2008 Eighth IEEE International Conference on Data Mining

Learning on Weighted Hypergraphs to Integrate Protein Interactions and Gene Expressions for Cancer Outcome Prediction

TaeHyun Hwang*, Ze Tian*, and Rui Kuang†
Department of Computer Science and Engineering
University of Minnesota Twin Cities
thwang, tianze, kuang@cs.umn.edu

Jean-Pierre Kocher
Bioinformatics Core
Mayo Clinic College of Medicine
Kocher.JeanPierre@mayo.edu

Abstract

Building reliable predictive models from multiple complementary genomic data for cancer study is a crucial step towards successful cancer treatment and a full understanding of the underlying biological principles. To tackle this challenging data integration problem, we propose a hypergraph-based learning algorithm called HyperGene to integrate microarray gene expressions and protein-protein interactions for cancer outcome prediction and biomarker identification. HyperGene is a robust two-step iterative method that alternatively finds the optimal outcome prediction and the optimal weighting of the marker genes guided by a protein-protein interaction network. Under the hypothesis that cancer-related genes tend to interact with each other, the HyperGene algorithm uses a protein-protein interaction network as prior knowledge by imposing a consistent weighting of interacting genes. Our experimental results on two large-scale breast cancer gene expression datasets show that HyperGene utilizing a curated protein-protein interaction network achieves significantly improved cancer outcome prediction. Moreover, HyperGene can also retrieve many known cancer genes as highly weighted marker genes.

1. Introduction

Finding gene predictors of cancer outcome from genomic data is becoming an increasingly important focus in cancer research under the assumption that the genomic information can shed light on the molecular mechanisms underlying cancer development and progression. In the past decade, enormous amount of large-scale microarray gene expression profiles have been produced to study different cancers such as breast cancer [18, 19], lung cancer [16] and prostate cancer [7] for the purposes of 1) detecting marker genes for cancer-relevant phenotypes and 2) building reliable predictive models for cancer prognosis or diagnosis. The two tasks are closely intervened with each other because on one hand, a predictive model built from highly predictive marker genes is often more accurate in outcome prediction; on the other hand, a highly accurate prediction model can also be analyzed to reveal unknown cancer marker genes. Different machine learning and data mining strategies for feature selection have been applied to identifying a subset of genes that can maximize the prediction performance of a classifier [18].

Although many interesting and promising findings have been reported in these studies, the reliabilities of the studies have been questioned with the concern on the unstable and inconsistent results in cross-validations and cross-platform comparisons due to the relatively small sample sizes in the studies [6]. To overcome this difficulty, it has been proposed to include other complementary genomic information such as pathway information or functional annotations to aid the process of model building and biomarker discovery such that the prior knowledge from the complementary data can improve the robustness of the model and result in more consistent discoveries across independent datasets [4, 3, 13]. The availability of large protein-protein interaction networks, which contain information on gene functions, pathways and modularity of gene regulations, provides a desirable source of data for this purpose. Protein-protein interactions can be derived from a number of experimental techniques such as yeast two-hybrid system and mass spectrometry [11]. The high consistency between the networks derived from different organisms allows integration of many small networks into a large scale network. It has been observed that cancer genes tend to be highly connected with each other in large scale protein-protein interaction networks [3]. It has been shown in [4] that by incorporating protein-protein interaction network into the model built from microarray gene expressions, the authors can improve cancer outcome prediction and get more reproducible

*The first two authors contributed equally to this work.
†Author to whom correspondence should be addressed

1550-4786/08 $25.00 © 2008 IEEE
DOI 10.1109/ICDM.2008.37

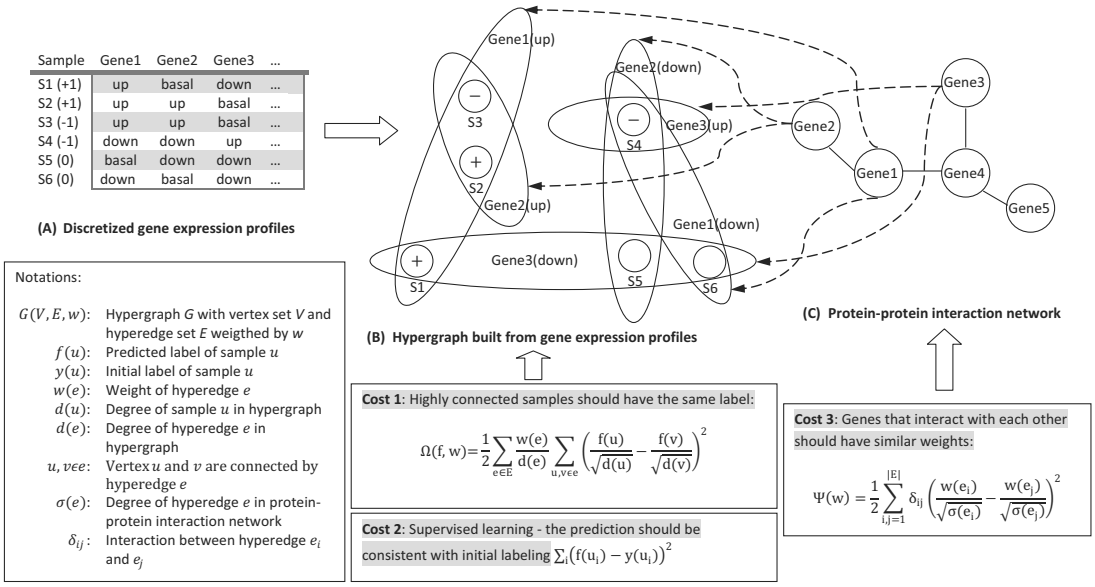


Figure 1. Regularization framework of HyperGene.

results on two large scale gene expression datasets. In their approach, the integration of gene expressions and protein-protein interactions is achieved by two independent procedures: discriminative subnetworks are first identified from a curated protein-protein interaction network and the subnetworks are then used as features to predict cancer metastasis. Authors in [13] proposed a method which first computes the spectral graph structure of a gene network and then, uses the spectral graph structure to smooth microarray gene expressions before used for sample classification. A statistics-based method is proposed in [3] to identify cancer genes by scoring genes by their degree in a cancer-specific interaction network, their differential expressions in microarray data and their structural, functional and evolutionary properties. However, designing a unified strategy for integrating protein-protein interactions and microarray gene expressions is still a challenging problem due to the complexity of a joint learning on two different data types.

