Learning Yeast Gene Functions from Heterogeneous Sources of Data Using Hybrid Weighted Bayesian Networks

Xutao Deng
Dept of Comp. Sci.
Univ. of Nebraska at Omaha
Omaha, NE 68182
xdeng@mail.unomaha.edu

Huimin Geng
Dept of Pathology & Microbio.
Univ. of Nebr. Medical Center
Omaha, NE 68198
huimingeng@unmc.edu

Hesham Ali
Dept of Comp. Sci.
Univ. of Nebraska at Omaha
Omaha, NE 68182
hali@mail.unomaha.edu

Abstract

We developed a machine learning system for determining gene functions from heterogeneous sources of data sets using a Weighted Naive Bayesian Network (WNB). The knowledge of gene functions is crucial for understanding many fundamental biological mechanisms such as regulatory pathways, cell cycles and diseases. Our major goal is to accurately infer functions of putative genes or ORFs (Open Reading Frames) from existing databases using computational methods. However, this task is intrinsically difficult since the underlying biological processes represent complex interactions of multiple entities. Therefore many functional links would be missing when only one or two source of data is used in the prediction. Our hypothesis is that integrating evidence from multiple and complementary sources could significantly improve the prediction accuracy. In this paper, our experimental results not only suggest that the above hypothesis is valid, but also provide guidelines for using the WNB system for data collection, training and predictions. The combined training data sets contain information from gene annotations, gene expressions, clustering outputs, keyword annotations and sequence homology from public databases. The current system is trained and tested on the genes of budding yeast Saccharomyces cerevisiae. Our WNB model can also be used to analyze the contribution of each source of information toward the prediction performance through the weight training process. The contribution analysis could potentially lead to significant scientific discovery by facilitating the interpretation and understanding of the complex relationships between biological entities.

Keywords: Bayesian network, gene function prediction, machine learning, yeast

1. Introduction

A primary goal of molecular biology is to understand the functional role of molecular machinery and their interactions. Traditional biological approaches to determining gene functions mainly focus on testing specific hypotheses through ingeniously designed experiments (see examples in [1, 2]). However, methods of this kind suffer from the high cost of labor and funds so that they typically do not scale well to deal with the great complexity of biological systems. Genome-scale sequencing and microarray projects provide researchers a big picture of genome structure and behavior. This big picture offers the opportunity to study functional genomics in an alternative way, that is, the machine learning approach. Machine learning takes advantage of the increasingly cheaper computing resources and rapidly growing biological databases. Successful machine learning methods include gene recognition, motif finding, gene clustering, gene function classification, protein profiling, regulatory networks reconstruction, and so on (reviewed in [3]).

This paper addresses the problem of inferring gene functions by integrating biological information such as DNA sequences, expressions, gene structures, database annotations and homologies. This problem can be viewed as a function classification of new genes or Open Reading Frames (ORFs) using heterogeneous sources of information. Gene function prediction has long been a difficult problem. Earlier studies focus on classification of gene functions based on single source of data such as protein homology and gene

expressions. It is reported that Support Vector Machines (SVMs) and *K*-Nearest Neighbors (KNNs) offer the best prediction performance among other methods using gene expression data [4-7]. However the studies also showed that the prediction accuracy is generally poor even with SVMs and KNNs, which can only achieve about 40% accuracy. Homology based methods can be regarded as a special case of KNNs classifier and they are also widely used for assigning functions to new genes. The results from the *S. cerevisiae* genome project [8] have illustrated both the potential and limitations of homology analysis as a means of assigning functions to new genes. One of the problems is its inability to assign functions for those yeast genes (about 30%) which have no homologs in the databases.

Besides the expression and DNA sequence homology data, information from protein sequence and structure, keywords of literature abstracts in major databases provide opportunities for improving the prediction of gene functions. Intensive research has been conducted in this direction. To name a few examples, LOCkey [9] is a lexical analysis system which annotates protein functions based on keywords from SWISS-PROT. It is reported that 82% classification accuracy is achieved by using LOCkey for fewer than half of all proteins in SWISS-PROT [9]. King [10] presented a protein-homology-based function inferring method which inducts rules from public databases through logic programming; the rules are then used to enhance the weak homologue found by PSI-BLAST. Pavlidis et al. [5] developed a predicting tool that integrates expression data with homology data based on SVMs.

Despite these efforts, some fundamental questions remain. What methods should we choose for my specific data? How to do the analysis if my data does not contain all the inputs the system need? What data should we collect first for the prediction? What are the contributions of each kind of data toward the prediction? How to extend the current system to hand new types of information? How to weight each kind of evidence for the prediction?

An observation is that specific methods are designed primarily for certain kinds of data so that the above questions can not be easily answered. For example, SVMs prefer numerical data while logic programming prefers symbolic data. Integrating new types of data to those existing data is non-trivial and error-prone. There is a need to develop a general purpose system that could easily integrate and weight each source of data in order to achieve better prediction performance.

