Statistics and Its Interface Volume 2 (2009) 257–269

Support vector machines with disease-gene-centric network penalty for high dimensional microarray data

Yanni Zhu[†], Wei Pan[*†], and Xiaotong Shen[‡]

With the availability of gene pathways or networks and accumulating knowledge on genes with variants predisposing to diseases (disease genes), we propose a disease-gene-centric support vector machine (DGC-SVM) that directly incorporates these two sources of prior information into building microarray-based classifiers for binary classification. DGC-SVM aims to detect genes clustering together and around some key disease genes in a gene network. Toward this end, we propose a penalty over suitably defined groups of genes. A hierarchy is imposed on an undirected gene network to facilitate the definition of such gene groups. Our proposed DGC-SVM utilizes the hinge loss penalized by a sum of the L_∞-norm over each group. The simulation studies show that DGC-SVM not only detects more disease genes along pathways than the existing standard-SVM and SVM with an L_1-penalty (L1-SVM), but also captures disease genes that potentially affect the outcome only weakly. Two real data applications demonstrate that DGC-SVM improves gene selection while retaining predictive performance of the standard-SVM and L1-SVM. The proposed method has the potential to be an effective classification tool that encourages gene selection along paths to or clustering around known disease genes for microarray data.

Keywords and phrases: DAG, Gene expression, Gene network, Grouped penalty, Hierarchy, Penalization.

1. INTRODUCTION

Genes interact with each other through their RNA and protein expression products. For example, the rate at which transcription factor genes are transcribed into RNA molecules may govern the transcriptional rate of their regulatory target genes, which as a result become either up- or down- regulated. A gene network is a collection of effective interactions, describing the multiple ways through which one gene affects all the others to which it is connected. A gene network reveals genetic dynamics underlying the aggregate function that the network maintains.

*Corresponding author.
†NIH grants HL65462 and GM081535.
‡NIH grant GM081535 and NSF grants IIS-0328802 and DMS-0604394.

High-throughput genomic advances have generated various databases providing gene network information, such as the Biomolecular Interaction Network Database (BIND) [1], the Human Protein Reference Database (HPRD) [15], and the Kyoto Encyclopedia of Genes and Genomes (KEGG) [11]. In recent years, genetic studies have uncovered hundreds of genes with variants that predispose to common diseases, such as cancer, Parkinson's disease, and diabetes. For example, gene *TP53* is among the most well-known ones, which, as a tumor suppressor, is central to many anti-cancer mechanisms. Gene *TP53* encodes tumor protein *p53*, the so-called "the guardian of the genome," which mediates the cellular response to DNA damage and is involved in other important biological processes, e.g., cell cycle. Among its other functions, *p53* activates other genes to fix the damage if *p53* determines that the DNA can be repaired. Otherwise, *p53* prevents the cell from dividing and signals its death. Most mutations that deactivate *TP53* destroy protein *p53*'s ability to regulate other genes properly and thus leads to increasing risk of tumor development ([4, 17]). Hence, not just a single gene, but a subnetwork of *TP53* and its interacting partners, are involved in the disease progression.

With the availability of various repositories of gene networks and the accumulating knowledge on genes linked to diseases, one question naturally arises: how to integrate the two sources of prior information into a model to detect genes involved in disease-related biological processes. A network-based approach is based on such a coherent view and makes use of the network information in building statistical models. Employing a network-based perspective not only sheds insight within the network modules ([2, 6, 8, 12]) but also permits identification of disease genes that have only weak effects. Such genes often play a central role in discriminative subnetworks by interconnecting groups of genes involved in various biological processes. [8] pointed out that several well-known cancer genes, such as *TP53*, *KRAS*, and *HRAS*, were ignored by gene-expression-alone analysis but successfully detected by using network information. However, their network-based approach involves a random search over subnetworks, leading to possibly instable and suboptimal final results.