In this paper, we propose a hypergraph-based iterative learning algorithm called HyperGene to integrate microarray gene expressions with protein-protein interactions for robust cancer outcome prediction and marker gene identification. The HyperGene algorithm minimizes a cost function under a unified regularization framework which elegantly takes a protein-protein interaction network as constraints on a hypergraph built from microarray gene expressions. The HyperGene algorithm is a natural extension of label propagation algorithms on hypergraphs [2, 1, 22]. HyperGene is based on a hypergraph in which each sample is denoted by a vertex and each gene is denoted by two hyperedges: a "up-regulated" hyperedge and a "down-regulated" hyperedge. The two edges group samples by the expression state (up/down) of the gene in the samples (Figure 1 A&B). Our cluster assumption on the hypergraph is that the samples of the same type tend to have similar gene expression patterns and thus are highly connected by the hyperedges. Since the original hypergraph-based learning algorithms assume uniform weighting of the hyperedges [1, 22], direct application of these algorithms to high-dimensional and noisy genomic data results in inferior prediction accuracy. The HyperGene algorithm is fundamentally different in reformulating the optimization problem as learning labels and hyperedge weights together with the assignment of edge weights constrained by a protein-protein interaction network. Essentially, to avoid overfitting training data, the HyperGene algorithm tries to find a weighting of hyperedges that nicely balances the two-class separation on the hypergraph and the consistency with the protein-protein interaction network. These properties of the HyperGene algorithm promise to improve prediction accuracy and provide more robust identification of marker genes. Furthermore, the resulted weights on the genes can be used to discover highly weighted subnetworks in the protein-protein interaction network, which might also suggest important pathways related to cancer outcomes.

2. Regularization Framework

In Figure 1, we show the regularization framework in our formulation. We first discretize gene expression profiles into three states: basal or up/down-regulated (Figure 1A), and build a hypergraph with (positive/negative/test) samples as vertices and gene expression states as hyperedges (Figure 1B). The regularization framework seeks for a global solu-

tion to both outcome prediction and gene weighting by considering the connectivities in the hypergraph, and the incorporation of the protein-protein interaction network provides useful prior knowledge on weighting interacting genes with similar values (Figure 1C). The cost function is defined on three loss terms: 1) inconsistent labeling of samples that are highly connected in the hypergraph; 2) inconsistent labeling of training samples with known outcomes; 3) inconsistent weighting of the hyperedges associated with the interacting genes in the protein-protein interaction network. Our objective is to find a solution that can minimize the weighted sum of the three loss terms.

2.1. Learning on weighted hypergraphs

A hypergraph is a special graph which contains hyperedges. In a simple graph, each edge connects a pair of vertices, but in a hypergraph each edge can connect arbitrary number of vertices in the graph. Hypergraphs are often used with algorithms for exploring higher order correlation between objects in data mining and bioinformatics [17, 20, 21]. Let $V = \{v_1, v_2, \ldots, v_{|V|}\}$ be a set of vertices and $E = \{e_1, e_2, \ldots, e_{|E|}\}$ be a set of edges defined on V: for any hyperedge $e \in E$, $e = \{v_1^{(e)}, v_2^{(e)}, \ldots, v_{|e|}^{(e)}\}$, where $\{v_1^{(e)}, v_2^{(e)}, \ldots, v_{|e|}^{(e)}\}$ is a subset of V. A hyperedge e and a vertex v are called incident if $v \in e$. A non-negative real number (a weight) can be assigned to each hyperedge by a function w (w can also be defined as a vector variable and we will use both notations interchangeably). The vertex set V, hyperedge set E and the weight function w fully defines a weighted hypergraph denoted by $G(V, E, w)$. The incidence matrix H for hypergraph $G(V, E, w)$ is a $|V| \times |E|$ matrix with elements defined as $h(v, e) = 1$ when $v \in e$ and 0 otherwise. The degree of a vertex v is defined as $d(v) = \sum_{e \in E} h(v, e)w(e)$, which is the sum of the weights of the hyperedges incident with v. The degree of a hyperedge e is defined as $d(e) = |\{v|v \in e\}|$, which is the number of vertices incident with e. Finally, we define W as the diagonal matrix whose elements on the diagonal are weights of hyperedges, and D_v and D_e as the diagonal matrices with elements on the diagonal being the degrees of vertices and hyperedges (the row and column sum of H). Note for a hyperedge i, $D_e(i, i) = d(i)$. However, for a vertex j, $D_v(j, j) = d(j)$ if and only if $\sum_{e \in E} h(j, e)w(e) = d(j)$.

We use a weighted hypergraph $G(V, E, w)$ to model the gene expression data: each sample is denoted by a vertex $v \in V$ and each hyperedge denotes one of the two expression states (up/down-regulated) of a gene (Figure 1A). Thus, each gene will be associated with two hyperedges in the hypergraph. The incidences between the V and E are decided by the gene expression values on the samples. If the expression value of a gene i is positive for sample set V_1 and negative on sample set V_2, the up-state hyperedge e_i^{up} is incident with V_1 and the down-state hyperedge e_i^{down} is incident with V_2. Note that $V_1 \cup V_2$ is a proper subset of V if the expression levels of the gene in some of the samples are zero (basal). After the hypergraph is constructed, we define a function y to assign initial labels to the corresponding vertices in the hypergraph. If a vertex v is in the positive group, $y(v) = +1$; If it is in the negative group, $y(v) = -1$ and if v is a test sample, $y(v) = 0$ (Figure 1B).

