant. In simpler terms, the co-similarity is a measure of the balance between two nodes.

At the first iteration, all co-similarities are initialized to zero. Let *L* and *R* be the two most similar root nodes (i.e., cosine product of V_L and V_R is the largest) at a given iteration step; they are joined together to form a new subtree. Let *P* be the root node of the new subtree. *P* thus has two children, *L* and *R*, such that V_P, the corresponding vector of the new root node *P*. *P* and V_P have the following properties:

$$V_P = V_L + V_R \quad (11) \quad , \quad c_P = \frac{\|V_L\| * \|V_R\|}{\|V_L\| + \|V_R\|} \quad (12)$$

where V_L, V_R and V_P are vectors corresponding respectively to the root nodes *L*, *R*, and *P*, while $\|V_L\|$ and $\|V_R\|$ are modules of V_L and V_R; and c_P is the co-similarity of *P*. We assign a "length" value to each of the two branches connecting *L* and *R* to *P*, as follows:

$$d_{L,P} = \frac{\|V_R\|}{\|V_L\| + \|V_R\|} \quad (13) \quad , \quad d_{R,P} = \frac{\|V_L\|}{\|V_L\| + \|V_R\|} \quad (14)$$

These values are the estimate of the phylogenetic distance[1] from either node L or R to their parent P in the tree.

5.2.3 Step 3: Separating nodes

The CLUSS algorithm makes use of a systematic method for deciding which subtrees to retain as a trade-off between searching for the highest co-similarity values and searching for the largest possible clusters. We first separate all the subtrees into two groups, one being the group of high co-similarity subtrees and the other the low co-similarity subtrees. This is done by sorting all possible subtrees in increasing order of co-similarity and computing a separation threshold according to the method based on the maximum interclass inertia [7].

5.3 Stage 3: Extracting clusters

From the group of high co-similarity subtrees, we extract those that are largest. A high co-similarity subtree is largest if the following two conditions are satisfied: 1) it does not contain any low co-similarity subtree; and 2) if it is included in another high co-similarity subtree, the latter contains at least one low co-similarity subtree. Each of these (largest) subtrees corresponds to a cluster and its leaves are then collected to form the corresponding cluster.

6. RESULTS

To illustrate its efficiency, we tested CLUSS extensively on a variety of protein datasets and databases and compared its performance with that of some mainstream clustering algorithms. We analyzed the results obtained for the different tests with support from the literature and functional annotations. Full data files and results cited in this section are available on CLUSS website.

6.1 The clustering quality measure

To highlight the functional characteristics and classifications of the clustered families, we introduce the *Q-measure* which quantifies the quality of a clustering by measuring the percentage of correctly clustered protein sequences based on their known functional annotations. This measure can be easily adapted to any protein sequence database. The *Q-measure* is defined as follows:

$$Q\text{-}measure = \frac{\left(\sum_{i=1}^{C} P_i\right) - U}{N} \qquad (15)$$

where N is the total number of clustered sequences, C is the number of clusters obtained, P_i is the largest number of obtained sequences in the i^{th} cluster belonging to the same function group according to the known reference classification, and U is the number of orphan sequences. For the extreme case where each cluster contains one protein with all proteins classified as such, the *Q-measure* is 0, since C becomes equal to N, and each P_i the largest number of obtained sequences in the i^{th} cluster is 1.

6.2 COG and KOG databases

To illustrate the efficiency of CLUSS in grouping protein sequences according to their functional annotation and biological classification, we performed extensive tests on the phylogenetic classification of proteins encoded in complete genomes, commonly named the Clusters of Orthologous Groups of proteins database [28]. As mentioned in the web site for the database, the COG (for unicellular organisms) and KOG (for eukaryotic organisms) clusters were delineated by comparing protein sequences encoded in complete genomes, representing major phylogenetic lineages. Each COG and KOG consists of individual proteins or groups of paralogs from at least 3 lineages and each thus corresponds to an ancient conserved domain. COG and KOG contain (to date) 192,987 and 112,920 classified protein sequences, respectively.

To perform a biological and statistical evaluation of CLUSS, we randomly generated two sets of 1000 large subsets, one from the COG database and the other from the KOG database. Each subset contains between 47 and 1840 non-orphan protein sequences (i.e., each selected protein sequence has at least one similar from the same functional classification) from at least 10 distinct groups in the COG or KOG classification. We tested CLUSS on both sets of 1000 subsets using each of the substitution matrices BLOSUM62 [14] and PAM250 [6]. The average *Q-measure* value of the clusterings obtained for the COG classification is superior to **88%** with a standard deviation of **5.61%**, and the value for the KOG classification is superior to **80%** with a standard deviation of **9.50%**. The results obtained show clearly that CLUSS is indeed effective in grouping sequences according to the known functional classification of COG and KOG databases.

In the aim of comparing the efficiency of CLUSS to that of alignment-dependent clustering algorithms, we performed tests using CLUSS, BlastClust [3], TRIBE-MCL [10] and gSPC [34] on the COG and KOG classifications. In all of the tests performed, we used the widely known protein sequence comparison algorithm ClustalW [36] to calculate the similarity matrices used by TRIBE-MCL [10] and gSPC [34]. Due to the complexity of alignment, these tests were done on two sets of six randomly generated subsets, named COG1 to COG6 for COG and KOG1 to KOG6 for KOG. The obtained results are summarized in Table 1.

The results in Table 1 show clearly that CLUSS obtained the best *Q-measure* compared to the other algorithms tested. Globally, the clusters obtained using our new algorithm CLUSS correspond better to the known characteristics of the biochemical activities and modular structures of the protein sequences according to COG and KOG classifications.

The execution time reported in Table 1 for algorithm comparison, show clearly that the fastest algorithm is BlastClust [3], closely followed by our algorithm CLUSS, while TRIBE-MCL [10] and gSPC [34], which use ClustalW [36] as similarity measures, are much slower than BlastClust [3].

6.3 Glycoside Hydrolase family 2 (GH2)

To show the performances of CLUSS with multi-domain protein families which are known to be hard-to-align and have not yet been definitively aligned, experimental tests were performed on 316 proteins belonging to the Glycoside Hydrolases family 2 (FASTA file is provided at CLUSS website) from the CAZy

[1] This distance has no strict mathematical sense; it is merely a measure of the evolutionary distance between the nodes. It is closer to the notion of dissimilarity.

Table 1. *Q-measure* (Q-m) and execution time (in seconds) obtained on each COG and KOG subset.

Protein sets and number of sequences	**CLUSS+SMS**		**BlastClust**		**MCL+Clustal**		**SPC+Clustal**	
	Q-m	**Time**	**Q-m**	**Time**	**Q-m**	**Time**	**Q-m**	**Time**
COG1 (336)	**96.73**	116	81.25	**10**	92.26	332	93.45	340
COG2 (214)	**95.33**	49	84.22	**7**	88.78	141	93.92	146
COG3 (215)	**93.06**	74	87.50	**14**	83.68	273	73.26	285
COG4 (355)	**90.42**	86	82.81	**12**	78.59	315	79.71	324
COG5 (667)	**98,08**	667	94.00	**105**	63.46	5393	70.01	5338
COG6 (309)	**95.15**	68	88.02	**18**	87.70	224	88.99	239
KOG1 (363)	**96.14**	414	67.21	**44**	69.69	1168	76.85	1209
KOG2 (425)	**90.12**	289	31.01	**27**	68.70	1208	53.64	1230
KOG3 (411)	**93.92**	258	42.33	**55**	74.85	270	75.91	325
KOG4 (360)	**93.06**	361	38.88	**127**	66.66	1123	67.22	1220
KOG5 (326)	**97.24**	221	77.91	**33**	75.46	688	82.51	718
KOG6 (590)	**90,68**	779	50.33	**405**	85.25	3782	66.94	4181

database [35]. The CAZy database describes the families of structurally-related catalytic and carbohydrate-binding modules or functional domains of enzymes that degrade, modify, or create glycosidic bonds. Among proteins included in CAZy database, the Glycoside Hydrolases are a widespread group of enzymes which hydrolyse the glycosidic bond between two or more carbohydrates or between a carbohydrate and a non-carbohydrate moiety. Among Glycoside Hydrolases families, the GH2 family, extensively studied at the biochemical level includes enzymes that perform five distinct hydrolytic reactions. Only complete protein sequences were retained for this study. In our experimentation, the GH2 proteins were subdivided into 28 subfamilies, organized in four main branches. Three branches correspond perfectly to enzymes with known biochemical activities. The first branch (subfamilies 1–7) includes enzymes with “*β-galactosidase*” activity from both Prokaryotes and Eukaryotes. The third branch (subfamilies 18 to 22) groups enzymes with “*β-mannosidase*” activity, while the fourth branch (subfamilies 23 to 28) includes “*β-glucuronidases*”.

The clustering scheme obtained warrants further comment. The “orphan” subfamily 17 includes nineteen sequences labelled as “*β-galactosidases*” in databases. While the branch 1 “*β-galactosidases*” are composed of five modules, known as the “*sugar binding domain*”, the “*immunoglobulin-like β-sandwich*”, the “*(αβ)8-barrel*”, the “*β-gal small_N domain*” and the “*β-gal small_C domain*”, the members of subfamily 17 lack the last two of these domains, which makes them more similar to “*β-mannosidases*” and “*β-glucuronidases*”. These enzymes are distinct from those of branch 1 [11] and their separate localization is justified.