These questions and observations motivate us to design a simple, extendable, and high performance system that is able to handle all kinds of data. In this paper, we present a hybrid weighted naive Bayesian Network (WNB) model which can elegantly integrate literally all kinds of evidence for predicting the functions of new genes/ORFs and proteins. It has been proved that Bayesian network classifiers provide superior performance among others [11]. The goal is two-fold. First, we provide a computational tool that can effectively predict the cellular and biological functions of novel genes and ORFs by integrating evidence from multiple sources of data. We require that the tool is easily extended to accommodate the dynamic nature of biological technology. Second, the tool can be used to analyze the contribution of each source of data toward the gene function prediction performance.

The remainder of this paper is organized as follows. Section 2 introduced WNB and its basic computing. Section 3 explains the WNB system architecture and learning and prediction algorithms. Section 3 describes the statistics of multiple data sets, the preprocessing of data, the smoothing and probability models for each data source. In section 4, we demonstrate the results with yeast data sets.

2. The WNB Computing Model

A Bayesian network [12, 13] is a graph-based model for representing probabilistic relationships between random variables. The random variables, which may represent source data such as gene expression levels, are modeled as graph nodes; probabilistic relationships are captured by directed edges between the nodes and conditional probability distributions associated with the nodes. Formally, a Bayesian network for a set of random variables is a pair $B=<G, \theta>$, where the first component is the network structure and the second component is the numerical parameters for conditional distributions associated with each node. Bayesian networks have been applied to solve many data mining tasks such as classification and diagnosis. In classification, a classifier, which assigns a class label to an example, is induced from a set of training examples with class labels. A simple Bayes classifier, called Naive Bayes (NB), is one of the most widely used classification models. NB has a special node *C*, representing class labels, which is the parent of all other attributes nodes A_1, A_2, …, A_n, where n is the number of feature attributes for each instances. An observation O is represented by a vector $(a_1, a_2,\ldots,a_n)$, where a_i is the

value of A_i. A NB has an umbrella structure (e.g. the oval nodes in Figure 1 and the edges between them), that implies the conditional independence assumption; that is, given the value of the class node, each attribute variable is independent with each other. This assumption, however, enables us to decompose the likelihood function and posterior probability $\Pr(c|a_1, a_2,\ldots,a_n)$ so that both learning and inference can be performed in a timely fashion. The classification using NB is based on the score of posterior probability, which is defined as:

$$\begin{aligned} &Score(c \mid a_1, a_2, \ldots, a_n) = \log \Pr(c \mid a_1, a_2, \ldots, a_n) + \alpha \\ &= \log \Pr(c) + \sum_{i=1}^{n} \log \Pr(a_i \mid c) \end{aligned} \quad (1)$$

We can see that the score for each class label c is defined as a linear combination of the logarithm likelihood. The classification using NB is based on the above posterior probability over all possible class labels, where $\hat{c}$ is the predicted class label for an instance and m is the number of possible classes for all instances.

$$\hat{c} = \arg\max_{c_j} \left(Score(c_j \mid a_1, a_2, \ldots, a_n) \right) \quad j = 1, 2, \ldots, m \quad (2)$$

A natural extension of NB is the so called WNB (weighted Naive Bayes), as shown in Figure 1, in which each edge is assigned a numerical value called weight and the WNB is a tuple $B=<G, \theta, W>$. The weight vector $W=<w_1,w_2,\ldots,w_n>$ represent the contribution of each attribute toward the classification.

$$Score_w(c \mid a_1, a_2, \ldots, a_n) = \log \Pr(c) + \sum_{i=1}^{n} w_i \log \Pr(a_i \mid c) \quad (3)$$

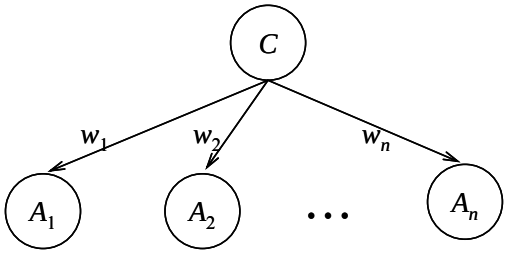

Figure 1. Network structure of Naive Bayesian Classifier

We choose WNB as our computing model for four reasons: First, it is easy to integrate numerical, symbolic and textual data in a Bayesian framework because probability is the common language for all data types. Second, the inference and learning in a Bayesian network model is localized which means that the model can be easily extended to handle the dynamic nature of biological databases. In addition, features with missing value for new genes can be dealt with consistently and easily. Third, the weights are automatically determined through a training procedure which is equivalent to a feature selection process. This automatic feature selection process avoids subjective human selection and optimizes the prediction accuracy. Fourth, it has been proved that probability classifiers are among the most effective and efficient classification algorithms in many domains.