For cancer outcome prediction, our goal is to find the correct labels for the unlabeled vertices of the test samples in the hypergraph. Let f be the objective function (vector) of labels to be learned. Intuitively, there are two criteria for learning optimal f: 1) we want to assign the same label to vertices that share many incidental hyperedges in common; 2) assignment of the labels should be similar to the initial labeling y. For criteria 1), we define the following cost function,

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

If the predicted labels on the vertices are consistent with the incidences with the hyperedges, the value of $\Omega(f)$ should be minimized. For criteria 2), we directly measure the 2-norm distance between the vectors of the predicted and the original labels as follows,

$$||f - y||^2 = \sum_{u \in V} (f(u) - y(u))^2$$

2.2. Using interactions as constraints

To introduce protein interactions as prior knowledge into the hypergraph-based learning, we assume that interacting genes should receive similar weights on their associated hyperedges. We define a binary indicator δ_{ij} to capture the interaction between a pair of hyperedge e_i and e_j. The indicator $\delta_{ij} = 1$ if the two genes associated with e_i and e_j have the nearest distance k in the protein-protein interaction network, otherwise 0. The distance k picked larger than 1 can relax the definition of interaction between two genes by allowing indirect interactions through neighbors. When $k = 1$, it is reduced to measure the direct interaction between genes. In our experiments in this paper, we set $k = 2$. To assign weights to hyperedges consistent with the prior knowledge in the protein-protein interaction network, we define the following cost function over the hyperedge weights,

$$\Psi(w) = \frac{1}{2} \sum_{i,j=1}^{|E|} \delta_{i,j} \left(\frac{w(e_i)}{\sqrt{\sigma(e_i)}} - \frac{w(e_j)}{\sqrt{\sigma(e_j)}}\right)^2, \quad (2)$$

where $\sigma(e_i) = \sum_{j=1}^{|E|} \delta_{i,j}$, which is the number of hyperedges interacting with the hyperedge e_i. Minimizing $\Psi(w)$

ensures that hyperedges associated with interacting genes will get similarly weighted. When there is no prior knowledge, we can simply set $\Psi(w) = ||w||^2$.

2.3. Optimization formulation

After the prior knowledge is introduced from a protein-protein interaction network, our task is to minimize the sum of the three cost terms defined as

$$\Phi(f, w) = \Omega(f, w) + \mu||f - y||^2 + \rho\Psi(w),$$

where μ and ρ are positive real numbers. This objective can be achieved with the following optimization problem,

$$\underset{f,w}{\text{minimize}} \quad \Phi(f, w) \tag{3}$$

subject to

$$\begin{array}{ll} w(e) \geq 0 & \text{for } \forall e \in E \\ \sum_{e\in E} h(v, e)w(e) = d(v) & \text{for } \forall v \in V. \end{array}$$

The intuition of adding $\sum_{e\in E} h(v, e)w(e) = d(v)$ as constraints is to maintain the hypergraph structure. The weighting of the hyperedges should not be biased towards some samples (such as training samples) and thus, the degree of each vertex, the sum of the hyperedge weights on the vertex, should be kept the same as in the initial graph. Mathematically, these constraints also guarantee that the covariance matrix in $\Omega(f, w)$ is positive semi-definite with respect to f [22], which makes our learning problem solvable.

Let $\Delta = I - D_v^{-1/2}HWD_e^{-1}H^TD_v^{-1/2}$, where I is the identity matrix and W is the diagonal matrix with $W_{ii} = w(e_i)$. We can show $\Omega(f, w) = f^T\Delta f$ by

$$\begin{aligned}
\Omega(f, w) &= \sum_{e\in E}\sum_{u,v\in V} \frac{w(e)h(u, e)h(v, e)}{\delta(e)} \\
&\quad (\frac{f^2(u)}{d(u)} - \frac{f(u)f(v)}{\sqrt{d(u)d(v)}}) \\
&= \sum_{e\in E}\sum_{u\in V} \frac{w(e)h(u, e)f^2(u)}{d(u)} \sum_{v\in V} \frac{h(v, e)}{\delta(e)} \\
&\quad - \sum_{e\in E}\sum_{u,v\in V} \frac{w(e)h(u, e)h(v, e)}{\delta(e)} \frac{f(u)f(v)}{\sqrt{d(u)d(v)}} \\
&= \sum_{u\in V} f^2(u) \sum_{e\in E} \frac{w(e)h(u, e)}{d(u)} \\
&\quad - \sum_{e\in E}\sum_{u,v\in V} \frac{f(u)w(e)h(u, e)h(v, e)f(v)}{\sqrt{d(u)d(v)}\delta(e)} \\
&= \sum_{u\in V} f^2(u) \\
&\quad - \sum_{e\in E}\sum_{u,v\in V} \frac{f(u)w(e)h(u, e)h(v, e)f(v)}{\sqrt{d(u)d(v)}\delta(e)}.
\end{aligned}$$

HyperGene(y, H, A, α, ρ)

1 $t = 0, w_0 = 1, f_0 = y, c_0 = +\infty$

2 **do**

3 $t = t + 1$

4 Use Jacobi iteration method to find optimal f_t

$f_t = (I - \alpha D_v^{-1/2}HW_{t-1}D_e^{-1}H^TD_v^{-1/2})^{-1}y$

5 Use quadratic programming to find optimal w_t

$w_t = \text{argmin}_w\ \Omega(f = f_{t-1}, w) + \rho\Psi(w)$

subject to $Hw = diag(D_v)$ and $diag(W) \succeq 0$

6 $c_t = \Omega(f_t, w_t) + \mu||f_t - y||^2 + \rho\Psi(w_t)$

7 **while** $(c_{t-1} - c_t > \pi)$ *// if the decrease of the cost function is smaller than a threshold π, stop iterating.*

8 **return** (f_t, w_t)

Figure 2. The HyperGene algorithm.

Step three in the above derivation shows that $\Omega(f, w) = f^T\Delta f$ if and only if $\sum_{e\in E} \frac{w(e)h(u,e)}{d(u)} = 1$. The constraints $\sum_{e\in E} h(v, e)w(e) = d(v)$ for $\forall v \in V$ in equation 3 keep D_v unchanged during the optimization and thus make Δ always positive semi-definite. Finally, let A be the adjacency matrix defined on the protein-protein interaction network with $A_{ij} = \delta_{ij}$, where i and j are the indexes of hyperedges, and D be the diagonal matrix with $D_{ii} = \sum_j A_{ij}$, the optimization problem in equation (3) can be written in the following matrix form,

$$\underset{f,w}{\text{minimize}} \quad f^T\Delta f + \mu||f - y||^2 + \rho w^T(I - S)w \tag{4}$$

subject to

$$\begin{aligned} Hw &= diag(D_v) \\ diag(W) &\succeq 0, \end{aligned}$$

where $S = D^{-1/2}AD^{-1/2}$.