The second branch is the most heterogeneous in terms of enzyme activity. However, most of the subfamilies (9 to 16) group enzymes that are annotated as “*putative β-galactosidases*” in databases. To the best of our knowledge, none of these proteins, identified through genome sequencing projects, have been characterized by biochemical techniques, so their enzymatic activity remains hypothetical. At the beginning of this branch, subfamily 8 groups enzymes characterized very recently: “*exo-β-glucosaminidases*” [5,16] and “*endo-β-mannosidases*” [17]. Again, theses enzymes share only three modules with the enzymes from branches 1, 3 and 4. The close proximity among “*exo-β-glucosaminidases*” and “*endo-β-mannosidases*” emerging from this work has not been described so far. Furthermore, subfamily 8 includes closely related plant enzymes with “*endo-β-mannosidase*” activity and bacterial enzymes produced by members of the genus *Xanthomonas*, including several plant pathogens. This could be an example of horizontal genetic transfer between members of these two taxa.

Subfamily 22, also found at the beginning of a branch, has been recently analyzed by Côté *et al.* [5] and Fukamizo *et al.* [13], using structure-based sequence alignments and biochemical structure-function studies. It was shown that proteins from this subfamily have a different catalytic doublet and could recognize a new substrate not yet associated with GH2 members.

Globally, the clustering result for the GH2 proteins corresponds well to the known characteristics of their biochemical activities and modular structures. The results obtained with the CLUSS algorithm were highly comparable with those of the more complex analysis performed by Côté *et al.* [5] and Fukamizo *et al.* [13] using clustering based on structure-guided alignments, an approach which necessitates prior knowledge of at least one 3D protein structure.

6.4 Group of 33 (α/β)8-barrel proteins

To show the performance of CLUSS with multi-domain protein families which are known to be hard to align and have not yet been definitively aligned, experimental tests were performed on the group of the 33 (α/β)$_8$-barrel proteins, a group within Glycoside Hydrolases family 2 (GH2), from the CAZy database [35], studied recently by Côté *et al.* [5] and Fukamizo *et al.* [13]. The periodic character of the catalytic module known as “(α/β)$_8$-barrel” makes these sequences hard to align using classical alignment approaches. The difficulties in aligning these modules are comparable to the problems encountered with the alignment of tandem-repeats, which have been exhaustively discussed [15]. The FASTA file and clustering results of this subfamily are available on the CLUSS website. This group of 33 protein sequences includes “*β-galactosidase*”, “*β-mannosidase*”, “*β-glucuronidase*” and “*exo-β-D-glucosaminidase*” enzymatic activities, all extensively studied at the biochemical level. These sequences are multi-modular, with various types of modules, which complicate their alignment. Clustering such protein sequences using the alignment-dependent algorithms thus becomes problematic. In our experiments, we tested quite a few known algorithms to align the 33 protein sequences, such as MUSCLE [8], ClustalW [36], MAFFT [18], T-Coffee [26] etc. The alignment results of all these algorithms are in contradiction with those presented by Côté *et al.* [5] which in turn are supported by the structure-function studies of Fukamizo *et al.* [13]. This encouraged us to perform a clustering on this subfamily, to compare the behaviour of CLUSS with BlastClust [3], TRIBE-MCL [10] and gSPC [34] in order to validate the use of CLUSS on the hard-to-align proteins. The experimental results with the different algorithms are summarized in Table 2, which shows the cluster correspondence of each of the sequences by approach used. An overview of the results is given below. The corresponding names and database entries of the 33 $(\alpha/\beta)_8$-barrel proteins group are indicated at CLUSS website.

6.4.1 CLUSS results

The 33 $(\alpha/\beta)_8$-barrel proteins were subdivided by CLUSS into five subfamilies, organized in five main branches (details in Figure 2). The first and the second branch correspond, respectively, to the first and the second clusters, which include enzymes with "*β-mannosidase*" activities; the third branch corresponds to the third cluster, which includes enzymes with "*β-glucuronidase*" activities; the fourth branch corresponds to the forth cluster, which includes enzymes with "*β-galactosidase*" activities; the fifth branch corresponds to the fifth cluster, which includes enzymes with "*exo-β-D-glucosaminidase*" activities.

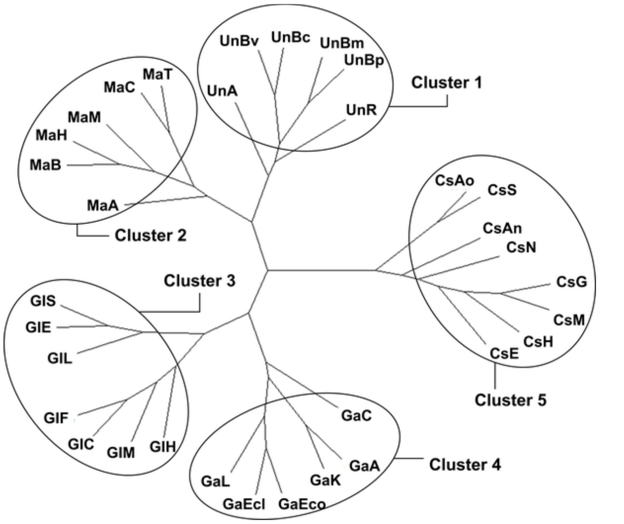


Figure 2. Phylogenetic analysis of 33 $(\alpha/\beta)_8$-barrel group

6.4.2 BLAST results

The 33 $(\alpha/\beta)_8$-barrel proteins were subdivided into five subfamilies. Almost all the enzymes were clustered in the appropriate clusters, except for seven proteins that were unclustered, among which we find the following well-classified enzymes: the "*β-galactosidase*" enzymes GaA, GaK and GaC; the "*β-mannosidase*" enzyme UnBc; and the "*exo-β-D-glucosaminidase*" enzyme CsAo.

Table 2. Clustering results on 33 (α/β)8-barrel group

Protein set	Côté *& al.*	CLUSS	Blast	MCL	SPC
UnA	**1**	**1**	1	1	1
UnBv	**1**	**1**	1	1	1
UnBc	**1**	**1**	/	1	1
UnBm	**1**	**1**	1	1	1
UnBp	**1**	**1**	1	1	1
UnR	**1**	**1**	1	1	1
MaA	**2**	**2**	2	2	1
MaB	**2**	**2**	2	1	1
MaH	**2**	**2**	2	1	1
MaM	**2**	**2**	2	1	1
MaC	**2**	**2**	2	2	1
MaT	**2**	**2**	2	2	1
GIC	**3**	**3**	3	2	2
GIE	**3**	**3**	3	2	2
GIH	**3**	**3**	3	2	2
GIL	**3**	**3**	3	2	2
GIM	**3**	**3**	3	2	2
GIF	**3**	**3**	3	2	2
GIS	**3**	**3**	3	2	2
GaEco	**4**	**4**	4	2	2
GaA	**4**	**4**	/	2	2
GaK	**4**	**4**	/	2	2
GaC	**4**	**4**	/	2	2
GaEcl	**4**	**4**	4	2	2
GaL	**4**	**4**	4	2	2
CsAo	**5**	**5**	/	2	3
CsS	**5**	**5**	5	2	3
CsG	**5**	**5**	5	2	3
CsM	**5**	**5**	5	2	3
CsN	**5**	**5**	/	2	3
CsAn	**5**	**5**	/	2	3
CsH	**5**	**5**	5	2	3
CsE	**5**	**5**	5	2	3

6.4.3 Tribe-MCL results

The 33 $(\alpha/\beta)_8$-barrel proteins were subdivided by TRIBE-MCL into two mixed subfamilies. We find the "*β-mannosidase*" enzymes MaA, MaC and MaT grouped in the "*β-galactosidase*" subfamily. Furthermore, the "*exo-β-D-glucosaminidase*" and "*β-glucuronidases*" enzymes are grouped in the same subfamily.

6.4.4 gSPC results

The 33 $(\alpha/\beta)_8$-barrel proteins were subdivided by gSPC into three subfamilies. Almost all the enzymes were grouped in the appropriate subfamily, except for the "*β-galactosidases*" and the "*β-glucuronidases*" which were grouped in the same subfamily.

Globally, the clustering of the 33 $(\alpha/\beta)_8$-barrel proteins generated by CLUSS corresponds better to the known characteristics of their biochemical activities and modular structures than do those yielded by the other algorithms tested. The results obtained with our new algorithm were highly comparable with those of the more complex, structure-based analysis performed by Côté *et al.* [5] and Fukamizo *et al.* [13].

7. DISCUSSION

The new similarity measure presented in this paper makes possible to measure the similarity between protein sequences based solely on the conserved motifs. Its major advantage compared to the alignment-dependent approaches is that it gives significant results with protein sequences independent of their alignability, which allows it to be effective on both easy-to-align and hard-to-align protein families. This property is inherited by CLUSS, our new clustering algorithm, which uses it as its similarity measure. CLUSS used jointly with SMS is an effective clustering algorithm for protein sets with a restricted number of functions, which is the case of almost all protein families. It more accurately highlights the characteristics of the biochemical activities and modular structures of the clustered protein sequences than do the alignment-dependent algorithms.

Our new clustering algorithm CLUSS gains several advantages by adopting an approach inspired by latent semantic analysis (LSA). The first is its use of high-dimensional space to automate the encoding and comparison of semantic relations. The second is its use of spectral decomposition, thereby benefiting from the global nature of this approach [27], since the Eigen decomposition used depends essentially on the globality of the similarity matrix *SMS*, and a change in one value in *SMS* makes changes in the entire Eigen decomposition.

So far, our similarity measure has been based on pre-determined substitution matrices. A possible future development is to propose an approach to automatically compute the weights of the conserved motifs instead of relying on pre-calculated substitution scores. There is also a need to speed up the extraction of the conserved motifs and the clustering of the phylogenetic tree, to scale the algorithm on datasets that are much larger in size with many more biological functions.

