

Gene expression

## Robust and efficient identification of biomarkers by classifying features on graphs

TaeHyun Hwang<sup>1</sup>, Hugues Sicotte<sup>2</sup>, Ze Tian<sup>1</sup>, Baolin Wu<sup>3</sup>, Jean-Pierre Kocher<sup>2</sup>, Dennis A. Wigle<sup>4</sup>, Vipin Kumar<sup>1</sup> and Rui Kuang<sup>1,\*</sup>

<sup>1</sup>Department of Computer Science and Engineering, University of Minnesota, Twin Cities, <sup>2</sup>Bioinformatics Core, Mayo Clinic College of Medicine, Rochester, <sup>3</sup>Division of Biostatistics, School of Public Health, University of Minnesota, Twin Cities and <sup>4</sup>Division of General Thoracic Surgery, Mayo Clinic Cancer Center, Rochester, MN, USA

Received on May 21, 2008; revised on July 19, 2008; accepted on July 21, 2008

Advance Access publication July 24, 2008

Associate Editor: Joaquin Dopazo

### ABSTRACT

**Motivation:** A central problem in biomarker discovery from large-scale gene expression or single nucleotide polymorphism (SNP) data is the computational challenge of taking into account the dependence among all the features. Methods that ignore the dependence usually identify non-reproducible biomarkers across independent datasets. We introduce a new graph-based semi-supervised feature classification algorithm to identify discriminative disease markers by learning on bipartite graphs. Our algorithm directly classifies the feature nodes in a bipartite graph as positive, negative or neutral with network propagation to capture the dependence among both samples and features (clinical and genetic variables) by exploring bi-cluster structures in a graph. Two features of our algorithm are: (1) our algorithm can find a global optimal labeling to capture the dependence among all the features and thus, generates highly reproducible results across independent microarray or other high-throughput datasets, (2) our algorithm is capable of handling hundreds of thousands of features and thus, is particularly useful for biomarker identification from high-throughput gene expression and SNP data. In addition, although designed for classifying features, our algorithm can also simultaneously classify test samples for disease prognosis/diagnosis.

**Results:** We applied the network propagation algorithm to study three large-scale breast cancer datasets. Our algorithm achieved competitive classification performance compared with SVMs and other baseline methods, and identified several markers with clinical or biological relevance with the disease. More importantly, our algorithm also identified highly reproducible marker genes and enriched functions from the independent datasets.

**Availability:** Supplementary results and source code are available at [http://compbio.cs.umn.edu/Feature\\_Class](http://compbio.cs.umn.edu/Feature_Class).

**Contact:** [kuang@cs.umn.edu](mailto:kuang@cs.umn.edu)

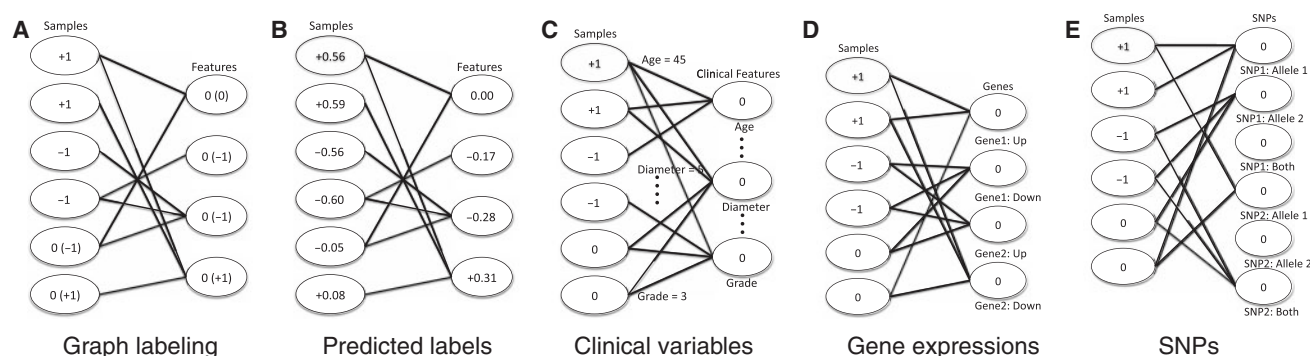
**Supplementary information:** Supplementary data are available at *Bioinformatics* online.

### 1 INTRODUCTION

Determining the causative factors of disease is critical for improving clinical treatment and understanding the biological principles of disease. Recent developments in high-throughput technology allow large-scale measurement of genomic variations such as gene expression and single nucleotide polymorphisms (SNPs) of a population. Associating these genomic and genetic variations with disease-related phenotypes provides good potential for elucidating etiology of diseases (Rebbeck *et al.*, 2007). It has also been shown that the discovered biomarkers can possibly provide better prognosis and diagnosis than the currently available clinical measures for risk assessment of patients with various diseases (Gevaert *et al.*, 2006; van't Veer *et al.*, 2002). However, computational identification of biomarkers of disease from high-throughput genomic data is an increasingly challenging problem. High-throughput data are both expensive to generate and difficult to obtain. Typically, only a small number of samples are available for analyzing tens of thousands of genes or even millions of SNPs. This analysis suffers from the curse of 'high-dimension and low-sample size', the number of samples being too limited to represent the class distribution of phenotypes.