Since its invention ([7, 18]), the support vector machine (SVM) has been acclaimed as a useful regularization method

due to its excellent empirical performance, especially for high dimensional data ([5, 10]), its possible extensions to accommodate various penalty functions, and resulting model sparsity if a suitable penalty (e.g. L_1-norm) is employed. For binary classification, the standard $L2$-norm SVM (STD-SVM) has good predictive performance, but is incapable of performing variable selection. The L1-SVM ([19, 23]) produces sparse models for data with $p >> n$. [25] developed a grouped variable selection scheme for factors by the use of an F_∞-norm SVM such that all features derived from the same factor (i.e. categorical predictor) are included or excluded simultaneously. Note that their grouping scheme was based on non-overlapping groups. [22] generalized grouped variable selection and introduced the composite absolute penalties (CAP) family. CAP achieves both grouped selection for non-overlapping groups and hierarchical selection for overlapping groups. Extending the idea of grouping to gene networks, [24] proposed a network-based SVM (NG-SVM), treating any two neighboring genes in a network as one group, and explicitly incorporating the network information into building classifiers. Both the simulation studies and real data applications showed that NG-SVM enjoyed advantages in gene selection and predictive performance compared with the popular STD-SVM and L1-SVM. However, a potential problem of NG-SVM resides in its tendency of selecting isolated genes or gene pairs, i.e., genes largely disconnected to each other in the network, which is not desirable given that some disease genes cluster together and form subnetworks.

In this paper, we incorporate the information of both a gene network and some crucial disease genes into the SVM framework by exploiting two ways of grouping genes for penalty construction. By considering an undirected network to be anchored on certain crucial disease gene(s), i.e., genes known to be central to a disease, a hierarchical structure is imposed on the network (with the anchoring crucial genes at the top) to facilitate the definition of various gene groups. By summing up an L_∞-norm over each group, we obtain the penalty for DGC-SVM. Ideally, by DGC-SVM, identification of one gene triggers the inclusion of disease genes along the connected paths towards the top crucial gene(s). In particular, we intend to capture disease genes, even if their direct effects on the outcome are weak, which are important in regulating functional activities of other genes along the pathways or within the subnetworks involved in the disease.

2. METHODS

2.1 Orienting an undirected network

Given an undirected network G, we convert it into a directed acyclic graph (DAG) $\tilde{G}$. Suppose that G originates from only one disease gene g and consists of a total of p genes. Genes (including g) in network G are indexed by $\{1, 2, \ldots, p\}$. We have the expression levels of the p genes and binary outcomes for N samples, $\{(x_i, y_i)\}_{i=1}^N$ with $x_i \in \mathbb{R}^p$ and $y_i \in \{1, -1\}$. The expression of each gene is normalized to have mean 0 and standard deviation 1 across samples. We define a directed edge by an ordered pair of ends (a, b) indicating that a is upstream to b, or equivalently, b is downstream to a. Since genetic interrelationships occur only between pairs of distinct genes, network G contains no loop, defined as a directed edge with identical ends. In addition, no two directed edges adjoin the same pair of genes. Gene g is the top (center) gene of network G. The distance between two genes a and b is the minimum number of directed edges traversed from a to b. Genes closer to the network origin, gene g, are said to be at an upper level than those farther apart. Genes with the same distance from the origin are at the same level. For example, the distance between gene g and any of its direct neighbors is 1. The distance between any two genes at the same level is 0. Thus, DAG $\tilde{G}$ is defined from the undirected network G. $\tilde{G}$ assigns directions from upper-level to lower-level genes but ignores edges connecting genes at the same level. Upper-level genes are called nodes whereas genes with no downstream genes are named as leaves. DAG $\tilde{G}$ captures the upper-lower interrelationships but ignoring the lateral ones.