We believe that CLUSS is an effective method and tool for clustering protein sequences to meet the needs of biologists in terms of phylogenetic analysis and function prediction. In fact, CLUSS gives an efficient evolutionary representation of the phylogenetic relationships between protein sequences. This algorithm constitutes a significant new tool for the study of protein families, the annotation of newly sequenced genomes and the prediction of protein functions, especially for proteins with multi-domain structures whose alignment is not definitively established. Finally, the tool can also be easily adapted to cluster other types of genomic data.

8. REFERENCES

[1] S. F. Altschul, W. Gish, W. Miller, E. W. Myers, D. J. Lipman. Basic local alignment search tool. *J. Mol. Bio.* 1990, 215:403–410.

[2] S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, D. J. Lipman. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucl. Acids Res.* 1997, 25:3389–3402.

[3] Basic Local Alignment www.ncbi.nlm.nih.gov/BLAST.

[4] M. W. Berry, R. D. Fierro. Low-rank orthogonal decomposition for information retrieval applications. *Numerical Linear Algebra Applications, Vol. 1*, 1996, 1-27.

[5] N. Côté, A. Fleury, E. Dumont-Blanchette, T. Fukamizo, M. Mitsutomi, R. Brzezinski. Two exo-β-D-glucosaminidases / exochitosanases from actinomycetes define a new subfamily within family 2 of glycoside hydrolases. *Biochem. J.* 2006, 394:675–686.

[6] M. O. Dayhoff, R. M. Schwartz, B. C. Orcutt. A model of evolutionary change in proteins. *Atlas of Protein Sequence and Structure vol. 5* 1978, suppl. 3:345-352.

[7] R. O. Duda, P. E. Hart, D. G. Stork. *Pattern Classification*, second edition, John Wiley and Sons, 2001.

[8] R. C. Edgar. MUSCLE: A multiple sequence alignment method with reduced time and space complexity. *BMC Bioinformatics* 2004, 5:113.

[9] R. C. Edgar. Local homology recognition and distance measures in linear time using compressed amino acid alphabets. *Nucl. Acids Res.* 2004, 32:380-385.

[10] A. J. Enright, S. Van Dongen, C. A. Ouzounis. An efficient algorithm for large-scale detection of protein families. *Nucl. Acids Res*. 2002, 30:1575-1584.

[11] S. Fanning, M. Leahy, D. Sheehan. Nucleotide and deduced amino acid sequences of Rhizobium meliloti 102F34 lacZ gene: Comparison with prokaryotic beta-galactosidases and human beta- glucuronidase. *Gene* 1994, 141:91-96.

[12] J. Felsenstein. An alternating least squares approach to inferring phylogenies from pairwise distances. *Syst. Biol.* 1997, 46:101.

[13] T. Fukamizo, A. Fleury, N. Côté, M. Mitsutomi, R. Brzezinski. Exo-β-D-glucosaminidase from Amycolatopsis orientalis: Catalytic residues, sugar recognition specificity, kinetics, and synergism. *Glycobiology* 2006, 16:1064-1072.

[14] S. Henikoff, J. G. Henikoff. Amino acid substitution matrices from protein blocks. *Proceedings of the National Academy of Sciences of the United States of America* 1992, 89:10915-10919.

[15] D. Higgins. Multiple alignment. In *The Phylogenetic Handbook: A Practical Approach to DNA and Protein Phylogeny*. Edited by Salemi M, Vandamme A.M. Cambridge University Press, 2004:45-71.

[16] M. Ike, K. Isami, Y. Tanabe, M. Nogawa, W. Ogasawara, H. Okada, Y. Morikawa. Cloning and heterologous expression of the exo-β-D-glucosaminidase-encoding gene (gls93) from a filamentous fungus, Trichoderma reesei PC-3-7. *Appl. Microbiol. Biotechnol.* 2006, 72: 687–695.

[17] T. Ishimizu, A. Sasaki, S. Okutani, M. Maeda, M. Yamagishi, S. Hase. Endo-beta-mannosidase, a plant enzyme acting on N-glycan: Purification, molecular cloning and characterization. *J. Biol. Chem.* 2004, 279:3855-3862.

[18] K. Katoh, K. Misawa, K. Kuma, T. Miyata. MAFFT: A novel method for rapid multiple sequence alignment based on fast Fourier transform. *Nucl. Acids Res.* 2002, 30:3059-3066.

[19] S. Karlin, G. Ghandour. Comparative statistics for DNA and protein sequences: Single sequence analysis. *Proc. Natl. Acad. Sci. USA* 1985, 82:5800-5804.

[20] S. Karlin, G. Ghandour. Comparative statistics for DNA and protein sequences: Multiple sequence analysis. *Proc. Natl. Acad. Sci. USA* 1985, 82:6186-6190.

[21] S. Karlin, F. Ost. Maximal length of common words among random letter sequences. *The Annals of Probability* 1988, 16:535-563.

[22] K. Kimura. Evolutionary rate at the molecular level. *Nature*, 1968 217:624–626.

[23] A. Krause, J. Stoye, M. Vingron. The SYSTERS protein sequence cluster set. *Nucl. Acids Res.* 2000:28:270–272.

[24] H. Lodish, A. Berk, P. Matsudaira, C. A. Kaiser, M. Krieger, M. P. Scott, L. Zipursky, J. Darnell. *Molecular Cell Biology*, 5th ed. New York and Basingstoke: W.H. Freeman and Co., 2004.

[25] D. W. Mount. *Bioinformatics. Sequence and Genome Analysis (2nd ed.)*, Cold Spring Harbor Laboratory Press, New York, 2004.

[26] C. Notredame, D. Higgins, J. Heringa. T-Coffee: A novel method for multiple sequence alignments. *Journal of Molecular Biology* 2000, 302:205-217.

[27] A. Paccanaro, J. A. Casbon, M. A. S. Saqi. Spectral clustering of protein sequences. *Nucleic Acids Research*. 2006, Vol. 34, No. 5 1571–1580.

[28] Phylogenetic classification of proteins encoded in complete genomes: www.ncbi.nlm.nih.gov/COG.

[29] P. Pipenbacher, A. Schliep, S. Schneckener, A. Schonhuth, D. Schomburg, R. Schrader. ProClust. Improved clustering of protein sequences with an extended graph-based approach. *Bioinformatics* 2002, 18:S182–S191.

[30] G. Reinert, S. Schbath, M. S. Waterman. Probabilistic and statistical properties of words: An overview. *J. Comp. Biol.* 2000, 7:1-46.

[31] J. Rocha, F. Rossello, J. Segura. The Universal Similarity Metric does not detect domain similarity. *Q-bio.QM* 2006, 1:0603007.

[32] K. Sjölander. Phylogenomic inference of protein molecular function: Advances and challenges. *Bioinformatics* 2004, 20:170-179.

[33] E. L. L. Sonnhammer, V. Hollich. Scoredist: A simple and robust sequence distance estimator. *BMC Bioinformatics* 2005, 6:108.

[34] I. V. Tetko, A. Facius, A. Ruepp, H. W. Mewes. Super paramagnetic clustering of protein sequences. *BMC Bioinformatics* 2005, 6:82.

[35] The carbohydrate-active enzymes (CAZy) database: afmb.cnrs-mrs.fr/CAZY.

[36] J. D. Thompson, D. G. Higgins, T. J. Gibson. CLUSTAL W: Improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position-specific gap penalties and weight matrix choice. *Nucl. Acids Res.* 1994, 22:4673-4680.

[37] J. S. Varré, J. P. Delahaye, R. Rivals. The transformation distance: A dissimilarity measure based on movements of segments. *Bioinformatics* 1999, 15:194–202.

[38] S. Vinga, J. Almeida. Alignment-free sequence comparison – A review. *Bioinformatics* 2003, 19:513-523.

[39] H. P. William, A. T. Saul, T. V. William, P. F. Brian. *Numerical recipes in C (2nd ed.): The art of scientific computing*, Cambridge University Press, New York, NY, 1992.

[40] G. Yona, N. Linial, M. Linial. ProtoMap: Automatic classification of protein sequences and hierarchy of protein families. *Nucl. Acids Res.* 2000, 28:49-55.

Discovering Ovarian Cancer Biomarkers using Gene Ontology Based Microarray Analysis

Wei Guan[*]
College of Computing
Georgia Institute of Technology
Atlanta, Georgia
wguan@cc.gatech.edu

Alexander Gray
College of Computing
Georgia Institute of Technology
Atlanta, Georgia
agray@cc.gatech.edu

Sham Navathe
College of Computing
Georgia Institute of Technology
Atlanta, Georgia
sham@cc.gatech.edu

Nathan Bowen
Department of Biology
Georgia Institute of Technology
Atlanta, Georgia
bowen@gatech.edu

John McDonald
Department of Biology
Georgia Institute of Technology
Atlanta, Georgia
john.mcdonald@biology.gatech.edu

Lilya Matyunina
Department of Biology
Georgia Institute of Technology
Atlanta, Georgia
lilya.matyunina@biology.gatech.edu

ABSTRACT

The advent of microarray data has opened new doorways for biological discovery. However, over the years, not all of the hoped-for possibilities have been realized, due to fundamental limitations of microarray data. In this paper, we present a method for augmenting microarray analysis with gene ontology data to provide insight into possible biomarkers (critical genes) for ovarian cancer pathogenesis which is not possible using microarray expression data alone. Using expression data for 12558 genes in 43 patients with both benign and malignant epithelial ovarian tumors, we apply representative state-of-the-art methods for microarray biomarker analysis including support vector machines, five data normalization methods, five feature selection methods, and two dimensionality reduction methods. Our findings showed that for this data: 1) Guanine Cytosine Robust Multi-array Average (GCRMA) appears to outperform other normalization methods, 2) the classification problem alone is not constraining enough to yield unique biomarkers with high confidence. Our new method combining statistical microarray analysis with ontological information is capable of finding putative biomarkers whose expression values are not significantly different between patient groups, but instead may be mutated or regulated at the post-translational level. For example, our method was capable of recovering the known importance of the TUMOR PROTEIN 53 (TP53) in the etiology of epithelial ovarian cancer (EOC) from expression data in which TP53 was not found to be differentially expressed.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Applications of data mining (biomedicine, business, e-commerce, defense)