3. System Architecture and Training Algorithms

The system architecture for gene function prediction is shown in Figure 2. The central part of the system is a WBN (oval nodes with edges between them) which has a fixed network structure G and is specified through training unknown parameters θ and W. The probability dependency of WBN is modeled from the top down and the inference is from the bottom up using Bayesian theorem. Heterogeneous data sets relevant to gene functions are collected from public databases such as SWISS-PROT, SGD [14], and MIPS [15]. Then the data were filtered and preprocessed to generate desired statistics that are ready for training. The model parameter θ for each attribute is determined by training and is stored in the modeler. When a new type of data is available, we can easily add a wrapper without changing the existing components of the system. For discrete distributions such as the keywords, the conditional probability θ is trained by Maximum Likelihood methods (counting the relative data frequency) and then smoothed using Laplace correction (adding a small pseudo count to each frequency). The continuous probability density is estimated using Gaussian kernel functions. The weight vector W is trained by optimizing the classification accuracy through hill-climbing. When the system is fully specified after training, it can be used to infer functions for novel genes through the Bayesian inference engine that applies the equations in the previous section. The entire system is hybrid in the sense that both continuous and categorical variables are included.

We use a hill-climbing algorithm to determine the value of W that optimizes the classification accuracy *Acc*. The classification accuracy *Acc* is defined as

$$Acc = \frac{\#hits}{\#hits + \#misses}, \quad (4)$$

After the parameter of conditional distribution θ is fixed by training, *Acc* can be regarded as a function of the weight parameter *W*. In hill-climbing, the optimization of *Acc* is performed by a search process consisting of a sequence of steps. In each step, the weight is revised to achieve higher *Acc*, according to the rule below:

$$w_i^N \leftarrow \arg\max_w \begin{cases} Acc(w_i^{N-1} + \delta^{N-1}) \\ Acc(w_i^{N-1} - \delta^{N-1}), \ N = 1, \ 2, \ ..., \\ Acc(w_i^{N-1}) \end{cases} \quad (5)$$

where w_i^N is the weight of the *i*th weight parameter at step *N*; δ^N is the step size at step *N* which is defined in proportion to the progress of *Acc*:

$$\delta^N = \begin{cases} \eta((Acc(w_i^{N-1}) - Acc(w_i^{N-2})) & \text{if } N > 1 \\ \delta^0 & \text{if } N = 1 \end{cases} \quad (6)$$

where η is the learning rate and δ^0 is the initial step size. The algorithm starts with NB settings, that is, all weights are assigned to 1. We adjust the weight of each attribute separately. The algorithm will stop when all w_i's won't change any longer. As the name of hill-climbing suggests, *Acc* monotonously increases during the steps and will converge to a certain local optimum (possibly a global optimum). The advantage of WNB is that it not only outperforms NB by automatic feature scaling but also gives each attribute's contribution toward the classification task.

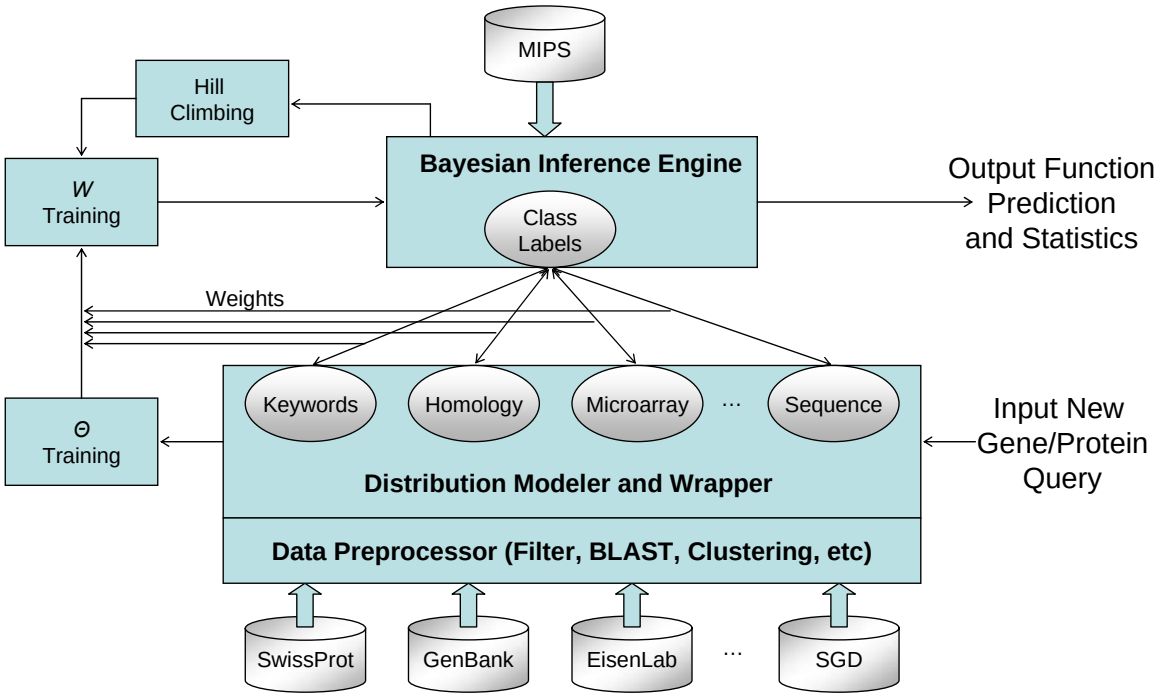

Figure 2. System architecture of the WBN and its training and prediction components.