3. The HyperGene Algorithm

The objective function $\Phi(f, w)$ in the optimization problem defined by equation 3 is not convex in (f, w). However, our formulation contains two sub-optimization-problems, both of which are convex if we independently optimize $\Phi(f, w)$ with respect to f or w. Specifically, if we fix w to be a specific weighting w_t satisfying the constraints

$w_t \geq 0$ and $Hw_t = diag(D_v)$, the objective function $\Phi(f, w = w_t)$ is convex in f; if we fix f to be a specific labeling of the vertices f_t, $\Phi(f = f_t, w)$ is also convex in w. Thus, a local optimal solution can be found by solving the two optimizations alternatively by iteration. Our assumption is that f and w can be independently optimized and this assumption does not guarantee a global optimal solution to the optimization problem.

For solving the optimization problem in the regularization framework, the HyperGene algorithm is a two-step iterative method that alternatively finds the optimal f and w in each step. The outline of the HyperGene algorithm is given in Figure 2. The HyperGene algorithm first initializes w with a uniform weighting 1 over the hyperedges. Note that $w = 1$ is a solution to the linear system $Hw = diag(D_v)$ by definition of D_v and thus, a valid solution to Equation 3. In the first step in each iteration, HyperGene fixes w and optimizes $\Phi(f, w = w_t)$ with respect to f in the following optimization problem,

$$\underset{f}{\text{minimize}} \quad \Omega(f, w = w_t) + \mu||f - y||^2 \tag{5}$$

The cost term $\Psi(w = w_t)$ is removed from $\Phi(f, w = w_t)$ since it is a constant in the above optimization problem. In the cost term $\Omega(f, w = w_t) = f^T \Delta f$ (Equation 4), Δ is positive semi-definite given $\Omega(f, w = w_t) \geq 0$ for any f (Equation 1), which also implies that $\Omega(f, w = w_t)$ is convex in f. Therefore, we can simply take derivative with respect to f to get the optimal solution $f^* = ((1 - \alpha)I + \alpha\Delta)^{-1}y$, where $\alpha = \frac{\mu}{1+\mu}$ [22]. This is equivalent to solving the linear system $((1 - \alpha)I + \alpha\Delta)f = y$, which can be efficiently computed by Jacobi Iteration method [14].

In the second step in each iteration, the HyperGene algorithm fixes $f = f_t$ learned in the previous step to learn the optimal weighting of hyperedges w by solving the quadratic programming problem:

$$\underset{w}{\text{minimize}} \quad \Omega(f = f_t, w) + \rho\Psi(w) \tag{6}$$

subject to

$$\begin{array}{ll} w(e) \geq 0 & \text{for } \forall e \in E \\ \sum_{e \in E} h(v, e)w(e) = d(v) & \text{for } \forall v \in V. \end{array}$$

The cost $\mu||f - y||^2$ is removed from $\Phi(f, w = w_t)$ since it is a constant in the above optimization problem, and $\Omega(f = f_t, w)$ is a linear function of w (Equation 1). Since $\Psi(w) = w^T(I - D^{-1/2}AD^{-1/2})w \geq 0$ for any w (Equation 2), $I - D^{-1/2}AD^{-1/2}$ is positive semi-definite, which implies that $\Phi(f = f_t, w)$ is convex in w. In both steps, the total cost $\Phi(f, w)$ is guaranteed to be reduced until there is only very small change. Thus, our algorithm will finally stop at a small total cost. We implemented the HyperGene algorithm in MATLAB and use ILOG/CPLEX package (version 11.1) for quadratic programming.