General Terms

Biomarkers Discovery

Keywords

biomarker, microarray, ovarian cancer, normalization, LOOCV, SVMRFE, Gene Ontology

1. INTRODUCTION

Our dataset consists of microarray data (Affymetrix U95Av2) from 43 ovarian cancer patients [1] [34]. Of the 43 ovarian cancer patients: 10 are benign cancer patients; 9 are malignant cancer patients with no chemotherapy treatment; 24 are malignant cancer patients with chemotherapy treatment. The gene expression dataset from this ovarian cancer patient microarray is of size $43 \times 12,558$ [2], which is high dimension low sample size data. This is a typical microarray dataset from which biologists have to extract meaningful information about genes and is hence hard to analyze. We began the project by doing a thorough statistical microarray analysis applying state-of-the-art methods on this unique dataset which has not been previously intensively studied. Our findings showed that for this data 1) Guanine Cytosine Robust Multi-array Average (GCRMA) appears to outperform other normalization methods, and 2) the classification problem alone is not constraining enough

[*] Correspondence Author

BIOKDD '07, August 12, 2007, San Jose, California, USA
Copyright 2007 ACM 978-1-59593-839-8/07/0008 ...$5.00.

[1] This dataset is provided by Professor McDonald's lab at Dept. of Biology, Georgia Institute of Technology

[2] Of the total 12,625 probes, 67 are Affymetrix reference probes, after microarray normalization, the expression value they measured are discarded from further analysis

to yield unique biomarkers with high confidence. We were led to the method we present at the end because the traditional microarray-only analysis seems insufficient. Our new method combining statistical microarray analysis with ontological information is capable of finding putative biomarkers whose expression values are not significantly different between patient groups, but instead may be mutated or regulated at the post-translational level.

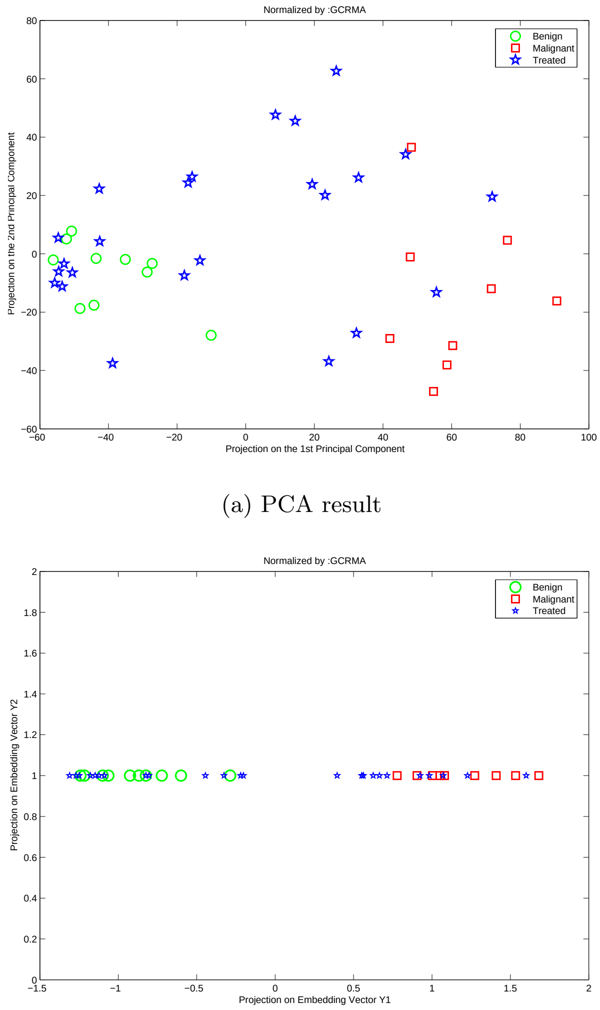
This microarray dataset is a high density oligonucleotide microarray data, generated using Affymetrix U95Av2 GeneChip. In this type of micorarray experiments, oligonucleotide sequences of length 25 base pairs are used to probe genes. There are two types of probes: perfect match (PM) reference probes which match a target sequence exactly; and mismatch (MM) partner probes which differ from perfect matches only by a single base in the center of the sequence. Typically 16-20 of probe pairs (PM+MM) interrogate different parts of a target gene sequence and are referred as a *probeset*. Gene expression value of a probeset is composed by the intensity information of each probe in the probeset [6].

This paper is organized as follows: Section 2 gives our quantitative comparison on the five commonly used oligonucleotide microarray normalization methods. Section 3 summarizes the standard techniques in biomarker discovery, and shows why the microarray-only methods are insufficient in biomarker discovery on our microarray data. Section 4 presents our novel gene selection method which incorporates the gene ontology information extracted from Affymetrix annotation files into the microarray data analysis. Finally, section 5 concludes this work and discusses the future directions.

2. NORMALIZATION ANALYSIS

Microarray *normalization* adjusts individual intensities to remove differences that are purely technical and do not represent true biological variation. Examples of such differences are difference in probe labelling (affinity to target genes, amounts of sample and label used), heat and light sensitivities, systematic biases in measured expression levels, scanner settings, print-tip variation and sample plate origin [2]. Determining an appropriate normalization method of a microarray dataset is thus a critical step which influences the rest of the microarray analysis, so our goal is to obtain the microarray gene expression data in its best possible normalized form. We found that overall, Guanine Cytosine Robust Multi-array Average (GCRMA) [36] appears preferable to the other methods for our microarray data. For researchers working on microarray data, there is still no consensus regarding the best normalization method. Hence the fact that one of the normalization methods is better than the others for our dataset is interesting in its own right. This section thus can be skipped for those not interested in the details and it is not directly related to the main result of our paper.

For oligonucleotide microarray data, there are five commonly used data normalization methods that preprocess the raw microarray data into gene expression data matrices: Affymetrix - Microarray Suite (MAS) [17], Model Based Expression Index (MBEI) [21], Probe Logarithmic Intensity Error Estimation (PLIER) [1], Robust Multiple-chip Analysis (RMA) [18, 6] and GCRMA [36]. These methods basically differ in error model, probe information for estimation and background adjustment method being used [1]. We use their implementation in Bioconductor [3] (part of the R statistical package).



(a) PCA result

(b) LLE result

Figure 1: 2D Projection on the Ovarian Cancer Microarray Data through Dimension Reduction

We extend the idea from [35], in which the authors compare ten cDNA microarray normalization methods according to the Leave-One-Out-Cross-Validation (LOOCV) classification accuracy using K nearest neighbor (kNN) classifier. As shown in the dimension reduction results (Figure 1) [4] through Principal Component Analysis (PCA) [20], and Local Linear Embedding (LLE) [26, 27], this high dimensional microarray dataset (12558 genes) is linear separable. We thus compare the performance of oligonucleotide microarray normalization methods by evaluating SVM (linear kernel) [7] LOOCV classification accuracy through the *Support Vector Machine Recursive Feature Elimination* (SVMRFE)

[3] Bioconductor: http://www.bioconductor.org/download

[4] Dimension reduction on microarray data normalized by other normalization methods (RMA, PLIER, MAS, MEBI) generates similar linear separable projection

process. The assumption is that the gene expression data obtained from better normalization method should have better discrimination among different groups of the patients, and thus the expression data of genes selected by SVMRFE algorithm at each iteration should also be more discriminative among different groups of patients.

Given a gene expression dataset (data matrix of patients by gene expression values), the SVM LOOCV classification accuracy is calculated as follows: for each patient, we take the corresponding gene expression value data out, and build an SVM classifier using the gene expression value data of all the other patients in the dataset, and then use the classifier to classify the label of the patient taken-out, we repeat the above procedure for all the patients and count how many patients have been classified correctly.

The evaluation procedure on SVM LOOCV performance through SVMRFE process is described in Table 1. For each normalization method, we first obtain the gene expression value dataset from our ovarian cancer microarray data preprocessed using the normalization method. Then, at each iteration of the SVMRFE process: we use the LOOCV classification accuracy with SVM classifier to measure the discriminative capability of the current gene expression value dataset X(N,D). Next, we apply the SVMRFE gene selection method on gene expression data to remove non-discriminative genes and thus select out gene expression dataset for the next iteration.

Table 1: Evaluation procedure of SVM LOOCV through SVMRFE process

```
For each normalization method $M_a$:
    Obtain corresponding expression dataset $X$;
    Repeat
        LOOCV on current expression dataset $X(N, D)$
        Build SVM classifier on the current dataset
        Rank gene $j$ according to $score(j) =| w_j |$
        Remove the bottom 10% genes
        Obtain new dataset $X_{new}(N, D_{new})$ , $D_{new} = 0.9D$
            $D = D_{new}$; $X = X_{new}$
    Until $D < 1$
end;
```

Figure 2 shows the comparison result of the SVM LOOCV classification accuracy of the gene expression profiles of the 19 cancer patients without treatment, i.e. the training data consists of the 10 benign cancer patients and 9 malignant cancer patients. The x axis gives the logarithm of the number of genes using in the SVM LOOCV classification accuracy calculation. The y axis gives the SVM LOOCV classification accuracy.