Common statistical criteria for biomarker discovery are correlation coefficients (van't Veer *et al.*, 2002) and statistics used with hypothesis testing methods such as the *t*-test and Wilcoxon rank-sum test (Dudoit *et al.*, 2002). These statistical methods rank the features only based on their individual correlation with the phenotypic label. Feature selection is a more general machine learning approach for identifying biomarkers (Sun *et al.*, 2007). The objective of feature selection is to find a (minimal) subset of features that can maximize the prediction performance of a classifier. However, the curse of dimensionality makes feature selection on high-throughput data particularly hard and unstable. To maximize the prediction performance of a classifier, existing algorithms rely on heuristic strategies searching for a sub-optimal feature set. Moreover, the sub-optimal feature set might not be unique given that there are many co-expressed genes or SNPs with high-linkage disequilibrium, which have similar discriminative power. Thus, feature selection algorithms often fail to reveal the modularity on the features when used for biomarker identification. Many other supervised machine learning techniques have also been applied to identify clinical and genetic markers of disease. These approaches

\*To whom correspondence should be addressed.



**Fig. 1.** Feature classification and disease-marker identification on bipartite graphs. **(A)** This example shows a graph with six sample vertices and four feature vertices. All the edges are assumed uniformly weighted. Four samples are initially labeled according to their phenotype class; the other two and all the feature vertices are unknown and labeled 0. The optimal labels are given in the parentheses: the two feature vertices strongly connected to the negative vertices are labeled negative, the one feature vertex strongly connected to the positive vertices is labeled positive, and the one that is connected to both classes is assigned 0. The two unlabeled samples are also labeled according to their connections in the graph. **(B)** The prediction scores (activation values) produced by network propagation with  $\alpha=0.5$  and 1000 iterations on the graph in (A). All the nodes are correctly labeled; note that the labels are relaxed into real numbers. **(C)** A bipartite graph with vertices of clinical variables. **(D)** A bipartite graph with vertices of gene expressions; the edge weights are the absolute expression levels of the genes. **(E)** A bipartite graph with vertices of SNPs; all the edges are uniformly weighted by 1.

are typically variations of commonly used supervised learning algorithms, such as SVMs (Zhang *et al.*, 2006) and Bayesian networks (Gevaert *et al.*, 2006). However, these algorithms are not directly designed for the purpose of supervised biomarker discovery, and thus, variable selection relies on the interpretation of the trained classifiers.

One commonly acknowledged problem in biomarker discovery is that in reality, the lists of marker genes discovered from independent gene expression profiles rarely overlap, although the genes are often involved in common pathways (Chuang *et al.*, 2007; Yu *et al.*, 2007). For example, in the study of breast cancer, van't Veer *et al.* (2002) and Wang *et al.* (2005) identified two sets of marker genes related to the metastasis of breast cancer using large-scale gene expression profiles produced in two different microarray experiments. However, there are only three genes in common between the two sets of the marker genes. Although the non-replicability is partially introduced by the difference of the microarray platforms and the experiment techniques used for generating the high-throughput data, cluster structures or modularities on the genes, such as co-expression can be used to leverage that discrepancy. However, it is a computational challenge to explore the cluster structures among the features together with the label information for biomarker discovery.

In this article, instead of selecting features based on their discriminant power in classifying samples, we propose to use labeled samples to classify features. We introduce a semi-supervised learning algorithm to associate clinical variables, gene expressions and SNPs with specific phenotypes in a disease context. We formulate the biomarker discovery task as a 'hybrid' semi-supervised classification problem: we use training samples (positive and negative) to classify both the test samples and the features into positive, negative or neutral classes (Fig. 1A and B). The positively classified features and negatively classified features are candidate biomarkers. This new learning algorithm can capture the dependence between both samples and features (clinical or genetic variables) by exploring the global structure of the bipartite graph, based on a 'bi-cluster assumption': those samples in the same class tend to be

heavily connected to a common set of features; those features that can characterize a class tend to be heavily connected to the samples in the class. In the bipartite graph, *feature vertices* represent clinical variables, up/down-regulated genes or homozygous or heterozygous SNPs; *object vertices* represent labeled and unlabeled samples, connected to the feature vertices by weighted edges. The object vertices are labeled with  $-1/+1$  if the label is known, 0 otherwise. Every clinical variable is denoted by one vertex, and it is connected to all the samples by the edges weighted by the original clinical values (Fig. 1C). Every gene is represented by two vertices, up-regulated and down-regulated; each sample will be connected to either the up-regulated vertex or the down-regulated vertex with an edge weighted by the expression level (Fig. 1D). Each SNP will have three states, two homozygous states and one heterozygous state; every sample will be connected to one of the three vertices depending on the SNP type of this sample (Fig. 1E).

Our algorithm is in a family of label propagation algorithms (Bengio *et al.*, 2006; Zhou *et al.*, 2004), which can be regarded as semi-supervised spectral graph-learning techniques with a global optimal solution (Bengio *et al.*, 2006). Recognized as having good generalization and high efficiency, these algorithms are receiving increasing attention from both machine learning and computational biology research communities (Kuang *et al.*, 2005; Tsuda *et al.*, 2005; Weston *et al.*, 2004). The common property of all these graph-based learning algorithms is the 'cluster assumption': there are often subtle underlying cluster structures in a large graph, which can be used implicitly to improve classification of unlabeled samples. We formulate the problem differently by classifying objects and features together. Our formulation implicitly explores the bi-cluster structure (Cheng and Church, 2000) in the data and labels the feature vertices based on their connections in the bi-clusters and the labels on the training samples. The bi-cluster structure can leverage the classification of the features by imposing the modularity on the features. In other words, features that are in the same bi-cluster tend to get similarly labeled by our algorithm. This property effectively utilizes the dependence among all the features for our biomarker discovery task.