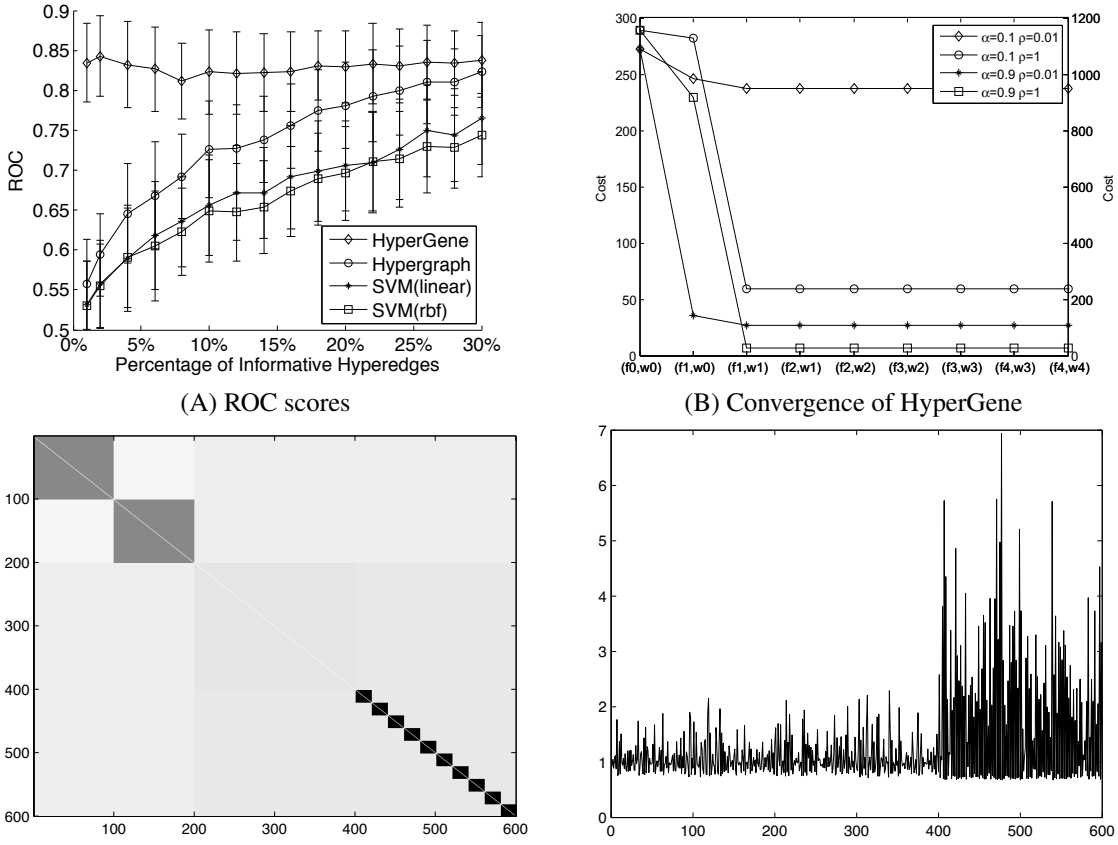
4. Experiments

We evaluate the HyperGene algorithm on both artificial datasets and two breast cancer gene expression datasets using as a prior a large curated protein-protein interaction network constructed by [4]. This protein-protein interaction network contains 57,235 interactions among 11,203 proteins integrated from yeast two-hybrid experiments, predicted interactions from orthology and co-citatioin, and other literature reviews [4]. We compare the classification performance of HyperGene with three baselines, the hypergraph-based learning algorithm [22] and SVMs with linear kernel and RBF kernel (Matlab Bioinformatics Toolbox (V3.0)). The classification performance of all methods are 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 [9].

4.1. Simulations

To mimic the noisy nature of microarray data, we test the HyperGene algorithm on artificial hypergraphs with many noisy hyperedges. In all experiments, we label 50% vertices for training and hold out the other 50% vertices in the hypergraphs for testing. We randomly generate hypergraphs with a large number of non-informative hyperedges and a certain number of special hyperedges, each of which alone is not very informative but in combination is highly informative. We first generate a highly discriminative hyperedge incident with 80% of vertices in one class and 20% of vertices in the other class, and the hyperedge is split into 5 weak informative hyperedges with equally number of vertices. The informative hyperedges are generated to simulate the expression behavior of cancer genes, which are often non-informative unless combined as a module. The prior knowledge is introduced as the interactions between the informative hyperedges and some other random interactions between non-informative hyperedges are also introduced as noise.

The algorithms are tested on 100 randomly generated such hypergraphs. We report the average ROC score of the baselines and HyperGene with different percentage of informative hyperedges in Figure 3A. Because the results are similar for different choice of ρ and α parameters, we only plot the case with $(\alpha, \rho) = (0.5, 1)$. It is clear in the plot that, when the prior knowledge gives useful information about interactions between informative hyperedges, the performance of our algorithm is significantly better than SVMs and the hypergraph-based algorithm with uniform weights. Since in this simulation, only very high-order combination of the hyperedges can provide good classification performance, SVMs perform poorly in all cases. To check the



(A) ROC scores

(B) Convergence of HyperGene

(C) Interactions among hyperedges

(D) Weights assigned to the hyperedges by HyperGene

Figure 3. Simulation on outcome prediction and marker gene discovery. (A) This plot compares the algorithms by averaged ROC scores over 100 trials. The x-axis is the percentage of informative hyperedges in the hypergraph. We set $(\alpha, \rho) = (0.5, 1)$ for the HyperGene algorithm. (B) This plot shows the decrease of the cost function after each iteration of HyperGene. (C) There are 10 small interacting modules among 200 informative edges and 2 larger interacting modules among non-informative modules. (D) The x-axis is the index of the hyperedges aligned with the indexes in plot (C). The y-axis is the weights. The 200 informative edges are assigned larger weights by HyperGene.

Table 1. Performance of cancer outcome prediction. On the van't Veer *et al* dataset, the ROC score on the 19-patient test set is reported. On the van de Vijver *et al* dataset, over the random 5-fold cross-validations (100 times on the 325 genes and 50 times on the 1,495 genes), the mean and standard deviation of the ROC scores and the pairwise comparison of classification performance between HyperGene and the baseline algorithms in each experiment are reported.

Algorithms	van't Veer *et al* dataset		van de Vijver *et al* dataset			
	231 genes	500 genes	325 cancer genes		1,465 cancer genes	
	ROC	ROC	Mean (Std)	Win/Tie/Loss versus HyperGene	Mean (Std)	Win/Tie/Loss versus HyperGene
SVM (linear)	0.845	0.845	0.679 (0.064)	163/60/277	0.675 (0.063)	79/29/142
SVM (rbf)	0.833	0.845	0.686 (0.063)	140/75/285	0.679 (0.065)	78/22/150
Hypergraph	0.857	0.821	0.685 (0.062)	83/77/340	0.684 (0.066)	45/68/137
HyperGene	**0.893**	**0.869**	**0.699** (0.061)	0/500/0	**0.691** (0.064)	0/500/0

convergence of the HyperGene algorithm, we also measure the value of the cost function in each iteration on the two real microarray gene expression datasets with selected 1,465 genes (see section 4.2). The change of the cost function for different α and ρ parameters is shown in Figure 3B. It is clear that the HyperGene algorithm converges very fast. We also found that the value of f and w variables stay unchanged after the first 2 to 3 iterations.

To test if the HyperGene algorithm can select informative hyperedges, we design one additional experiment with more diverse prior knowledge on both informative hyperedges and non-informative hyperedges. We generate 400 non-informative hyperedges and 200 informative hyperedges. The 200 informative hyperedges are grouped into 10 fully connected cliques in the interaction network. We also group 200 random hyperedges into 2 fully connected cliques. The adjacency matrix is shown in Figure 3C. The 2 cliques of non-informative hyperedges are on the top-left of the matrix and the 10 cliques of informative hyperedges are on the bottom-right of the matrix. The weights learned by HyperGene is plotted in Figure 3D. It is evident that informative hyperedges are assigned much larger weights, which shows that the HyperGene algorithm is capable of selecting true informative interaction components even under the presence of abundant irrelevant interactions. This result also suggests that the HyperGene algorithm assigns weights to hyperedges based on both the predictability and modularity of the hyperedges, instead of the number of interactions that they have in the interaction network. Accordingly, the HyperGene algorithm achieves the highest ROC score 0.874 in this experiment, while the hypergraph-based algorithm and SVM with linear kernel and RBF kernel only score 0.750, 0.596 and 0.604 respectively.