Figure 3 displays the comparison result of the SVM LOOCV classification accuracy on the expression values of the 24 malignant cancer patients with treatment. From the dimension reduction results on the microarray dataset (Figure 1), we can claim that our ovarian cancer microarray dataset is linearly separable, and the treated malignant cancer patients can be classified into benign-like or malignant-like classes. Therefore, we built an SVM classifier using the gene expression profiles of the 19 cancer patients without treatment, and used the classifier to determine the class of the gene expression profiles of the 24 malignant cancer patients with chemotherapy treatment. The classification result as used as the basis of the experiment, i.e. the training data consists of the 13 treated cancer patients whose expression values are classified as benign like, and the 11 treated cancer patients whose expression values are classified as malignant like.

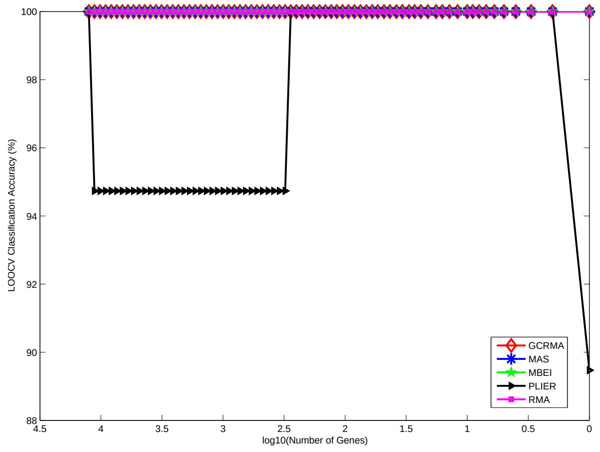


Figure 2: SVMRFE LOOCV Results of non-treated patients (benign 10: malignant 9)

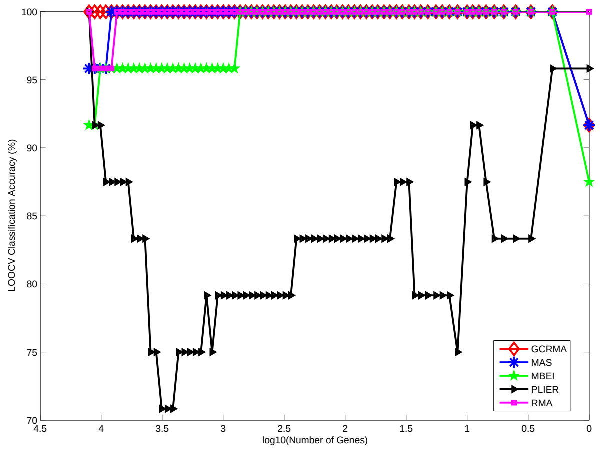


Figure 3: SVMRFE LOOCV Results of treated patients (benign-like 13: malign-like11)

As we can see from the resulting plots, gene expression profiles obtained from PLIER normalization method have the worst discriminative capability among different patient groups, and so does the gene set selected using PLIER. Gene expression profiles obtained from MAS and MEBI normalization methods are better, but still worse comparing to GCRMA and RMA. Gene expression profiles obtained from normalization method GCRMA are very stable over the experiments in treated patients and non-treated patient case except only for the case of top gene (i.e. gene number = 1) classification on treated cancer patients data. While gene expression profiles obtained from RMA normalization method are very stable through SVMRFE process when the size of the selected gene set becomes small. Therefore, the gene expression data normalized by GCRMA on the microarray data will be used in the gene selection experiments in following sections.

3. CLASSIFICATION-BASED ANALYSIS

Our explorative analysis on this microarray data shows that statistical microarray-only analysis does not appear capable of identifying unique biomarkers with high confidence. This section first summarizes the traditional biomarker discovery methods which are mainly gene-expression-only analysis. Next, four biomarker discovery methods are applied to select out putative biomarkers from the dataset. The gene selection results show that the traditional classification-based analysis may be insufficient to identify biological meaningful biomarkers in our microarray data.

3.1 Biomarker Discovery Methods

Biomarker discovery, i.e. gene selection methods are basically derived from feature selection methods used in text categorization and other scientific applications. The results of gene selection, i.e. putative biomarkers are usually evaluated by their discriminative capability among different sample classes. There are mainly two types of gene selection methods: *filter-based* approach and *wrapper-based* approach.

Filter-based approach uses statistical information between genes (and classes), including: information gain, symmetrical uncertainty, t-statistics, gini index, χ^2 statistics, Signal to Noise (S2N) ratio [13], RBF (Redundancy based filter) algorithm [28], CFS (Correlation Feature selection) Criteria [33], etc.

A micorarray experiment is a good example of multiple hypothesis testing, in which thousand of genes are measured simultaneously against the null hypothesis that gene j is not differentially expressed among two sample groups. Thus gene selection is reduced to finding genes that reject the null hypothesis. Therefore, false positive error control methods can be applied to correct the raw two-sided p value from Welch t-test on each gene. Putative biomarkers are those genes with small p values after correction. The commonly used methods of this category are: Bonferroni correction [12]; Holms step-down adjustment of Bonferroni correction [15]; Benjamini and Hochberg false discovery rate (BH-FDR) correction [4]; and Benjamini and Yekutieli false discovery rate (BY-FDR) correction [5].

Wrapper-based approaches determine the importance of gene(s) according to the discriminative capability over different classes of samples. One method is to build one dimensional support vector machine [29] for each gene, and then rank the genes according to their classification performance among different sample groups. Another method is to select genes according to their projection on the first principal component by performing dimension reduction technique PCA (principal component analysis) [20] on the dataset.

The most widely used method of this approach is Support Vector Machine Recursive Feature Elimination method (SVMRFE) [14, 16]. This method will iteratively repeat the following process until the desired number of remaining genes is reached: i) train a linear-kernel SVM using all the remaining genes; ii) sort the genes by $score(j) =| w_j |$, w is the slope of the discrimination hyperplane of the SVM classifier; iii) remove 10% genes with lowest $s(j)$. Recently, more complex methods like MSVM-RFE (multiple SVM-RFE) [10], PMBGA (probabilistic model building genetic algorithm) method [25], LS-SVM (least squares SVM) [38] method, etc. were proposed for handling gene selection in more complicated datasets.

3.2 Insufficiency of Standard Biomarker Discovery Methods

This subsection presents the gene selection results from four standard biomarker discovery methods: SVMRFE, PCA, 1D-SVM, and hypothesis approach (t-test), and then describes exactly why we think the microarray-only method is insufficient for obtaining reliable biomarkers.

Table 2, 3 lists the top 10 genes selected out using two wrapper-based biomarker discovery methods: SVMRFE method and PCA method, respectively.

Table 2: Top 10 Genes selected using SVMRFE

Symbol	Gene Name
C10orf72	Chromosome 10 open reading frame 72
TNXA /// TNXB	tenascin XA pseudogene /// tenascin XB
LOC388388	Chromodomain helicase DNA binding protein 3
PEG3	paternally expressed 3
TCF21	transcription factor 21
ECM2	extracellular matrix protein 2, female organ and adipocyte specific
UST	uronyl-2-sulfotransferase
CD22 /// MAG	CD22 antigen /// myelin associated glycoprotein
STAR	steroidogenic acute regulator
SERPINE2	serpin peptidase inhibitor, clade E (nexin, plasminogen activator inhibitor type 1), member 2

Table 3: Top 10 Genes selected using PCA

Symbol	Gene Name
TNXA /// TNXB	tenascin XA pseudogene /// tenascin XB
PEG3	paternally expressed 3
SPP1	secreted phosphoprotein 1 (osteopontin, bone sialoprotein I, early T-lymphocyte activation 1)
ECM2	extracellular matrix protein 2, female organ and adipocyte specific
MYH11	myosin, heavy polypeptide 11, smooth muscle
SPP1	secreted phosphoprotein 1 (osteopontin, bone sialoprotein I, early T-lymphocyte activation 1)
C7	complement component 7
STAR	steroidogenic acute regulator
TCF21	transcription factor 21
C10orf72	Chromosome 10 open reading frame 72

Biomarker discovery methods like Hypothesis Testing based methods and 1D-SVM classification method can give the estimation of the number of genes that are significantly differentially expressed over different patient groups, i.e. the number of genes that would be putative biomarkers. Table 4 summarizes the number of statistically significant genes according to the cut off value q under different false positive error control methods of the hypothesis testing approach. Table 5 summarizes the number of discriminative genes according to the classification accuracy of its 1D-SVM classifier built from gene expression data of the 19 non-treated patients. In which, *LOOCV classification accuracy* refers to the SVM LOOCV classification accuracy over the non-treated cancer patients; *Classification accuracy* refers to the classification performance over the 24 treated cancer patients using the classifier built from gene expression profiles of the 19 non-treated cancer patients; and the *Overlapped Genes* column gives the number of genes whose 1D-SVM classifiers have both 100% LOOCV classification accuracy over the non-treated cancer patients, and 100% classification accuracy over the treated cancer patients, and these genes are listed in Table 6. For comparison purpose, the top 17 genes selected out using hypothesis testing based approach are listed in Table 7.