4. Data Description and Attribute Modeling

We applied the method to a data set from budding yeast *S. cerevisiae*. Yeast is an excellent model organism which has reasonably simple genome structure, well-characterized gene functions, and huge expression data sets. It should be noted that given a suitable training set, the proposed method can also be applied to other organisms such as human beings. Deriving probability model for each attribute is a non-trivial problem. However, the general guideline is that the model should be able to describe the data in a parsimonious way. In other words, the model should not underfit or overfit the data. A rule of thumb is that the number of parameters in the model should be at most 1/10 of the size of the training set.

We collected four types of data sets believed to be relevant to the functions of yeast genes and their products. The first data set derived from a collection of

DNA microarray hybridization experiments [16]. Each data point represents the logarithm of the ratio of expression levels of a particular gene under two different experimental conditions. The data consists of a set of 79-element gene expression vectors for 2,465 yeast genes, which were selected by Eisen et al., based on the availability of accurate functional annotations. For gene expression data, we could simply model each experiment as a continuous attribute. Alternatively, we could perform a clustering analysis to partition the group of genes into a set of clusters. It was reported [16] that the clustering may provide some insight of the functional class. Under clustering, the likelihood of an expression vector **e** given certain class label *c* can be calculated as

$$\Pr(\mathbf{e} \mid c) = \sum_{clu} \Pr(\mathbf{e}, clu \mid c) = \sum_{clu} \Pr(\mathbf{e} \mid clu, c) \Pr(clu \mid c) \\ = \sum_{clu} \Pr(\mathbf{e} \mid clu) \Pr(clu \mid c) \quad (7)$$

To simplify the computation in the last equal sign, we assume that given the cluster a gene belongs to, its actual expression pattern is no longer important to its functional class. This is exactly what clustering is designed for—providing a summarized and compact representation for gene expressions. In the simplest form, the probability of **e** given a cluster *clu* is calculated as

$$\Pr(\mathbf{e} \mid clu) = \begin{cases} 1 & \text{if } clu = \arg\max_{clu'}(sim(\mathbf{e},\ clu')) \\ 0 & \text{otherwise} \end{cases} \quad (8)$$

where $sim(\mathbf{e}, clu)$ is the similarity between expression vector **e** and the centroid of *clu*. Note that this probability can also be estimated using other estimators, such as continuous kernel functions.

The second data set consists of a phylogenetic profile [17] for each of the 2,465 genes. For each yeast gene, a phylogenetic profile is a list of similarity measurements between yeast and other genomes indicating magnitude the gene of interest has a close homolog in the corresponding genome. The profiles employed in this paper contain, at each position, the negative logarithm of the lowest E-value, which was reported by BLAST version 2.0 [18] in a search against a complete genome, with negative values (corresponding to E-values greater than 1) truncated to 0. Two genes in an organism can have similar phylogenetic profiles for one of two reasons. First, genes with a high level of sequence similarity will have, by definition, similar phylogenetic profiles. Second, for two genes which lack sequence similarity, the similarity in phylogenetic profiles reflects a similar pattern of occurrence of their homologs across species. This coupled inheritance may indicate a functional link between the genes, based on the hypothesis that the genes are always present together or always both absent because they cannot function independently of one another. This data set has been compiled in [5]. They used 24 complete genomes collected from the Institute for Genomic Research website (www.tigr.org/tdb) and from the Sanger Centre website (www.sanger.au.uk). Similar to the gene expression data, we use a continuous random variable to model each position in the phylogenetic profile.

The third data set includes attributes regarding DNA sequences and gene structure such as the number of exons, the gene location (chromosome), the length of the ORF, GC contents, 6-mer entropy and codon usages. *S. cerevisiae* contains a haploid set of 16 well-characterized chromosomes. The average GC contents for all the genes is 0.403 with the minimum 0.314 and maximum 0.580. The 6-mer entropy ranges in bits from 4.07 to 7.82 with an average 6.81. The average size of yeast genes is 1.45 kb, or 483 codons, with a range from 40 to 4,910 codons. Only 3.8% of the ORFs contain introns. We also build codon usage distributions for the 64 codons (61 coding and 3 stop) since the information could discriminate genes of different classes.

From the DNA sequences and database annotations, we build probability models of these attributes for each functional class. One of the goals of this paper is to establish the relationship between DNA annotations and its protein function using machine learning techniques. Data on DNA sequences and gene annotations may not have a direct relationship with the gene functions, but with the automatic feature selection procedure, it would be desirable for integrating all sources of information in order to improve the gene function prediction accuracy.