Our graph-based learning algorithm captures dependence between all features simultaneously by exploring the graph structure, which is essentially a non-linear method for selecting features. After relaxing the labels into real numbers, our method can always converge toward the unique global optimum using an efficient network propagation algorithm. The time complexity of the algorithm scales linearly with the total number of features given that our algorithm converges within a small number of iterations. Thus, our method is stable and fast to generate replicable results across independent datasets, even under the curse of dimensionality in biomarker identification. Finally, our semi-supervised learning algorithm can use unlabeled data in the process of classifying the features, which can possibly improve the quality of the selected features.

## 2 METHOD

In this section, we first define our formulation of marker discovery and disease diagnosis/prognosis as a semi-supervised learning problem on bipartite graphs. An efficient network propagation algorithm is then introduced to compute the closed-form solution of the objective function for the semi-supervised learning.

### 2.1 Semi-supervised learning on bipartite graphs

We formally define an undirected bipartite graph  $G=(V, U, E, w)$ , where  $V$  and  $U$  are two disjoint vertex sets and  $E \in V \times U$  is a set of weighted edges; each edge  $(v, u) \in E$  connects two vertices  $v$  and  $u$  with a positive weight  $w(v, u)$ . Let  $d(v) = \sum_{(v, u) \in E} w(v, u)$  and  $d(u) = \sum_{(v, u) \in E} w(v, u)$  denote the sum of the weights of the edges on the same vertex. Let  $y: V \cup U \rightarrow \{-1, 0, +1\}$  be the initialization function assigning initial labels to the labeled and unlabeled vertices in  $V$  and  $U$ . Let  $f$  denote a label-assignment function over vertex sets  $V$  and  $U$ . If we let  $V$  be the sample set and  $U$  be the variables/feature set, a label assignment on a variable indicates its association with a sample class. Under this context, we define an objective function over  $G=(V, U, E, w)$  as follows,

$$\Omega(f) = \sum_{(v, u) \in E} w(v, u) \left( \frac{f(v)}{\sqrt{d(v)}} - \frac{f(u)}{\sqrt{d(u)}} \right)^2 + \varrho \sum_{v \in V} (f(v) - y(v))^2 + \varrho \sum_{u \in U} (f(u) - y(u))^2, \quad (1)$$

where  $\varrho > 0$  is a regularization parameter for balancing the cost terms on the right side of the equation. The first term enforces a consistency between the strongly connected vertex pairs  $(u, v) \in V \times U$ . This term penalizes those  $f$  functions with a cost proportional to the  $w(v, u)$  if  $f$  assigns different labels to  $v$  and  $u$ . The second term is a fitting term which keeps the new label assignment consistent with the initial labeling. This can be viewed as a supervised way of minimizing the training errors measured by the difference between the initial labels  $y(v)$  and the new label  $f(v)$  for labeled vertices  $v \in V$ . For the unlabeled vertices  $v \in V$  with  $y(v)=0$ , the second term is used to regularize these  $f(v)$ s, such that the total cost is constrained. The third term is used in the same spirit to constrain the cost on the vertices in  $U$ .

If we restrict the labels to discrete values, i.e.  $f: V \cup U \rightarrow \{-1, 0, +1\}$ , minimizing  $\Omega(f)$  is NP hard. But if we relax the label values as  $f: V \cup U \rightarrow R$ ,  $\Omega(f)$  is convex and differentiable. Let  $D_U$  be a diagonal matrix with  $D_{i_u i_u} = d(u)$  and  $D_V$  be a diagonal matrix with  $D_{i_v i_v} = d(v)$ , where  $v \in V$  and  $u \in U$ , and  $i_v$  and  $i_u$  are the index of vertices  $v$  and  $u$  in the matrix. We define the normalized connectivity matrix  $S$  of  $G$  as follows,

$$S = \begin{bmatrix} 0 & D_V^{-\frac{1}{2}} * W * D_U^{-\frac{1}{2}} \\ D_U^{-\frac{1}{2}} * W^T * D_V^{-\frac{1}{2}} & 0 \end{bmatrix},$$

where  $W$  denotes a  $|V|$  by  $|U|$  matrix with  $W_{i_v, i_u} = w(v, u)$ . Similar to the derivation in Zhou *et al.* (2004), we can rewrite Equation (1) as follows,

$$\Omega(f) = [f(V)^T \ f(U)^T] * (I - S) * \begin{bmatrix} f(V) \\ f(U) \end{bmatrix} + \varrho \left\| \begin{bmatrix} f(V) \\ f(U) \end{bmatrix} - \begin{bmatrix} y(V) \\ y(U) \end{bmatrix} \right\|^2,$$

where  $I$  is the identity matrix. We then differentiate  $\Omega(f)$  with respect to  $f$  to compute the closed-form solution  $f^*$  for minimizing  $\Omega(f)$ ,

$$\frac{\partial \Omega}{\partial f} = 2(I - S) * f^* + 2\varrho(f^* - y) = 0.$$

Let  $\alpha = 1/(1 + \varrho)$  and after rearrangement, the closed-form solution  $f^*$  can be computed as follows,

$$f^* = \frac{\varrho}{1 + \varrho} \left( I - \frac{1}{1 + \varrho} S \right) * y = (1 - \alpha)(I - \alpha S)^{-1} * y. \quad (2)$$

### 2.2 Network propagation algorithm

It is computationally intensive to compute the matrix inverse in Equation (2), when the graph  $G$  is large and contains a lot of non-zero entries in  $S$ . We use a network propagation algorithm to compute the closed-form solution more efficiently. The propagation algorithm iteratively performs a diffusion operation between the two vertex sets in both directions. Theoretically, the diffusion process will finally converge to the closed-form solution  $f^*$  defined in Equation (2). The network propagation algorithm is described as follows.