Table 4: Estimation on the number of significant genes - Hypothesis testing approach

q	Raw p value	Bonferroni	Holms	FDR BH	FDR BY
0.01	2247	111	111	1080	362
0.05	3677	191	191	2130	791

Table 5: Estimation on the number of significant genes - 1D-SVM approach

100% LOOCV performance (non-treated)	100% Classification Accuracy (treated)	Overlapped Genes
191	99	17

Table 6: Top 17 genes selected using 1D-SVM

Symbol	Gene Name
ST13	suppression of tumorigenicity 13 (colon carcinoma) (Hsp70 interacting protein)
PDZRN3	PDZ domain containing RING finger 3
PROS1	protein S (alpha)
PPT2 /// EGFL8	palmitoyl-protein thioesterase 2 /// EGF-like-domain, multiple 8
—	Retinoic acid-inducible endogenous retroviral DNA
TCEAL4	transcription elongation factor A (SII)-like 4
RPL23	ribosomal protein L23
WDFY3	WD repeat and FYVE domain containing 3
KIAA0368	KIAA0368
ARHGEF9	Cdc42 guanine nucleotide exchange factor (GEF) 9
C16orf45	chromosome 16 open reading frame 45
PMM1	phosphomannomutase 1
FHL2	four and a half LIM domains 2
DIRAS3	DIRAS family, GTP-binding RAS-like 3
SDC2	syndecan 2 (heparan sulfate proteoglycan 1, cell surface-associated, fibroglycan)
CIRBP	cold inducible RNA binding protein
FOXO1A	forkhead box O1A (rhabdomyosarcoma)

Since this microarray dataset is linearly separable by one dimension as shown in Figure 1, many single genes can discriminate between benign like patients class and malignant like patients class (see Table 4, 5). Therefore, the classification problem on this microarray dataset is too simple, admitting too many possible solutions in such a high-dimensional space for us to be able to pinpoint critical features. Also the different feature selection methods (hypothesis testing approach, One-dimensional-SVM, as well as SVMRFE, PCA) don't give a lot of overlap (see Table 2, 3, 6, 7). Furthermore, the putative biomarkers found by these methods are not biologically compelling, i.e. relevant to ovary cancer pathogenesis. As an indication of this: we evaluate these gene lists using the function annotation tools provided in DAVID (Database of Annotation, Visualization, and Integrated Discovery) [9]. DAVID statistically measures the Gene-Enrichment, i.e. whether the user input gene list is highly associated with certain biological annotations [24]. The results show that the gene lists generated by the traditional biomarker discovery methods, only associated with a few biological pathway annotations with good confidence, i.e. p-value < 0.1, which is computed through a variant of Fish Exact test. Table 8 displays the genes that are annotated by these good-confidence biological pathways. In which, gene list (10 genes each) generated by SVMRFE, PCA only have 3 genes associated with these biological pathways, respectively; gene lists (17 genes each) generated by 1D-SVM and hypothesis testing approach have 7 and 5 genes associated with the high confidence biological pathways, respectively. Gene expression is regulated by transcription, thus differentially transcribed genes can be identified by microarray-only analyses, which rely only on the gene expression values. However, biological molecules may be regulated at other levels, called as post-translational levels: that is, they are phosphorylated, sumoylated, glycosylated, etc. These differences also render proteins active and inactive are not picked up directly by expression analyses.

4. MICROARRAY+ONTOLOGY BIOMARKER DISCOVERY METHOD

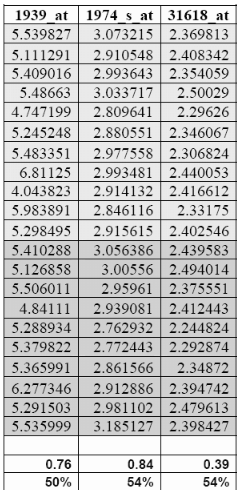
Traditional gene selection methods, like S2N, PCA, SVM-RFE, etc. are able to select out differentially expressed genes from the microarray data. However, some other biologically important genes, like TP53 (tumor protein p53) which is known to play a role in the etiology of many cancers [23], are not differentially expressed across the different biological samples (i.e. different patient classes). These microarray-only gene selection methods only monitor one level of biological regulations, of which there are many. Thus they are not biologically comprehensive, often fail to detect this kind of biomarkers. Therefore, it is necessary to integrate expression data analyses in an easy and automatic way with other levels of biological regulation that have been determined in experimental literature. This section addresses this issue using our new gene selection method which combines mi-

Table 7: Top 17 genes selected using hypothesis testing approach

Symbol	Gene Name
C10orf72	Chromosome 10 open reading frame 72
COL14A1	collagen, type XIV, alpha 1 (undulin)
NAV3	neuron navigator 3
TCF21	transcription factor 21
EMILIN1	elastin microfibril interfacer 1
CDO1	cysteine dioxygenase, type I
MAPRE2	Microtubule-associated protein, RP/EB family, member 2
STAR	steroidogenic acute regulator
CSDC2	cold shock domain containing C2, RNA binding
—	CDNA FLJ26796 fis, clone PRS05079
—	CDNA clone IMAGE:4820330
RECK	reversion-inducing-cysteine-rich protein with kazal motifs
NAP1L3	nucleosome assembly protein 1-like 3
GATM	glycine amidinotransferase (L-arginine:glycine amidinotransferase)
AOX1	aldehyde oxidase 1
ECM2	extracellular matrix protein 2, female organ and adipocyte specific
TNXA /// TNXB	tenascin XA pseudogene /// tenascin XB

Table 8: Putative biomarkers that are functionally annotated

Method	Genes in the Pathways
SVM-RFE	USF, CD22 /// MAG, TNXB
PCA	SPP1, C7, TNXB
1D-SVM	FOXO1A, PROS1, FHL2, PMM1, SDC2, RPL23, PPT2 /// EGFL8
Hypothesis Testing	RECK, AOX1, TNXB, GATM, CDO1



1939_at	1974_s_at	31618_at
5.539827	3.073215	2.369813
5.111291	2.910548	2.408342
5.409016	2.993643	2.354059
5.48663	3.033717	2.50029
4.747199	2.809641	2.29626
5.245248	2.880551	2.346067
5.483351	2.977558	2.306824
6.81125	2.993481	2.440053
4.043823	2.914132	2.416612
5.983891	2.846116	2.33175
5.298495	2.915615	2.402546
5.410288	3.056386	2.439583
5.126858	3.00556	2.494014
5.506011	2.95961	2.375551
4.84111	2.939081	2.412443
5.288934	2.762932	2.244824
5.379822	2.772443	2.292874
5.365991	2.861566	2.34872
6.277346	2.912886	2.394742
5.291503	2.981102	2.479613
5.535999	3.185127	2.398427
0.76	0.84	0.39
50%	54%	54%

Figure 4: Gene Expression Values of Probes on Gene TP53

croarray data analysis with the domain knowledge: Gene Ontology [3, 8] information from the associated Affymetrix annotation files (HU95Av2). This proposed method aims to discover those biologically meaningful genes whose gene expression values are non-differentially expressed across the different biological samples; but may be recovered by ontological linkage to genes that are differentially expressed.

The utility of our methods will be demonstrated using the well-characterized tumor suppressor gene, TP53. In our microarray dataset, TP53 was measured by 3 probes (1939_at, 1974_s_at, 31618_at, the expression values are listed in Figure 4). The p-values, which are computed from Welch t-tests on the expression data of each probe, are listed in the last-but-one row of Figure 4, and the LOOCV classification results of the 1D-SVM classifiers of each probe are listed in the last row of Figure 4.

Gene Ontology (GO) is produced by Gene Ontology Consortium [5] to describe the function of gene products, their location in the cell and the biological process they are involved in. Three structured ontologies of defined terms have been established: Biological Process, Cellular Component and Molecular Function [3, 8]. Gene Ontology, developed manually by experts, is generally used for annotating genes. In particular, it is useful in interpreting the significantly differentiated genes selected from microarray data analysis, or used for further analysis like grouping/classifying the selected genes according to the functions of the genes, or biological process they are involved in [22, 19]. In summary, it is mainly used as a resource to understand, annotate or validate the gene selection results. Currently, there are several attempts which try to integrate GO-based similarity [31, 30, 32], or GO-based structure [11] into microarray analysis like missing value estimation, clustering, etc. In [37], the authors proposed to select informative genes from micorarray data by incorportating gene ontology. Our method differs from their method in several aspects: 1) we determine the discriminative capability of a GO term in a more sophisticated way rather than a simple statistics of the ratio of discriminative genes annotated by this GO-term, 2) we score genes using the sum of discriminative capability of the GO-terms annotated on the genes rather than using the raw discriminative score of the genes.

Before we present our biomarker discovery method, we first need to introduce the concept of *function group*, which is defined as a group of genes with the same GO term annotation. Since one gene could have more than one GO term annotation, it is possible that different function groups can have overlaps with each other. The gene expression data of a function group is composed of the expression data of the genes in the group. The discriminative capability of a function group is determined by discrimination among different patient groups of its gene expression data. We predicate our method on the assumption that the putative markers are those genes that belong to the maximum number of function groups with good discriminative capability.

Our method is composed of three steps (see Table 9). In the first step, we incorporate the gene ontology knowledge by dividing genes into functional groups according to their annotated GO terms. For example, in the application to our microarray data, we use the GO term annotations from the Affymetric annotation file for HG_U95Av2 microarray data. We only use GO terms in the first level of the GO term

[5] http://www.geneonotology.org/

Table 9: Incorporating Gene Ontology into Biomarker Discovery

```
Divide Genes into Function Groups
Compute the discriminative capability of each function group
    Obtain the gene expression submatrix of group $j$
    Compute SVM LOOCV accuracy through SVMRFE process
    Score group $j$ by $w_j = f(LOOCV\_full, max\_LOOCV)$
Rank gene $i$ by $s(i) = \sum_{j=1}^{m} m_{ij} * w_j$
(Optional) Normalize on the gene score
```

hierarchy for each of the three parts of gene ontology, i.e. hierarchies with root node (level 0) as biological process, cellular component, molecular function, respectively. We thus obtain a gene-function group mapping matrix (see Figure 5), $I = (I_{i,j})_{m,n}$, where I is a binary matrix and $I_{i,j} = 1$ means that gene i is included in function group j, n is the number of genes, m is number of groups. For example, the group of GO term with Locus Link ID 739 will consist of all genes in the microarray data that have molecular function 739, namely, DNA strand annealing activity.