4.2. Cancer outcome prediction on breast cancer datasets

We next test the HyperGene algorithm for cancer outcome prediction on two breast cancer gene expression datasets, the van't Veer *et al* dataset [18] and the van de Vijver *et al* dataset [10]. The van't Veer *et al* dataset and the van de Vijver *et al* dataset contain 24,481 gene expressions of 97 and 295 patients respectively. The patients are divided into two groups based on whether the patient had been free of disease after their diagnosis for an interval of at least 5 years or had developed distant metastasis within 5 years after a poor prognosis. The details for quantization and normalization of scanned microarray images are described in [18, 10]. In the experiments on the van't Veer *et al* dataset, two subsets of gene expressions, 231 genes suggested by [18] and the top ranked 500 genes selected by the correlation coefficients between the gene expressions and the cancer outcomes, are used for classification. Note that the two subsets of genes are selected on a training set of 78 patients and the remaining 19 patients are held out as the test set as suggested by [18]. In the experiments on the van de Vijver *et al* dataset [10], we use for classification two subsets of hypothetical cancer susceptibility genes, 326 genes from Ingenuity[1] and 1,465 genes from Cancer Genomics tool[2]. We randomly run 5-fold cross-validation multiple times on the van de Vijver *et al* dataset and measure the average ROC. Note that within each experiment of a 5-fold cross-validation, another 4-fold cross-validation is used on the training set to pick the best parameters for HyperGene and the baseline algorithms to test the held-out set. The classification results in Table 1 show that HyperGene performs significantly better than both SVMs and the hypergraph-based learning algorithm in all the experiments. Particularly, HyperGene outperforms the three baseline algorithms in classifying the 19 test samples on the van't Veer *et al* dataset by around 4 to 6 percents when the 231 genes are used, and around 2.5 to 5 percents when the 500 genes are used. It is interesting that the optimal values of ρ for HyperGene in the two experiments are both 1. When ρ is large, the prior knowledge from the protein-protein interaction network is emphasized, and the interacting genes will get very similar weights in the optimizations of the HyperGene algorithm. When the interaction network contains accurate and helpful information, larger ρs will be picked in cross-validation to take advantage of the prior knowledge. However, when the quality of the interaction network is poor, larger ρs will lead to deteriorated classification performance. Thus, we speculate that the protein-protein interaction network plays important role in learning the better classifiers, given the relatively large value for ρ. On the van de Vijver *et al* dataset, HyperGene achieves an improvement of 1.3 to 2.4 percents on the average ROC score. Although the improvement seems to be small, pairwise comparisons between HyperGene and the baseline algorithms show that in many more cases, HyperGene outperforms the other algorithms.

4.3. Breast cancer biomarker identification

4.3.1 Identification of known biomarkers

To demonstrate that HyperGene is capable of identifying true cancer susceptibility genes, we examine the weights of genes obtained by the HyperGene algorithm. In this experiment, we construct a hypergraph with the 1,465 candidate cancer genes and all the labeled patient vertices on the van de Vijver *et al* dataset. We compare the genes that are highly weighted by HyperGene with those known breast cancer causative genes reported in previous literatures. We collect a list of 30 breast cancer causative genes,

[1] http://www.ingenuity.com/

[2] http://cbio.mskcc.org/cancergenes/Select.action

16 of which are presented in our data, from [18] and the overview section of breast cancer (MIM 114480) in Online Mendelian Inheritance in Man (May, 2007)[3]. While Correlation Coefficients give very low ranking to the 16 known breast cancer causative genes, the HyperGene algorithm in two different settings (ρ = 1 and 0.001) assigns high ranks to most of the genes, with 14 out of 16 genes ranked in the top 300 genes (Table 2). The difference of the ranking of known breast cancer causative genes calculated in the two ρ values is small, which indicates that the Hypergene algorithm is not sensitive to ρ parameter to identify marker genes in this case. Notable examples of the biomarker genes are tumor protein p53 (TP53), estrogen receptor 1 (ESR1), v-Ha-ras Harvey rat sarcoma viral oncogene homolog (HRAS), and v-Ki-ras2 Kirsten rat sarcoma viral oncogene homolog (KRAS), all of which are not identified in [18] but are highly ranked by the HyperGene algorithm. Other novel susceptibility candidates that are not in our list of known causative genes but have a large number of interactions with known susceptibility genes such as CREB binding protein (CREBBP), B-cell CLL/lymphoma 2 (BCL2), and Mdm2 p53 binding protein homolog (MDM2) are also highly ranked by the HyperGene algorithm.

Table 2. The ranking of known breast cancer susceptibility genes. We compare the ranking of the known cancer genes obtained by the HyperGene algorithm with the ranking calculated by Correlation Coefficients (CC). We set $\alpha = 0.5$ and $\rho = 1$ and 0.001 to test the HyperGene algorithm.

Known Disease Gene	Gene Ranking		
	HyperGene α=0.5, ρ=1	HyperGene α=0.5, ρ=0.001	CC
TP53	1	2	601
BRCA1	14	19	629
ESR1	17	22	208
BARD1	51	72	562
ATM	75	77	1054
HRAS	96	81	437
AKT1	99	154	1024
TGFB1	130	152	760
CASP8	142	201	1221
PTEN	157	198	725
PPM1D	182	60	266
KRAS	183	257	1267
SERPINE1	207	118	973
BRCA2	227	299	924
PIK3CA	415	363	712
STK11	632	609	773

4.3.2 Functional enrichment and pathway analysis

We also analyze the biological functions of the biomarker genes by Gene Ontology (GO) annotations and pathway analysis with Ingenuity (version 5.5). We investigate whether the identified marker genes involve significantly over-represented GO categories and biological pathways that are related with breast cancer. With the top 100 marker genes as input, Ingenuity identifies 17 enriched functions scoring a p-value less than $1.0e-9$ on the van de Vijver *et al* dataset. Figure 4 shows the enriched biological functions from the van de Vijver datasets. All the 17 enriched functions of top 100 marker genes shows strong consistency with those identified by [5, 19], indicating that these processes are significantly involved with the progression of cancer. Especially, the most significant functions such as cell cycle (p-value = $4.03e-47$), cell death (p-value = $3.44e-44$) , gene expression (p-value = $2.43e-43$), and cellular growth and proliferation (p-value = $2.7e-36$) are well known to be functionally involved with metastasis and development of breast cancer [15, 19, 4, 18]. Note that among the 17 functions, 11 functions are closely or exactly matched with the 21 functions discovered previously in [19].