- (1) Normalize the bipartite graph by computing  $B = D_V^{-\frac{1}{2}} * W * D_U^{-\frac{1}{2}}$ .
- (2) Choose parameter  $\alpha$  and perform a two direction propagation, until convergence ( $t$  denotes the time step):
  - For each  $v \in V$ ,  
 $f(v)^t = (1 - \alpha)y(v) + \alpha \sum_{u \in U} B_{i_v, i_u} f(u)^{t-1}$
  - For each  $u \in U$ ,  
 $f(u)^t = (1 - \alpha)y(u) + \alpha \sum_{v \in V} B_{i_v, i_u} f(v)^{t-1}$
- (3) The sequence  $f^t$  converges to its limit  $f^*$  and  $f^*$  gives the class labels on the unlabeled vertices in both  $V$  and  $U$ .

This algorithm propagates the label information of every vertex to its neighbors in the other vertex set. This propagation process will leverage the activation values of the vertices in a densely connected neighborhood. In other words, if we assume that the vertices with the same label tend to be in the same clusters in the graph, the vertices in the same class will eventually converge to having similar values (same labels). This iterative propagation process was originally proposed to spread the activation values in a psychology network (Shrager *et al.*, 1987). It is intuitively consistent with the definition of our objective function in Equation (1). In Figure 1C, we show the predictions of network propagation on a toy graph. Note that in the method by Kuang *et al.* (2005), a similar algorithm has been used for protein ranking, but the normalization of  $S$  is different and no regularization framework was introduced.

We can show that this algorithm will finally converge to the closed-form solution of the objective function  $\Omega(f)$ . We first rewrite the network diffusion algorithm in matrix form as,

$$f(V)^t = (1 - \alpha)y(V) + \alpha B * f(U)^{t-1}$$

$$f(U)^t = (1 - \alpha)y(U) + \alpha B^T * f(V)^{t-1},$$

which can be rearranged as  $f^t = (1 - \alpha)y + \alpha S * f^{t-1}$ . Following the proof by Zhou *et al.* (2004), we can show that  $f$  converges to  $f^* = (1 - \alpha)(I - \alpha S)^{-1} * y$ , which is exactly the closed-form solution in Equation (2).

The time complexity of the network propagation algorithm is  $O(k|V||U|)$ , where  $k$  is the number of iterations for reaching convergence.

Theoretically,  $k$  depends on some properties of the graph such as the eigenvalues of its Laplacian (Bengio *et al.*, 2006). Empirically, we observe that our network propagation algorithm converges very fast on the bipartite graphs in our experiments. For example, when the convergence is defined as the maximum change of activation values over all the graph nodes being smaller than  $1e-9$ , our algorithm converges between 10 and 200 iterations, depending on the choice of the  $\alpha$  parameter, on a dataset with 24 000 gene expressions (about 48 000 features in the graph).

### 3 EXPERIMENTS

We evaluated the network propagation algorithm on three public breast cancer datasets. We first show that our algorithm is a highly competitive classification algorithm in Section 3.2, and then we show that our algorithm identifies highly reproducible marker genes on independent microarray datasets in Section 3.3. We also analyze the convergence rate and measure the empirical running time of the network propagation algorithm in Section 3.4. Finally, we validate the marker genes identified by network propagation by comparing with known cancer genes in the literature and checking their biological functions in Section 3.5.

#### 3.1 Breast cancer datasets

We used three independent large-scale microarray gene expression breast cancer datasets (van de Vijver *et al.*, 2002; van't Veer *et al.*, 2002; Wang *et al.*, 2005) in our experiments. The three datasets were generated for studying breast cancer metastasis. The dataset (Rosetta dataset) in van't Veer *et al.* (2002) measures expression profiles of 24 481 genes generated by Agilent (Santa Clara, CA) oligonucleotide Hu25K microarrays as well as eight clinical variables: age, estrogen receptor positive (ERp), progesterone receptor positive (PRp), tumor size, tumor grade, angiogenesis, lymphocytic infiltration and BRCA1 mutation. This dataset contains 97 patient samples. Among the 97 patients, 51 patients had a good prognosis, meaning being free of disease after their diagnosis for an interval of at least 5 years, and 46 patients had developed distant metastasis within 5 years. The van de Vijver *et al.* (2002) dataset contains microarray gene expressions produced by the same technique for generating the Rosetta dataset on 295 samples (194 with good outcome and 101 with poor outcome). The details of the quantization and normalization of the scanned microarray images of the two datasets are described in van't Veer *et al.* (2002) and van de Vijver *et al.* (2002). The Wang *et al.* (2005) dataset was produced by the Affymetrix oligonucleotide microarray U133a GeneChip. The expression of 22 283 transcripts were collected from total RNA of frozen samples from 286 lymph-node-negative breast cancer patients. Among the 286 patients, 95 had developed cancer metastasis within 5 years and 114 had been free of metastasis for at least 8 years. These two groups of patients (209 in total) are used in our experiments. We normalized the Wang *et al.* dataset with GeneSpring (version 7.0) by per-gene and per-chip median polish.