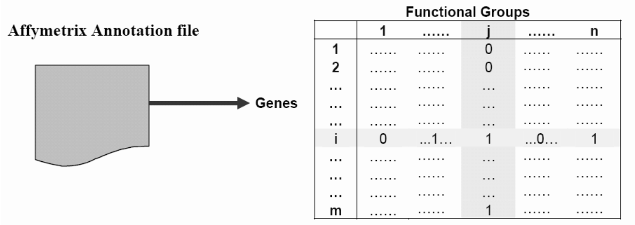


Figure 5: Illustration on the Algorithm - Step 1

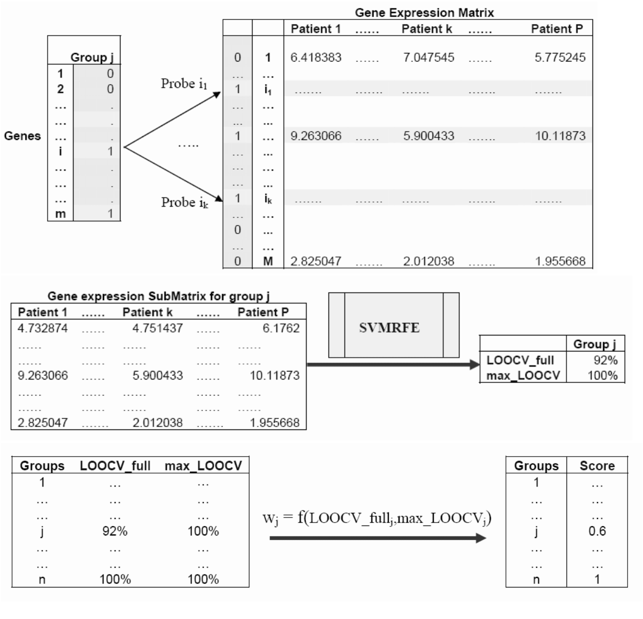


Figure 6: Illustration on the Algorithm - Step 2

In the second step: the discriminative capability of a function group is determined using SVM LOOCV accuracy rate through the SVMRFE process (described in Section 2). For each function group, we first obtain the corresponding gene expression value submatrix through the gene-probe mapping in the annotation file. Next, we evaluate the SVM LOOCV classification accuracy rate of the gene expression data through the SVMRFE process. We record two values in the process: i) LOOCV_full: the LOOCV classification accuracy with expression value of all the probes in the group. ii) max_LOOCV: the maximum LOOCV classification accuracy this gene expression dataset can achieve through recursive gene elimination process. We then score and rank the discriminative capability of the function group as $w_j = f(LOOCV\_full, max\_LOOCV)$, in which f is a thresholding function on the LOOCV classification accuracy.

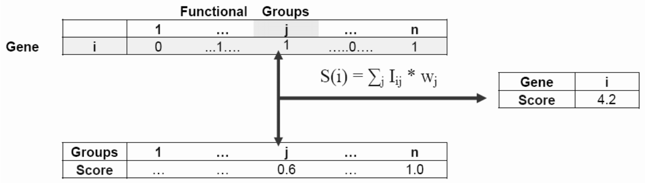


Figure 7: Illustration on the Algorithm - Step 3

In the third step, genes are scored and ranked based on how many functions groups a gene participates in and the discriminative capability of the function groups it belongs to. For each gene i, its score are computed as $s(i) = \sum_{j=1}^{n} I_{ij} * w_j$. We also considered normalization on the gene score to reduce bias toward genes having a large number of ontology entries, i.e. normalize the score of each gene by some penalty function based on how many gene group it belongs to. For example, give penalty p (some constant) on genes belong to more than $M * r$ groups (M: maximal number of gene groups a gene belongs to, r: penalizing ratio). But the normalization either generates worse results or gives little change to the results (position of TP53 in the gene rank list, and overlapped gene of the top500 genes using each of the three ontologies). Therefore, we just include it as an optional step of our algorithm, and we only show the results without gene score normalization.

We illustrate the proposed method in the example of TP53, which is of particular interests for ovarian cancer pathogenesis. Since TP53 is involved in 26 biological process based groups, 7 cellular component based groups, 12 molecular function based groups, we reported on the molecular function based grouping because space limitations on this paper. Table 10 lists the 12 function groups TP53 belongs to and their SVM LOOCV performance through the SVMRFE process. Thus the score for gene TP53 is $\sum_j I_{ij} * w_j = 8$. Therefore, gene TP53 has raw score 8 and it is among the top 350 genes (out of 12558) selected by this method. Table 11 lists the position of TP53 in the gene rank lists generated from our method by incorporating the annotation information from the biological process, cellular component, molecular function part of the Gene Ontology, respectively.

Table 12 lists the overlapped Genes from the top 500 genes obtained from our method by combining gene expression data with the annotation information from each of the three parts of the Gene Ontology: biological process, cellular component, and molecular function. There are 13 genes in total, where p-value is the minimum p-value of Welch t-test on the gene expression values of each of the probesets that measure the expression value of the gene, LOOCV

Table 10: Function Groups that TP53 belongs to

LocusLink ID	Annotated GO Term	# of Genes	LOOCV full	max LOOCV
739	DNA strand annealing activity	3	46%	54%
3677	DNA binding	1591	100%	100%
3700	transcription factor activity	890	100%	100%
4518	nuclease activity	74	100%	100%
5507	copper ion binding	58	75%	100%
5515	protein binding	3242	100%	100%
5524	ATP binding	1294	100%	100%
8270	zinc ion binding	1288	96%	100%
19899	enzyme binding	44	79%	92%
46872	metal ion binding	1674	100%	100%
46982	protein heterodimerization activity	89	96%	100%
47485	protein N-terminus binding	11	42%	63%

Table 11: Position of TP53 in the Gene Rank List Generated by the Method

	Biological Process	Cellular Component	Molecular Function
Rank of TP53	1	121	301
# of Genes that rank the same as TP53	0	47	50

Table 12: Overlap in the Top 500 Genes

Symbol	**Gene Name**	**p-value**	**LOOCV rate**
ADAM10	ADAM metallopeptidase domain 10	0.00072	79%
ALK	anaplastic lymphoma kinase (Ki-1)	0.25	54%
ATP1A1	ATPase, Na+/K+ transporting, alpha 1 polypeptide	0.077	58%
ATP2A2	ATPase, Ca++ transporting, cardiac muscle, slow twitch 2	8.9E-07	88%
CSF1R	colony stimulating factor 1 receptor, formerly McDonough feline sarcoma viral (v-fms) oncogene homolog	0.0063	71%
EGFR	epidermal growth factor receptor (erythroblastic leukemia viral (v-erb-b) oncogene homolog, avian)	0.0038	63%
EPHB1	EPH receptor B1	0.59	54%
GPR125	G protein-coupled receptor 125	0.37	50%
INSR	insulin receptor	1.2E-05	50%
NTRK1	neurotrophic tyrosine kinase, receptor, type 1	0.92	54%
PDGFRA	platelet-derived growth factor receptor, alpha polypeptide	7.2E-05	96%
RARA	retinoic acid receptor, alpha	0.68	54%
TP53	tumor protein p53 (Li-Fraumeni syndrome)	0.39	54%

rate is the maximum SVM LOOCV classification accuracy on the gene expression values of each of the probesets that measure the gene. As shown from the table, whether genes like ADAM10, ATP2A2, CSF1R, EGFR, INSR, PDGFRA are either with low p value or high LOOCV classification accuracy, which can be detected through traditional gene selection method, our method is capable to select out non-differentially expressed genes like ALK, ATP1A1, EPHB1, GPR125, NTRK1, RARA, and TP53. Since these additional genes are recovered in the same way that TP53 was recovered, they warrant further investigation.

From a biological viewpoint, by combining microarray data with gene ontology annotation information, our method seems capable of detecting potential biomarkers whose expression values are not significantly different but are likely to be mutated or regulated at post-translation levels. For example, TP53 function has been implicated in the clinical response among those patients treated with chemotherapy prior to surgery by Prof. McDonald's lab [23]. Also the functional annotation analysis from the Database for Annotation, Visualization, and Integrated Discovery (DAVID) [9] on our gene list shows that: i) 8 of the 13 putative biomarkers found by our method (see Table 12) are proteins capable of being phosphorylated (i.e. post-translationally modified); ii) 8 of these putative biomarkers listed in Table 12 are annotated by those biological pathways that have high gene enrichment confidence [24]. Table 13 lists these biological pathways and the associated putative biomarkers found by our method, where p-value is computed from a modified fisher exact test adopted in DAVID which measures the enrichment of the pathway annotations on the input gene list, the smaller, the more enriched. Comparing to the results in Table 8, we see an increase in the number of putative biomarkers involved in biological pathways. We also received more positive feedbacks from the biologists: the biological investigations on the roles these putative biomarkers plays in ovary cancer pathogenesis are being conducted.