The fourth data set is obtained from SWISS-PROT which contains biochemical functional annotations — keywords. The keyword annotation is at a very detailed level which provides indirect information for cellular information; for example, a given sequence has keyword *cdc2* kinase but not involved in intracellular communication [9]. A number of text-mining tools have been implemented that infer various aspects of cellular functions from annotations of biochemical functions [9, 20]. Annotations from SWISS-PROT currently form a dictionary of 890 relevant keywords for yeast functions. Each of the keywords is modeled as a binary discrete random variable and their probability of appearance for a given function class is estimated by maximum likelihood method.

The attributes and their models in the current system are summarized in Table 1. The combined data sets can be accessed at website http://bioinformatics.ist.unomaha.edu/~xdeng.
Functional class labels were obtained from the Munich Information Center for Comprehensive Yeast Genome Database (CYGD). The experiments reported here use 12 classes, each containing 90 genes or more for the purpose of reliable training and testing. The 12 classes have no direct inheritance relationship but there exist genes with multiple function annotations (See detail in Table 2).

Table 1. Summary of the information of source data and modeling method

Attribute	Model Type	# of Variables	Smoothing Method	Source
Class Label	12-nary	1	Laplace	MIPS FUNCAT SCHEME VERSION 2.0 http://mips.gsf.de/genre/proj/yeast/
#Exon	3-nary	1	Laplace	SGD http://www.yeastgenome.org/
GC%	continuous	1	Gaussian	
ORF Length	continuous	1	Gaussian	
Chromosome	16-nary	1	Laplace	
6-mer Entropy	continuous	1	Gaussian	
Codon Usage	multinomial (64 states)	1	Laplace	
Microarray	continuous	79	Laplace	http://rana.lbl.gov/EisenData.htm
Gene Cluster	12-nary	1	Laplace	–
Homology	continious	24	Gaussian	http://www.cs.columbia.edu/compbio
Keywords	binary	890	Laplace	SWISS-PROT http://www.ebi.ac.uk/swissprot/

Table 2. Class labels and descriptions used in experiments

Class Label	Description	Number of Instances
01	METABOLISM	597
02	ENERGY	148
10	CELL CYCLE AND DNA PROCESSING	359
11	TRANSCRIPTION	495
12	PROTEIN SYNTHESIS	267
14	PROTEIN FATE	411
20	CELLULAR TRANSPORT	409
32	CELL RESCUE, DEFENSE AND VIRULENCE	134
34	INTERACTION WITH THE CELLULAR ENVIRONMENT	194
40	CELL FATE	95
42	BIOGENESIS OF CELLULAR COMPONENTS	231
43	CELL TYPE DIFFERENTIATION	175
Total		3515 (2166 unique genes)

5. Experimental Results

Using the proposed WNB system, we performed computational experiments for each configuration of source data. A configuration is a binary string representing whether a source information is available or not. The four digits represent keywords, homology, expression, sequence respectively (e.g., 0110 means homology and expression are available while keywords and sequence annotations are not). For each configuration, we ran WNB with 10-fold cross-validation six times. To reduce the risk of overfitting,

we use four weights, one for each source of data, not one for each attribute. Negative weights are allowed in case that certain attributes may contribute negatively toward the classification. The mean and standard deviation of final weights and accuracy are summarized in Table 3.

To test for the probability of our predictions occurring by chance, we performed a binomial test for the random classifier with probability of hit $p = \mathbf{P} \cdot \mathbf{P}'$ where $\mathbf{P}$ is a row vector with each element representing the prior probability of a class. Both normal approximation and Monte-Carlo simulation were performed for the test and we conclude that all of the accuracy results in Table 3 are statistically highly significant (<1e-10).

Table 3. Results of classification accuracy for all possible input data configurations

Source Data	w_1 Keywords	w_2 Homology	w_3 Expression	w_4 Sequence	Accuracy (%)
0001	–	–	–	1.00	37.01±0.36
0010	–	–	1.00	–	32.95 ±0.66
0011	–	–	0.77±0.08	1.12±0.09	43.26±0.35
0100	–	1.00	–	–	24.77±0.46
0101	–	0.87±0.17	–	1.12±0.15	39.03±0.34
0110	–	1.00±0.13	1.04±0.06	–	37.56±0.25
0111	–	1.00±0.14	1.08±0.10	1.01±0.03	44.17±0.26
1000	1.00	–	–	–	80.59±0.13
1001	1.87±0.28	–	–	0.09±0.02	80.65±0.25
1010	1.89±0.24	–	0.06±0.04	–	80.66±0.28
1011	2.28±0.31	–	0.00±0.07	0.06±0.05	80.73±0.42
1100	1.81±0.13	0.05±0.11	–	–	80.59±0.39
1101	2.11±0.18	0.03±0.04	–	0.04±0.02	80.96±0.20
1110	2.08±0.30	0.03±0.03	0.13±0.03	–	80.61±0.20
1111	2.30±0.20	0.02±0.06	0.13±0.02	0.03±0.05	80.99±0.18

One important observation from Table 3 is that if the keywords are used as one of attributes in the WNB system, they dominate the prediction. With keywords present, all configurations reach classification accuracy around 80% which is significantly greater than configurations without keywords; while the contributions of other attributes are very low if not zero, evidenced by the weights given in the Table 3. We also performed paired two-tailed t-tests with 5% significance level which show that there are no significant differences in accuracy between all pairs of configurations when keywords are present in both.