#### 3.2 Sample classification

To validate that the identified discriminant features are indeed strongly correlated with the patient classes and can be used to classify samples accurately, we measured the classification results on the test samples. We compared the classification performance of the network propagation algorithm against SVMs with RBF kernel and linear kernel (Vapnik, 1998), linear discriminant analysis

**Table 1.** Sample classification

Algorithms	Rosetta		Vijver	Wang
	Clinical	Genes	Genes	Genes
(A) Classification results on three datasets				
Network propagation	<b>0.788</b>	<b>0.740</b>	<b>0.667</b>	<b>0.564</b>
SVM (linear)	0.773	0.730	<b>0.662</b>	0.536
SVM (RBF)	0.783	0.737	0.661	<b>0.568</b>
Naïve Bayes	<b>0.795</b>	0.617	0.476	0.554
LDA	0.579	<b>0.740</b>	0.648	0.502
(B) Comparison between network propagation and the baseline algorithms				
NP versus SVM (linear)	278/31/191	247/27/226	242/86/172	309/25/166
NP versus SVM (RBF)	248/44/208	214/124/162	254/81/165	137/130/233
NP versus Naïve Bayes	144/106/250	393/10/97	466/3/31	261/24/215
NP versus LDA	460/8/32	232/36/232	297/61/142	359/15/126

Panel A: the mean ROC scores of classifying patients with good/poor prognosis in the Rosetta dataset, the van de Vijver *et al.* dataset and the Wang *et al.* dataset using network propagation (NP), SVMs with linear and RBF kernels, naïve Bayes classifier and LDA. The two best performing algorithms in each experiment are marked in bold.

Panel B: the number of times of win/draw/loss on classification performance between network propagation and the baseline algorithms.

(LDA) and naïve Bayes classifier. The classification performance is evaluated using the receiver operating characteristics (ROC) score: the normalized area under a curve plotting the number of true positives against the number of false positives by varying a threshold on the decision values (Gribskov and Robinson, 1996). In all experiments, we run 5-fold cross-validation on the whole dataset. An additional cross-validation on the training set is used to select the best parameters and we compute ROC scores on the test set. We repeated the process 100 times and report the mean and the variance of the ROC scores for each method.

We tested the classification performance of the network propagation algorithm on the three datasets in four different experiment setups: (1) using eight clinical variables on the Rosetta dataset; (2) using all 24 481 gene expressions on the Rosetta dataset; (3) using all 24 481 gene expressions on the van de Vijver *et al.* (2002) dataset; (4) using all 22 283 gene expressions on the Wang *et al.* (2005) dataset. We prune the gene expression features with a cutoff of 0.3 on the absolute values of correlation coefficients calculated on the training samples in Experiment (2) and (3) and 0.2 in Experiment (4). In Table 1(Panel A), we report the mean of the ROC scores computed from 100 runs of 5-fold cross-validation on each dataset. The variance of all the methods are similar in each experiment, and are thus reported in Supplementary Tables 1–5. In Table 1(Panel B), we also report how many times network propagation wins or loses to the others. For a more rigorous comparison, we calculated the  $P$ -values with one-sided paired  $t$ -test or proportion test to evaluate whether network propagation performs better than the other algorithms over 100 runs of randomized 5-fold cross-validations. Overall the proposed network propagation has very competitive performance (see Supplementary Tables 1–5 for details).

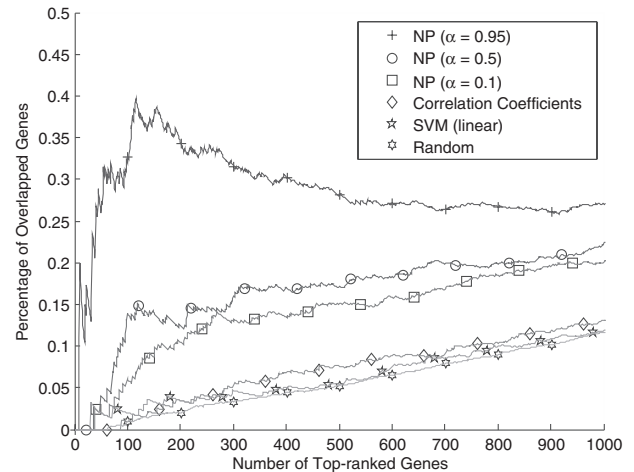
The results in Table 1(Panel A) show that in all experiments, network propagation is always among the two best performing algorithms. Although the naïve Bayes classifier performs best on the clinical data, it does not handle gene expression data well in the other three experiments. LDA does not perform well on the clinical data; furthermore, it is the worst performing algorithm on Wang *et al.* dataset. SVM with linear kernel and RBF kernel also perform stably well in all experiments. Although SVM with RBF kernel achieves the highest average ROC scores on the Wang *et al.* dataset and SVM with linear kernel is the best performing algorithm on the van de Vijver *et al.* dataset, network propagation is the best performing algorithm in the other two experiments. It appears that the difference between the performance of network propagation and SVMs are marginal if only measured by the mean ROC scores. However, the pairwise comparison between network propagation and SVMs shows that the differences are statistically significant either by the number of loss and win or by  $P$ -values. The comparisons with the baseline algorithms show that network propagation is a competitive classification algorithm for cancer outcome prediction and statistically, network propagation also has more robust performance in the four experiments.

### 3.3 High reproducibility of marker genes

To verify that network propagation identifies highly reproducible marker genes on independent microarray datasets, we report the number of common marker genes identified in the van de Vijver *et al.* dataset and Wang *et al.* dataset. Since the gene expressions in the two datasets are produced on different microarray platforms, there are only 8733 common genes that can be matched by the probe names. Thus, in this analysis we focus on using the labels of all the patients to identify the marker genes from the 8733 common genes independently on the two datasets.

After we ran network propagation to classify the gene features, we ranked the genes by the absolute value of their  $z$ -scores calculated from the activation values. We compared the percentage of common genes between the top-ranked genes in the two datasets identified by each method in Figure 2. We tested network propagation with three different  $\alpha$ -values (0.95, 0.5 and 0.1) and compared them with the commonly used correlation coefficients for identifying differentially expressed genes, SVM with linear kernel and the random case. The random case is calculated by the average ratio of common genes identified by network propagation on bipartite graphs with randomly permuted edges.