In Figure 5, we show the identified sub-networks among the top 100 marker genes. The genes in the same protein complex or biochemical pathway tend to perform similar biological functions and may lead to same or similar diseases [12, 8]. As shown in Figure 5, many known causative cancer genes play critical roles and are present with other susceptibility candidate genes in the pathway networks. TP53-subnetwork is involved with glucocorticoid receptor signaling, p53 signaling and B cell receptor signaling pathways, and BRCA1-subnetwork is over-represented with glucocorticoid receptor signaling, estrogen receptor signaling, and RAR activation. Other networks are also involved with glucocorticoid receptor signaling, RAR activation, estrogen receptor signaling and other canonical pathways. All those over-represented biological pathways are closely linked with breast cancer [4]. This observation again supports the hypothesis that cancer genes share specific pathways involved with disease and they often interact with each other in a protein-protein interaction network [3, 19, 4, 8].

5. Conclusion

Utilizing the prior knowledge introduced from a protein-protein interaction network, the HyperGene algorithm outperforms SVMs and the original hypergraph-based learn-

[3] http://www.ncbi.nlm.nih.gov/omim/

[4] http://cgap.nci.nih.gov/

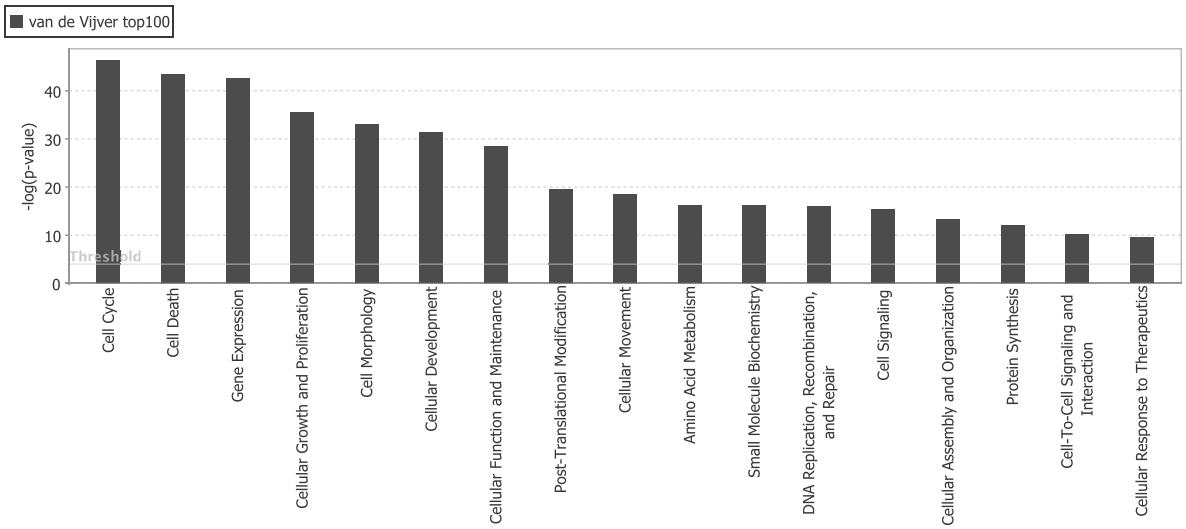
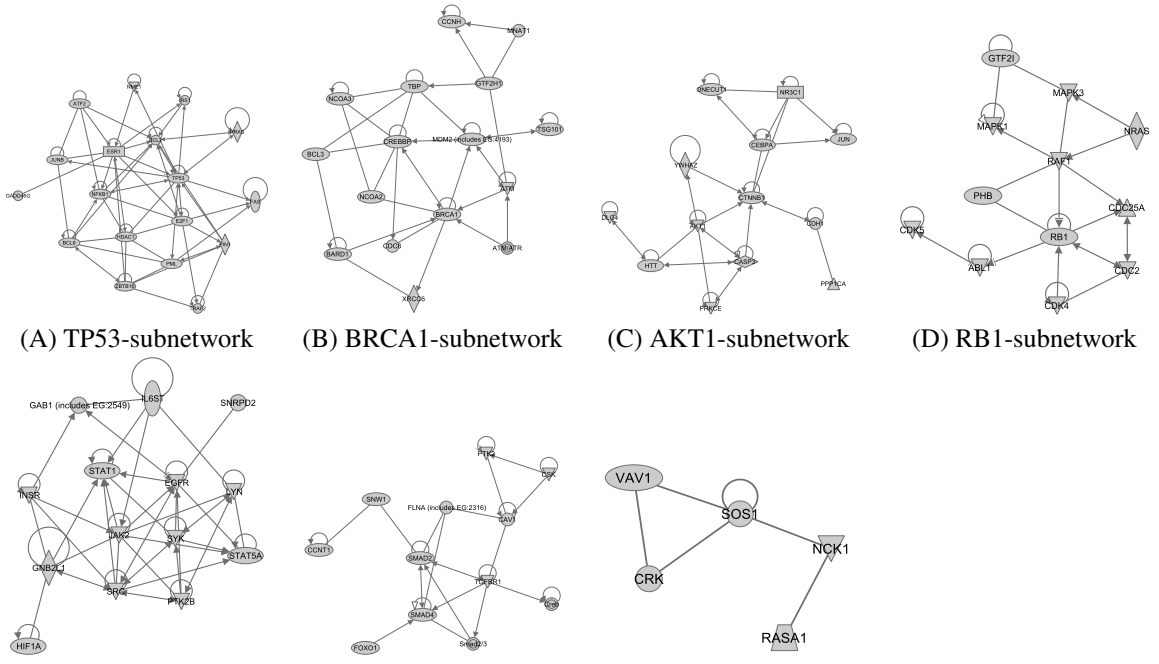


Figure 4. Enriched biological functions by the top 100 marker genes on the van de Vijver *et al* dataset. The enriched functions are sorted by p-values calculated using the right-tailed Fisher Exact Test. All the enriched functions have p-value less than $1.0e-9$.