When the keywords are absent, the accuracy ranges from 24% to 44%. The results are quite competitive compared with methods designed specifically for corresponding data sets e.g. SVM for expression [5], LOCkey for SWISS-PROT keywords [9]. We performed paired two-tailed t-tests with 5% significance level and have the following conclusions:

$$H < E < S \approx HE < HS < ES < HES << K^*$$

K: Keyword, *H*: Homology, *E*: Expression, and *S*: Sequence. For example, *HS* means that homology and sequence information are present while keyword and expression information are absent. This result is rather important since it empirically proved our hypotheses that using the information of multiple sources is able to improve prediction accuracy. The missing functional links may be complemented evidence by other sources of data. The WNB users are encouraged to include as many data sources as available, because our system has a hill-climbing component that is able to reach high accuracy by strengthening the signal data sources and suppressing the noise data sources. This result also provides hints on the priority of choosing data sources when

predicting gene functions. For example, it is interesting that we find the sequence annotations (GC contents, codon usage, etc.) provide more direct information of gene functions than the expression data do. We also find that when Expression is replaced by Clustering, the accuracy is slightly (but statistically significant) lower.

Figure 3 shows the dynamics of the weights and accuracy during one run with configuration 1111. As we discussed, keyword provides a strong signal about the functions and it dominates all other attributes and forces them to almost zero during hill-climbing. The initial weights are all 1's which represent the setting of standard NB. It is clear that the WNB (Accuracy 0.80) significantly outperforms NB (Accuracy 0.58).

The confusion matrix of the multi-classification is shown in Table 4. We removed all genes with multiple class labels and this left 1223 uniquely labeled genes. A total of 983 hits were determined using WNB with initial configuration 1111. From this table, we can observe that all classes with reasonable sample size (>100) achieved good prediction accuracy (>70%).

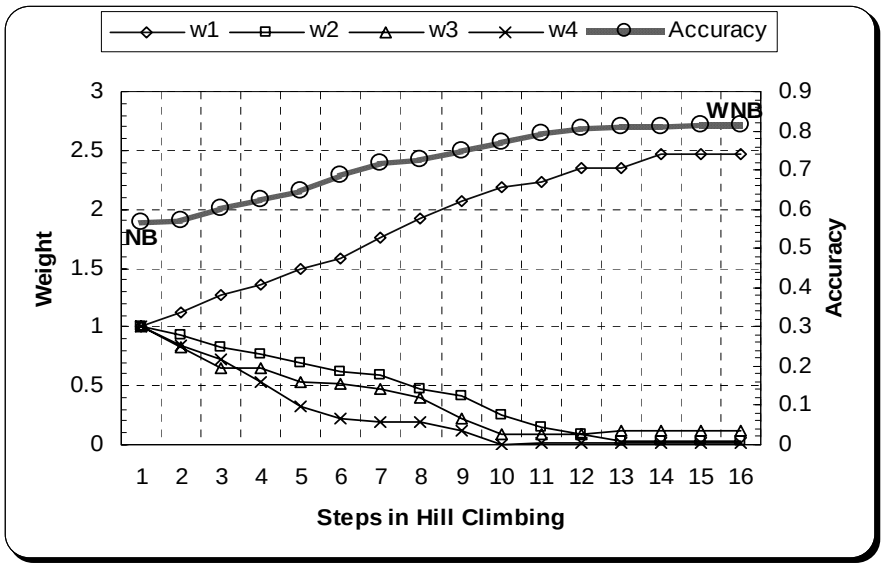

Figure 3. The dynamics of weights and accuracy during the steps in hill climbing

Table 4. Confusion Matrix of cases with unique function labels

Pred / True	01	02	10	11	12	14	20	32	34	40	42	43	Sub
01	**215**	8	2	3	2	14	9	0	1	2	0	1	257
02	6	**16**	0	0	0	1	1	0	0	0	1	0	25
10	4	0	**108**	13	3	3	7	0	0	0	2	2	142
11	5	0	8	**227**	2	3	4	0	0	0	2	0	251
12	1	0	0	5	**163**	1	2	0	0	0	0	0	172
14	3	0	8	4	3	**94**	17	1	0	0	0	0	130
20	4	0	1	3	0	5	**149**	0	0	6	0	1	169
32	6	0	6	3	4	5	3	**4**	0	1	0	0	32
34	1	1	1	2	0	3	5	0	**3**	0	0	0	16
40	0	0	0	0	0	0	0	0	0	**0**	0	0	0
42	5	0	2	5	0	1	3	0	0	1	**2**	1	20
43	0	0	2	1	0	2	1	0	0	1	0	**2**	9
Sub	250	25	138	266	177	132	201	5	4	11	7	7	**983/1223**