It is clear in Figure 2 that network propagation identifies significantly more reproducible marker genes on the two datasets. For example, among the top-100 genes selected by network propagation from the two datasets, there are 32 common genes when  $\alpha=0.95$ , 14 common genes when  $\alpha=0.5$  and 6 common genes when  $\alpha=0.1$ , while SVM with linear kernel and correlation coefficients can only identify two common genes. One interesting observation is that the  $\alpha$  parameter strongly influences the percentage of common genes: the larger the  $\alpha$  parameter, the more the common genes identified. This can be explained by our optimization formulation in Equation (1); when  $\alpha$  is large, we put more weight on the cluster structures in the bipartite graph and thus, network propagation favors the modularity structure in the gene expressions by assigning highly consistent weights to the coexpressed genes. In other words, those genes

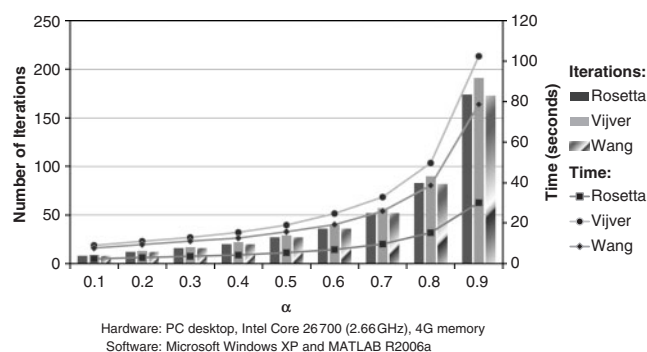


**Fig. 2.** Common marker genes identified by network propagation on the van de Vijver *et al.* dataset and the Wang *et al.* dataset. The  $x$ -axis is the number of selected marker genes ranked by  $z$ -scores converted from their activation values. The  $y$ -axis is the percentage of the overlaps between the selected markers from the two datasets.

that are highly coexpressed in the related functional modules will get highly weighted in both datasets. When  $\alpha$  is close to 1, our algorithm almost becomes a completely unsupervised learning algorithm; on the contrary, when  $\alpha$  is close to 0, our algorithm is similar to computing correlation coefficients for the features. Because our algorithm uses both cluster structures and label information to identify marker genes, it can retrieve more overlapped marker genes than those methods that ignore the dependence among the gene features, such as correlation coefficients and SVM with linear kernel. Our result also implies that on the two microarray datasets, although the overlap between the rankings of genes is almost random if the significance is computed independently, the modular structures between genes are still preserved. Network propagation is an effective way of exploring the modular structures to produce a more reliable gene ranking.

### 3.4 Convergence rate and running time

To test the convergence and the scalability of our network propagation algorithm, we measured the convergence rate and the running time of network propagation on the Rosetta dataset (97 samples, 24 481 genes), the van de Vijver *et al.* dataset (295 samples, 24 481 genes) and the Wang *et al.* dataset (209 samples, 22 283 genes). We define the convergence as the maximum change of activation values over all the graph nodes being smaller than  $1e-9$ . Theoretically, the convergence rate is decided by the Laplacian of the bipartite graph, which in our case is strongly related to the choice of the  $\alpha$  parameter. We tested nine different  $\alpha$  parameters and reported the running time and the number of iterations for reaching convergence in Figure 3. For all the choices of  $\alpha$  parameter on the three datasets, network propagation converges within 200 iterations. When  $\alpha$  is small, the algorithm converges in very few iterations. Intuitively, this can be explained by the nature of network propagation: when  $\alpha$  is large, the propagation operation puts more weight on the graph structure and less weight on the relatively static label information, and it takes more iterations to fully explore the



**Fig. 3.** Convergence and running time of network propagation. This plot shows the convergence rate and the running time of the network propagation algorithm on the three microarray datasets. The  $x$ -axis is the values of the  $\alpha$  parameter. The left  $y$ -axis is the number of iterations needed for reaching convergence. The right  $y$ -axis is the running time in seconds.

whole structure, given a certain threshold on the contribution of change on the activation values from the subtle structure of the bipartite graph. It is notable that on a regular PC, our algorithm only needs at most 103 s to reach a convergence for the datasets with more than 24 000 genes.

Empirically we observed that for any choice of  $\alpha$ , the number of iterations that network propagation takes to converge is independent of the number of genes in the dataset. Thus, the running time of the algorithm is approximately linear in the number of genes in the dataset. To verify this hypothesis, we randomly selected a certain number of genes from the van de Vijver *et al.* dataset and tested our network propagation algorithm for three different  $\alpha$  values. The result is reported in Supplementary Figure 1. Clearly, in all the three cases, the actual running time scales linearly in the number of selected genes.

### 3.5 Biological interpretations of marker genes

We compared the identified marker genes from all the genes in the van de Vijver *et al.* dataset and the Wang *et al.* dataset with previous findings in the literature. We also report the over-represented Gene Ontology functions by the selected marker genes. Finally, to assess the statistical significance of our results, we estimated  $P$ -values for the activation values of the marker genes by running network propagation on bipartite graphs with randomized edges. In all the experiments in this section, we used the label on all the patients to classify all the gene feature vertices. We chose  $\alpha = 0.5$  to have a balanced contribution from label information and graph structures.