5. CONCLUSIONS

In this paper, we present a method for augmenting microarray analysis with gene ontology data to provide insight on possible biomarkers (critical genes) for ovarian cancer pathogenesis which is not possible with microarray data alone. Using expression data for 12558 genes in 43 patients with both benign and malignant epithelial ovarian tumors, we apply representative state-of-the-art methods for microarray biomarker analysis including support vector machines, five data normalization methods (MAS5.0, MBEI,

Table 13: Pathway Analysis (from DAVID database) on the putative Biomarkers

Database	Pathway Term	p-value	Genes from Table 9
BIOCARTA	h_cblPathway: CBL mediated ligand-induced downregulation of EGF receptors	0.0017	CSF1R, PDGFRA, EGFR,
BIOCARTA	h_telPathway: Telomeres	0.0804	TP53, EGFR,
KEGG_PATHWAY	HSA05120: EPITHELIAL CELL SIGNALING IN HELICOBACTER PYLORI INFECTION	0.0857	ADAM10, EGFR
KEGG_PATHWAY	HSA04060: CYTOKINE-CYTOKINE RECEPTOR INTERACTION	0.0835	CSF1R, PDGFRA, EGFR
KEGG_PATHWAY	HSA04020: CALCIUM SIGNALING PATHWAY	0.0440	PDGFRA, ATP2A2, EGFR
KEGG_PATHWAY	HSA04010: MAPK SIGNALING PATHWAY	0.0129	TP53, NTRK1, PDGFRA, EGFR

PLIER, RMA, GCRMA), four feature selection methods, and two dimensionality reduction methods (PCA, LLE). Our findings showed that for this data 1) GCRMA appears to outperform other oligonucleotide microarray normalization methods through evaluation on reconstruction error after dimension reduction, as well as the SVM LOOCV classification accuracy through SVMRFE process; 2) the classification problem alone is not constraining enough to yield unique biomarkers with high confidence. Our new method combines statistical microarray analysis with ontological information. The result indicates that our method is capable of finding key regulators of oncogenesis whose expression values are non-differentially expressed at gene expression level but may be mutated or regulated at the post-transitional level, as is TP53 [23].

Based on the current work, there are several possible future research directions. Several studies are possible to improve the approach: i) we can compare the normalization methods on another data set or multiple data sets to get a more conclusive evidence that GCRMA is indeed superior; ii) We could benefit by incorporating the full hierarchical structure of GO in our analysis. We would also improve on our biomarker discovery methods to consider not only gene-class correlation (relevance) but also gene-gene correlation (redundancy) [28]. And gene ontology based similarity [31] would be a good measure for redundancy between genes. We would further incorporate domain knowledge on biological pathways, such as KEGG (Kyoto Encyclopedia of Genes and Genomes) PATHWAY[6], BioCarta[7]databases, into biomarker discovery, since one of the ultimate goals of biomarker discovery is to analyze their roles in the pathological pathways. We could evaluate these putative biomarkers through Hidden Markov Model sequence analysis/classification, as well as the gene-expression/function correlation analysis. Additional analyses of the literature and/or experimental procedures will be needed to verify the biological significance of the non-differentially expressed genes identified here to ovarian cancer metastasis.

6. ACKNOWLEDGMENTS

We would like to thank all the people from Prof. McDonald ś lab, and Prof. Gray ś lab for their feedbacks and comments on this work.

7. REFERENCES

[1] Affymetrix. *Guide to Probe Logarithmic Intensity Error (PLIER) Estimation.*

[2] N. Armstrong and M. van de Wiel. Microarray data analysis: from hypotheses to conclusions using gene expression data. *Cell Oncol.*, 26(5-6), 2004.

[3] M. Ashburner, C. Ball, J. Blake, D. Botstein, H. Butler, J. Cherry, A. Davis, K. Dolinski, S. Dwight, J. Eppig, M. Harris, D. Hill, L. Issel-Traver, A. Kasarskis, S. Lewis, J. Matese, J. Richardson, M. Ringwald, and G. Rubin. Gene ontology: tool for the unification of biology gene ontology. *Nature Genetics*, 25(1):25–29, 2000.

[4] Y. Benjamini and Y. Hochberg. Controlling the false discovery rate: a practical and powerful approach to multiple testing. *Journal of the Royal Statistical Society. Series B. Methodological*, 57(1):289–300, 1995.

[5] Y. Benjamini and D. Yekutieli. The control of the false discovery rate in multiple testing under dependency. *Annuals of Statistics*, 29(4):1165–1188, 2001.

[6] B. Bolstand, R. Irizarry, M. Astrand, and T. Speed. A comparison of normalization methods for high density oligonucleotide array data based on variance and bias. *BMC BioInformatics*, 19(2), 2003.

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

[8] G. O. Consortium. The gene ontology (go) database and informatics source. *Nucleic Acids Research*, 32(Database Issue), 2004.

[9] G. J. Dennis, B. Sherman, D. Hosack, J. Yang, W. Gao, H. Lane, and R. Lempicki. David: Database for annotation, visualization, and integrated discovery. *Genome Biology*, 4(9), 2003.

[10] K. Duan, J. Rajapakse, H. Wang, and F. Azuaje. Multiple svm-rfe for gene selection in cancer classification with expression data. *IEEE Transaction on Nanobioscience*, 4(3), 2005.

[11] Z. Fang, J. Yang, Y. Li, Q. Luo, and L. Liu. Knowledge guided analysis of microarray data. *Journal of Biomedical Informatics*, 39(4):401–411, 2006.

[12] L. Fisher and G. van Belle. *Biostatistics: A Methodology for the Health Sciences.* New York : John Wiley and Sons, 1993.

[13] T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, C. Bloomfield, and

[6] http://cgap.nci.nih.gov/Pathways/Kegg_Standard_Pathways

[7] http://cgap.nci.nih.gov/Pathways/BioCarta_Pathways

E. Lander. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, 1999.

[14] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machine. *Artificial Intelligence in Medicine*, 2002.

[15] S. Holm. A simple sequentially rejective mutliple test procedure. *Scandinavian Journal of Statistics*, 6:65–70, 1975.

[16] T. Huang and V. Kecman. Gene extraction for cancer diagnosis by support vector machines–an improvement. *Artificial Intelligence in Medicine*, 2005.

[17] R. Irizarry, B. Bolstad, F. Collin, L. Cope, B. Hobbs, and T. Speed. Summaries of Affymetrix GeneChip probe level data. *Nucleic Acids Research*, 31(4), 2003.

[18] R. Irizarry, B. Hobbs, F. Collin, Y. Beazer-Barclay, K. Antonellis, U. Scherf, and T. Speed. Exploration, normalization, and summaries of high density oligonucleotide array probe level data. *Biostatistics*, 4(2):249–264, 2003.

[19] R. Johnson, J. Williams, B. Schreiber, C. Elfe, K. Lennon-Hopkins, M. Skrzypek, and R. White. Analysis of gene ontology features in microarray data using the proteome bioknowledge library. *In Silico Biology*, 5(4):389–399, 2005.

[20] I. Jolliffe. *Principal Component Analysis*. Series in Statistics. Springer, 2nd edition, 2002.

[21] C. Li and W. Wong. Model-based analysis of oligonucleotide arrays: expression index computation and outlier detection. *Genome Biology*, 98:31–36, 2001.

[22] S. Li, M. Becich, and J. Gilbertson. Microarray data mining using gene ontology. *Medinfo*, 11(2):778–782, 2004.

[23] C. Moreno, L. Matyunina, E. Dickerson, N. Schubert, N. Bowen, S. Logani, B. Benigno, and J. McDonald. Evidence that p53-mediated cell-cycle-arrest inhibits chemotherapeutic treatment of ovarian carcinomas. *PLoS ONE*, 2(e441), 2007.

[24] National Institute of Allergy and Infectious Diseases (NIAID), NIH. *Functional Annotation Tool in DAVID database*, 2001. http://david.abcc.ncifcrf.gov/content.jsp?file=functional_annotation.html.

[25] T. Paul and H. Iba. Gene selection for classification of cancers using probabilistic model building genetic algorithm. *Bio Systems*, 82(3), 2005.

[26] S. Roweis and L. Saul. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 2000.

[27] S. Roweis and L. Saul. Think globally , fit locally, unsupervised learning of low dimensional manifolds. *Journal of Machine Learning Research*, 4:119–155, 2003.

[28] G. Smyth and T. Speed. Redundancy based feature selection for microarray data. In *Proceeding of ACM KDD*, pages 737–743, 2004.

[29] Y. Su, T. Murali, V. Pavlovic, M. Schaffer, and S. Kasif. Rankgene: identification of diagnostic genes based on expression data. *BMC Bioinformatics*, 19(12), 2003.

[30] J. Tuikkala, L. Elo, O. Nevalainen, and T. Aittokallio. Improving missing value estimation in microarray data with gene ontology. *BMC Bioinformatics*, 22(5):566–572, 2006.

[31] H. Wang, F. Azuaje, O. Bodenreider, and J. Dopazo. Feature dimension reduction for microarray data analysis using locally linear embedding. In *IEEE Symp. Computational Intelligence in Bioinformatics and Computational Biology*, pages 31–35, 2004.

[32] H. Wang, F. Azuaje, O. Bodenreider, and J. Dopazo. Incoporating gene ontology in clustering gene expression data. In *IEEE Symp. Computer-Based Medical Systems*, pages 31–35, 2006.

[33] Y. Wang, I. Tetko, M. Hall, E. Frank, A. Facius, K. Mayer, and H. Mewes. Gene selection from microarray data for cancer classification : a machine learning approach. *Computational Biology and Chemistry*, 29(1), 2005.

[34] S. Warrenfeltz, S. Pavlik, S. Datta, E. Kraemer, B. Benigno, and J. McDonald. Gene expression profiling of epithelial ovarian tumours correlated with malignant potential. *Mol Cancer.*, 3(27), 2004.

[35] W. Wu, E. Xing, C. Myers, I. Mian, and M. Bissell. Evaluation of normalization methods for cdna microarray data by k-nn classification. *BMC Bioinformatics*, 6(191), 2005.

[36] Z. Wu, R. Irizarry, F. Gentlemen, F. Martinez-Murillo, and F. Spencer. A model-based background adjustment for olignonucleotide expression arrays. *Journal of the American Statistical Association*, 99(468):909–917, 2004.

[37] X. X. A. Zhang. Selecting informative genes from microarray dataset by incorporating gene ontology. In *Fifth IEEE Symposium on Bioinformatics and Bioengineering*, pages 241–245, 2005.

[38] X. Zhou and K. Mao. Ls bound based gene selection for dna microarray data. *BMC Bioinformatics*, 21(8), 2005.