In order to study the prediction performance of individual class, we cast the 12-class classification problem as 12 separate one-versus-all (OVA) binary comparisons. We performed Receiver Operating Characteristic (ROC) analysis for each of the 12 binary classifiers. ROC analysis [19] is a ranking-based method that compares the classifiers' performance across the entire range of class distributions and error costs. The ROC curves and their corresponding Area Under ROC (AUC) for configuration 1111 are illustrated in Figure 4. Each of these 12 binary OVA comparisons has its own AUC, which can be used as a metric of how well the classifier separates one class from all the others. Each subplot shows the binary classification performance for each class. The comparison is between each class and its complement class which include all other classes. Figure 4 demonstrates that the AUCs are relatively high in all 12 classes, ranging from 0.673(CELL RESCUE, DEFENSE AND VIRULENCE) to 0.967 (PROTEIN SYNTHESIS). These results suggest a relatively high confidence level when using the WNB system for gene functional prediction.

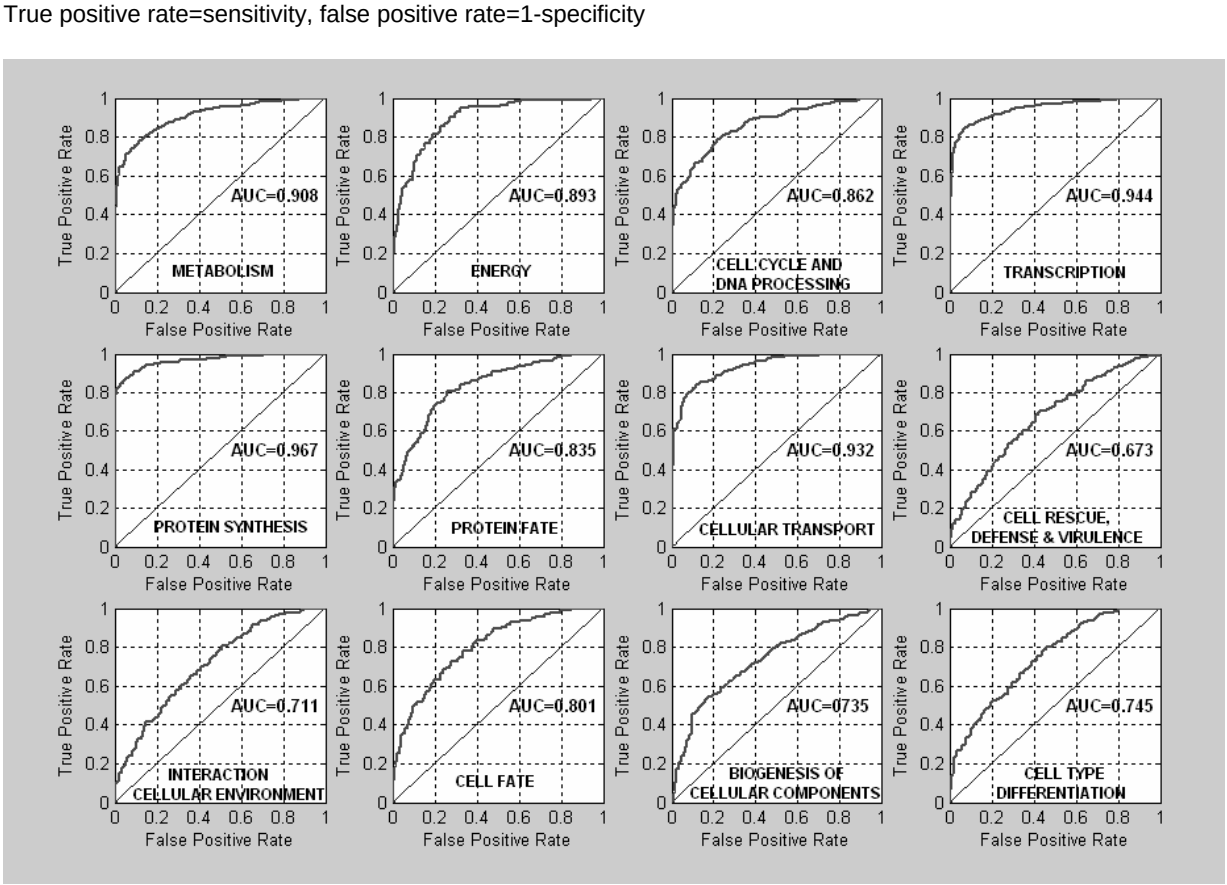

Figure 4. Roc curves of 12 one-versus-all comparisons. Each subplot represents the binary classification performance of each class compared with its complement class.