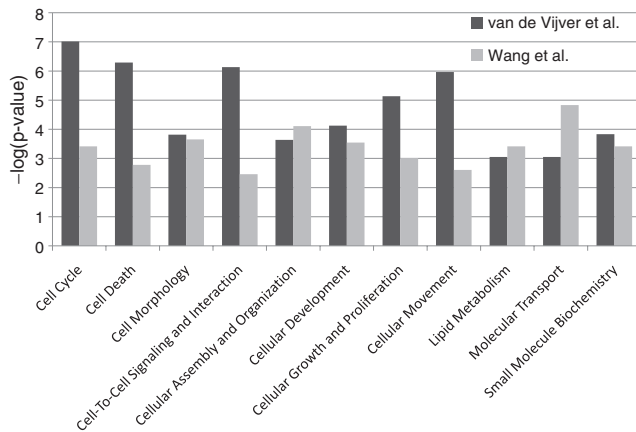
With a false discovery rate 0.18 and 0.1 on the van de Vijver *et al.* dataset and the Wang *et al.* dataset, respectively, we selected the top-200 genes in both datasets as marker genes for a consistent comparison. We used Ingenuity (version 5.5) to analyze the marker genes. Out of the top-200 genes ranked by network propagation ( $\alpha = 0.5$ ) in the van de Vijver *et al.* dataset and the Wang *et al.* dataset, only 141 genes and 140 genes, respectively, are eligible for searching the biological networks and annotations provided by Ingenuity. Our analysis focuses on the 141 and 140 eligible marker genes from the two datasets. Fifteen genes are in common between the 141 and 140 marker genes. Interesting examples in the 15 common genes are trefoil factor 1 (TFF1) and transmembrane 4L-six-family member

1 (TM4SF1). The expressions of TFF1 and TM4SF1 are decreased by breast cancer 1 early onset (BRCA1) and estrogen receptor 1 (ESR1), both of which are well-known breast cancer susceptibility genes. In comparison with the 15 common genes identified by network propagation, there is only one common gene between the top-200 genes ranked by correlation coefficients on the two datasets. The higher ratio of overlapped genes also indicates that network propagation captures the dependence between the genes of similar roles in those biological networks, and thus finds more common pathways associated with a disease from heterogeneous datasets.

From the marker genes on the van de Vijver *et al.* dataset and the Wang *et al.* dataset, Ingenuity reports 92 (out of 141) and 51 (out of 140) genes as cancer-related genes, respectively. In the list of marker genes identified from van de Vijver *et al.* dataset, some interesting examples are v-erb-b2 erythroblastic leukemia viral oncogene homolog 2 (ERBB2), baculoviral IAP repeat-containing 5 (BIRC5) and matrix metalloproteinase 9 (MMP9). ERBB2 is one of the well-known breast cancer susceptibility genes. The amplification of the ERBB2 oncogene is found in around 30% of human breast cancers and used as a target of therapy (van de Vijver *et al.*, 2002; van't Veer *et al.*, 2002). The expression of BIRC5 is regulated by other well-known breast cancer susceptibility genes such as estrogen receptor 1, ER- $\alpha$  (ESR1) and TP53. MMP9 is known for promoting breast cancer metastasis. We find other marker genes interacting with known breast cancer susceptibility genes such as reproto, TP53-dependent G2 arrest mediator candidate (RPRM) and prominin 1 (PROM), both of which are regulated by ESR1, and Cbp/p300-interacting transactivator (CITED1), which regulates ESR1, ESR2 and ERBB2. In the list of marker genes identified from the Wang *et al.* dataset, androgen receptor (AR) is known for association with survival of breast cancer patients; complement component 4 binding protein-beta (C4BPB), cadherin 3 (CDH3) and E74-like factor 5 (ELF5) are regulated by ERBB2 and MAGEA2. ERBB2 and MAGEA2 are known to be able to decrease trans-activation activity of TP53. The activities of other genes in the list such as Thymosin-like 8 (TMSL8), POU class 4 homeobox 1 (POU4F1), stratifin (SFN) and prominin 1 (PRPM1) also involve TP53. We also found that, although some known breast cancer susceptibility genes are not included in the marker genes, their neighbor genes that are known to interact with them are identified. Two examples are v-myc myelocytomatosis viral oncogene homolog (MYC) and v-Ha-ras Harvey rat sarcoma viral oncogene homolog (HRAS). We report identified functions associated with our marker genes in Supplementary Tables 6–7.

Ingenuity identified 19 and 13 enriched functions scoring a  $P$ -value  $< 0.01$  and involving at least two marker genes on the van de Vijver *et al.* dataset and the Wang *et al.* dataset, respectively (Supplementary Tables 6–7). In Figure 4, we list the 11 common functions between the 19 and 13 enriched functions in the two datasets. The enriched functions show strong consistency with those identified by Hanahan (2000) and Wang *et al.* (2005), which have been shown to be significantly involved with the progression of cancer. Among the 11 functions, eight functions such as cellular growth and proliferation, cell death, cell cycle and, etc., are exactly or closely matched with the 21 functions discovered previously in Wang *et al.* (2005) (See Supplementary Tables 6–7).