(A) TP53-subnetwork (B) BRCA1-subnetwork (C) AKT1-subnetwork (D) RB1-subnetwork

(E) STAT1-subnetwork (F) SMAD2-subnetwork (G) SOS1-subnetwork

Figure 5. Seven Interaction networks of the top 100 marker genes on van de Vijver *et al* dataset. Known breast cancer causative genes such as TP53, ESR1 and BRCA1 play a central role in the networks. Other known susceptibility genes such as v-akt murine thymoma viral oncogene homolog 1 (AKT1), retinoblastoma 1 (RB1), signal transducer and activator of transcription 1, 91kDa (STAT1), SMAD family member 2 (SMAD2), and son of sevenless homolog 1 (SOS1) also tend to be hubs and interact with many other susceptibility genes in the networks. Note that we remove those marker genes that do not directly interact with other known susceptibility genes.

ing algorithm in experiments on both artificial datasets and two real breast cancer datasets. HyperGene is also capable of retrieving maker genes highly relevant to the cancer. Thus, HyperGene is an effective algorithm to integrate gene expressions and protein-protein interactions for cancer outcome prediction and biomarker identification.

As large volume of human genomic and proteomic data is becoming available for cancer studies, data integration for improving cancer prognosis and treatment is turning into one of the central problems in biomedical research. Our results suggest that large scale protein-protein interaction networks contain complementary information that can potentially aid cancer outcome prediction and biomarker identification with microarray gene expression data. The HyperGene algorithm is a powerful tool for handling this data integration problem.

We plan to extend the HyperGene algorithm to handle other types of prior knowledge such as Gene Ontology or pathways for learning with microarray gene expressions. The other prior knowledge might indicate other types of priors on the genes, which will need to be handled differently with variants of HyperGene. We also plan to apply the HyperGene algorithm to study other cancers such as lung cancer to improve the diagnosis and prognosis of these cancers.

Acknowledgments

Ze Tian is supported by the Biomedical Informatics and Computational Biology (BICB) Graduate Traineeship Program at University of Minnesota.

References

[1] S. Agarwal, K. Branson, and S. Belongie. Higher order learning with graphs. In *ICML '06: Proceedings of the 23rd international conference on Machine learning*, pages 17–24, 2006.

[2] S. Agarwal, J. Lim, L. Zelnik-Manor, P. Perona, D. Kriegman, and S. Belongie. Beyond pairwise clustering. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, pages 838–845. IEEE Computer Society, 2005.

[3] R. Aragues, C. Sander, and B. Oliva. Predicting cancer involvement of genes from heterogeneous data. *BMC Bioinformatics*, 9(172), 2008.

[4] H. Y. Chuang, E. Lee, Y. T. Liu, D. Lee, and T. Ideker. Network-based classification of breast cancer metastasis. *Mol Syst Biol*, 3, 2007.

[5] R. W. D. Hanahan. The hallmarks of cancer. *Cell*, 100:57–70, Jan 2000.

[6] A. Dupuy and R. M. Simon. Critical review of published microarray studies for cancer outcome and guidelines on statistical analysis and reporting. *J Natl Cancer Inst.*, 99(2):147–57, 2007.

[7] G. V. Glinsky, A. B. Glinskii, A. J. Stephenson, R. M. Hoffman, and W. L. Gerald. Gene expression profiling predicts clinical outcome of prostate cancer. *J. Clin. Invest.*, 113:913–923, 2004.

[8] K. I. Goh et al. The human disease network. *Proceeding National Academic Science USA*, 104(21):8685–8690, 2007.

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

[10] P. Helmbold, J. Haerting, H. Klbl, D. B. Kopans, I. H. Kunkler, D. F. Ransohoff, M. J. van de Vijver, Y. D. He, L. J. van't Veer, and R. Bernards. A gene-expression signature as a predictor of survival in breast cancer. *N Engl J Med*, 347:1999–2009, 2002.

[11] A. R. Mendelsohn and R. Brent. Protein interaction methods–toward an endgame. *Science*, (284):1948–50, 1999.

[12] M. Oti and H. Brunner. The modular nature of genetic diseases. *Clinical Genetics*, 71:1–11, 2007.

[13] F. Rapaport, A. Zinovyev, M. Dutreix, E. Barillot, and J. P. Vert. Classification of microarray data using gene networks. *BMC Bioinformatics*, (35), 2007.

[14] Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company, Boston, MA, 1996.

[15] C. Sotiriou et al. Gene expression profiling in breast cancer: Understanding the molecular basis of histologic grade to improve prognosis. *Journal of the National Cancer Institute*, 98:262–272, 2006.

[16] Z. Sun and P. Yang. Gene expression profiling on lung cancer outcome prediction: Present clinical value and future premise. *Cancer Epidemiology Biomarkers and Prevention*, pages 2063–2068, 2006.

[17] K. Tsuda. Propagating distributions on a hypergraph by dual information regularization. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 920–927. ACM, 2005.

[18] L. J. van't Veer, H. Dai, M. J. van de Vijver, S. H. Friend, and etc. Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 415:530–536, 2002.

[19] Y. Wang, J. Klijn, Y. Zhang, A. Sieuwerts, M. Look, F. Yang, D. Talantov, M. Timmermans, M. M. van Gelder, and J. Yu. Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer. *The Lancet*, 365:671–679, 2005.

[20] H. Xiong, X. He, C. Ding, Y. Zhang, V. Kumar, and S. R. Holbrook. Identification of functional modules in protein complexes via hyperclique pattern discovery. In *PSB '05: Proceedings of the 2005 Pacific Symposium on Biocomputing*, pages 221–232, 2005.

[21] H. Xiong, P. N. Tan, and V. Kumar. Mining strong affinity association patterns in data sets with skewed support distribution. In *ICDM '03: Proceedings of the Third IEEE International Conference on Data Mining*, pages 387–394, Melbourne, Florida, USA, 2003.

[22] D. Zhou, J. Huang, and B. Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. *NIPS*, 19:1633–1640, 2006.