5. Conclusions

We present a Bayesian framework for the prediction of functional classes of novel genes based on heterogeneous sources of data. There are two main contributions in this study. First, we provide a system that is able to predict functions of novel genes, based on virtually all sorts of information that is available. Second, the system can be used to analyze how the combination of heterogeneous types of data affects the classification results through weight training process. The system has many advantages including capability of handling new data, missing data, and automatic feature selection. We have applied our WNB system to the heterogeneous data sets including gene annotations, gene expressions (clustering results), keyword annotations and sequence homology from public databases of budding yeast *S. Cerevisiae.* We conclude that the SWISS-PROT keywords is the dominant information source (~80% accuracy) for determining gene functions among other data sources, such as expression, homology and sequence statistical features, each achieved less than 40% accuracy. Moreover, the use of multiple data sets generally improved prediction accuracy. The performance for single data type is competitive with other approaches such as SVM and LOCkey, which are designed for specific training data sets. We also compiled a heterogeneous data set for future development in this field. We are currently trying to integrate DNA motif patterns and protein structure information in the system to see how the prediction performance is affected.

6. Acknowledgements:

Thanks to Michael Eisen, Paul Pavlidis and all those who deposit their experimental data in public databases and to those who maintain these databases. This work was supported by the NIH grant number P20 RR16469 from the IMBRE program of national center for research resource.

7. References:

[1] M.J. Evans, M.B. Carlton, and A.P. Russ. Gene trapping and functional genomics. *Trends Genet.,* 13:370-374, 1997.

[2] A. Wach, A. Brachat, R. Pohlmann, and P. Philippsen. New heterologous modules for classical or PCR-based gene disruptions in *Saccharomyces cerevisiae*. *Yeast*, 10:1793-1808, 1994.

[3] P. Baldi and S. Brunak, *Bioinformatics: The Machine Learning Approach*, The MIT Press, 2nd edition, 2001.

[4] M.P.S. Brown, W.N. Grundy, D. Lin, N. Cristianini, C. Sugnet, T.S. Furey, Jr. M. Ares, and D. Haussler. Knowledge-based analysis of microarray gene expression data using support vector machines. *PNAS*, 97(1):262–267, 2000.

[5] P. Pavlidis, J. Weston, J. Cai, and W.S. Noble. Learning gene functional classifications from multiple data types. *Journal of Computational Biology,* 9(2):401-11, 2002.

[6] M. Kuramochi and G. Karypis. Gene Classification Using Expression Profiles: A Feasibility Study. In *Proceedings of the 2nd IEEE International Symposium on Bioinformatics & Bioengineering (BIBE 2001)*, 191-201, 2001.

[7] E. M. Marcotte, M. Pellegrini, M. J. Thompson, T. O. Yeates, and D. Eisenberg. A combined algorithm for genome-wide prediction of protein function. *Nature*, 402:83–86, 1999.

[8] B. Dujon. The yeast genome project: what did we learn? *Trends Genet.,* 12:263-270, 1996.

[9] R. Nair and B. Rost. Inferring sub-cellular localization through automated lexical analysis. *Bioinformatics*, 18 Suppl 1:S78-86, 2002.

[10] R.D. King. Applying Inductive Logic Programming to Predicting Gene Function. *AI Magazine,* 25(1):57-68, 2004.

[11] N. Friedman, D. Geiger, and M. Goldszmidt. Bayesian networks classifiers. *Machine Learning,* 29:131-163, 1997.

[12] J. Pearl. *Probabilistic reasoning for intelligent systems.* Morgan Kaufmann, San Francisco, 1988.

[13] D. Heckerman, D. Geiger, and D. Chickering. Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learnin,g* 20:197–243, 1995.

[14] K. Dolinski, et al. Saccharomyces Genome Database. http://www.yeastgenome.org/.

[15] H.W. Mewes, D. Frishman, U. Güldener, G. Mannhaupt, K. Mayer, M. Mokrejs, B. Morgenstern, M. Münsterkoetter, S. Rudd, and B. Weil. MIPS: a database for genomes and protein sequences. *Nucleic Acids Research,* 30(1):31-34, 2002.

[16] M.B. Eisen, P.T. Spellman, P.O. Brown, D. Botstein. Cluster analysis and display of genome-wide expression patterns. *PNAS,* 95:14863−14868, 1998.

[17] M. Pellegrini, E. M. Marcotte, M.J. Thompson, D. Eisenberg, and T.O. Yeates. Assigning protein functions by comparative genome analysis: protein phylogenetic profiles. *PNAS*, 96(8):4285–4288, 1999.

[18] S.F. Altschul, T.L. Madden, A.A. Schaffer, J. Zhang, Z. Zhang, W. Miller, and D.J. Lipman. Gapped BLAST and PSI-BLAST: A new generation of protein database search programs. *Nucleic Acids Research*, 25:3389–3402, 1997.

[19] A.P. Bradley. The use of area under ROC curve in the evaluation of learning algorithms. *Pattern Recognition*, 30(6):1145-1159, 1995.

[20] F. Eisenhaber, P. Bork. Evaluation of human-readable annotation in biomolecular sequence databases with biological rule libraries. *Bioinformatics,* 15(7-8):528-35, 1999.