To estimate the statistical significance of the marker genes, we compared the activation value on the  $k$ -th ranked feature vertex with its background distribution computed from 500 runs of



**Fig. 4.** Biological functions enriched by the markers genes identified by network propagation on the van de Vijver *et al.* dataset and the Wang *et al.* dataset. We list the functions scoring a  $P$ -value  $<0.01$  and involving at least two marker genes on both datasets. The functions are sorted by  $P$ -values calculated using the right-tailed Fisher exact test.

network propagation with randomized graph edges. Specifically, we randomize the graph edges by moving the edges to connect randomly selected samples and feature vertices. We ran network propagation on the random graphs and repeated it 500 times. For each random graph, we ranked the feature vertices by their activation values. We then used the 500 values on each  $k$ -th ranked feature vertex as a sample of the background distribution for obtaining the  $k$ -th rank in the list. We compared our observed activation values on the van de Vijver *et al.* dataset and the Wang *et al.* dataset with their respective background distributions by estimating  $P$ -values and scatter plotting. In Supplementary Figure 3, we show the scatter plot of the observed activation values and the expected activation values on the van de Vijver *et al.* dataset and the Wang *et al.* dataset. In the plots, clearly the largest and the smallest activation values significantly deviate from the expected values. In Supplementary Table 8, we list the top-50 marker genes identified on two datasets along with their  $P$ -values (See Supplementary website for the full list).

## 4 DISCUSSION

In this article, we present a new machine learning framework for supervised biomarker identification, in which we classify the features into ‘positive’, ‘negative’ and ‘neutral’ classes. We also design an efficient semi-supervised graph-based learning algorithm to compute the global optimal solution of this feature classification problem. In our experiments, we show that our algorithm can generate highly reproducible marker genes in two independent breast cancer datasets; we also show that our algorithm can handle hundreds of thousands of features simultaneously in  $<2$  min on a regular PC. One limitation of our current algorithm is that no prior knowledge on the dependence between the feature vertices such as linkage disequilibrium between SNPs, is used in the process of classifying features. We plan to design new algorithms that can utilize the dependence between the features as prior knowledge by running network propagation on graphs of more sophisticated topologies. Another limitation is that, although the

network propagation algorithm demonstrates high performance in classification, no improvement has been achieved from data integration (Supplementary Table 3). We postulate that naïve linear concatenation of different types of features is not a principled data-integration strategy for the network propagation algorithm. Thus, we are also designing new algorithms that can integrate different graphs in a non-linear manner. It has recently been shown that pathways and protein–protein interaction networks can improve the reproducibility of the marker genes (Chuang *et al.*, 2007). However, availability of pathway and protein–protein interaction data is often very limited. Our method can capture the modularity of the genes without using any extra information.

## ACKNOWLEDGEMENTS

We thank the anonymous reviewers for helpful comments and Minnesota Super-computing Institution (MSI) for providing the computational facility for the work in this article.

**Funding:** This work is supported by the Biomedical Informatics and Computational Biology Seed Grant for the University of Minnesota-Mayo-IBM Collaboration.

**Conflict of Interest:** none declared.

## REFERENCES

- Bengio, Y. *et al.* (2006) Label propagation and quadratic criterion. In Chapelle, E.O. *et al.* (eds), *Semi-Supervised Learning*. MIT Press, Cambridge, MA.
- Cheng, Y. and Church, G.M. (2000) Biclustering of expression data. In *Proceedings of the 8th International Conference on Intelligent Systems for Molecular Biology*. AAAI, La Jolla, California, pp. 93–103.
- Chuang, H. *et al.* (2007) Network-based classification of breast cancer metastasis. *Mol. Sys. Biol.*, **3**, 140.
- Dudoit, S. *et al.* (2002) Statistical methods for identifying differentially expressed genes in replicated cDNA microarray experiments. *Stat. Sin.*, **12**, 111–139.
- Hanahan, D. and Weinberg, R.A. (2000) The hallmarks of cancer. *Cell*, **100**, 57–70.
- Gevaert, O. *et al.* (2006) Predicting the prognosis of breast cancer by integrating clinical and microarray data with Bayesian networks. *Bioinformatics*, **22**, e184–e190.
- Gribnikov, M. and Robinson, N.L. (1996) Use of receiver operating characteristic (ROC) analysis to evaluate sequence matching. *Comput. Chem.*, **20**, 25–33.
- Kuang, R. *et al.* (2005) Motif-based protein ranking by network propagation. *Bioinformatics*, **21**, 3711–3718.
- Rebbeck, T.R. *et al.* (2007) Genetic association studies of cancer: where do we go from here? *Cancer Epidemiol. Biomarkers Prev.*, **16**, 864–865.
- Shrager, J. *et al.* (1987) Observation of phase transitions in spreading activation networks. *Science*, **236**, 1092–1094.
- Sun, Y. *et al.* (2007) Improved breast cancer prognosis through the combination of clinical and genetic markers. *Bioinformatics*, **23**, 30–37.
- Tsuda, K. *et al.* (2005) Fast protein classification with multiple networks. *Bioinformatics*, **21**, ii59–ii65.
- van de Vijver, M.J. *et al.* (2002) A gene-expression signature as a predictor of survival in breast cancer. *N. Engl. J. Med.*, **347**, 1999–2009.
- van't Veer, L.J. *et al.* (2002) Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, **415**, 530–536.
- Vapnik, V.N. (1998) *Statistical Learning Theory. Adaptive and Learning Systems for Signal Processing, Communications, and Control*. Wiley, New York.
- Wang, Y. *et al.* (2005) Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer. *Lancet*, **365**, 671–679.
- Weston, J. *et al.* (2004) Protein ranking: from local to global structure in the protein similarity network. *Proc. Natl Acad. Sci. USA*, **101**, 6559–6563.
- Yu, J.X. *et al.* (2007) Pathway analysis of gene signatures predicting metastasis of node-negative primary breast cancer. *BMC Cancer*, **7**, 182.
- Zhang, H. *et al.* (2006) Gene selection using support vector machines with non-convex penalty. *Bioinformatics*, **22**, 88–95.
- Zhou, D. *et al.* (2004) Learning with local and global consistency. In *Advances in Neural Information Processing Systems*, Vol. 16. MIT Press, Cambridge, MA, pp. 321–328.