In presence of multiple center genes $g_1 \ldots g_L$, DAG $\tilde{G}$ can be defined as follows: (1) Derive DAGs $\tilde{G_1} \ldots \tilde{G_L}$, each corresponding to one center gene in $g_1 \ldots g_L$; (2) $\tilde{G} = \cup_{l=1}^L \tilde{G_l}$ if $\tilde{G_1} \ldots \tilde{G_L}$ share no common nodes; (3) if the DAGs have common nodes, pick up any of them, align all the associated DAGs at the level where that common node is seated, treat that node in each associated DAG as being located at the same level (named as $level_v$), and merge the associated DAGs by recognizing only the upper-lower interrelationships but ignoring the lateral ones. Then, identify the common nodes of the merged DAG and the remaining untouched DAGs, repeat step (3) until no common nodes exist. Note that each node in the merged DAG has the same downstream genes no matter which center the node is derived from. The above process may result in different DAGs if the combination of the associated DAGs occurs at different common nodes, introducing certain arbitrariness.

2.2 Pathway grouping

To achieve our goal of detecting collectives of genes involved in disease along pathways or within subnetworks, we propose a penalty on suitably defined groups of genes. We experiment two ways of grouping: pathway (PW) grouping and partial tree (PT) grouping. We first describe the PW grouping. It forms groups along linear paths as an attempt to encourage linear pathway selection.

A path in $\tilde{G}$ is a connected sequence of directed edges and the length of the path is the number of directed edges traversed. Note that a path connects genes from upper- to lower-levels without any two consecutive genes from the same level. Since a path can be determined by the sequence of the nodes along the path, a path is simply specified by

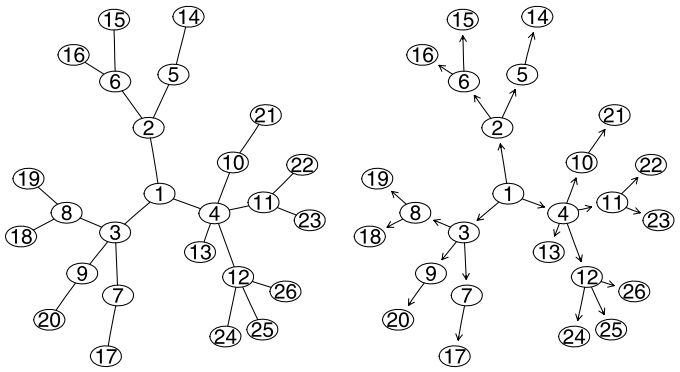

Figure 1. Left: a simple network originating from gene 1. Right: DAG derived from the simple network.

its node sequence. We define a single node as a trivial path. Define a complete path of leaf k in $\tilde{G}$, $\mathcal{E}_k$ $(k = 1, \ldots, K)$, as

$$\mathcal{E}_k = \{j : \text{ Gene } j \text{ appears on the path from top gene } g \text{ down to leaf } k\}.$$

Suppose $\mathcal{E}_k$ contains a total of n_k genes, including leaf k and gene g. Then we have n_k groups $\mathcal{G}_t^{(k)}$ $(t = 1, \ldots, n_k)$ by grouping the genes in $\mathcal{E}_k$ under the "lower nested within upper" rule, that is, node/leaf at a lower level must appear in all the groups that contain any node at its upper-level. For example, in the network displayed in Fig. 1, if gene 1 is considered to be at the top, then genes $1, \ldots, 12$ are nodes and genes $13, \ldots, 26$ are leaves. The complete path of leaf gene 16, $\mathcal{E}_{16}$, is $\{1, 2, 6, 16\}$. Groups derived from $\mathcal{E}_{16}$ or leaf gene 16 are $\{2, 6, 16\}$, $\{6, 16\}$, $\{16\}$, and $\mathcal{E}_{16}$ itself. Note that multiple distinct complete paths may exist between leaf k and gene g, for example, $\{g, a, c, k\}$ and $\{g, b, c, k\}$. In this case, group $\{c, k\}$ and group $\{k\}$ are defined twice respectively. When forming groups, we count each distinct group only once. Therefore, groups formed from $\{g, a, c, k\}$ and $\{g, b, c, k\}$ include 6 groups: $\{g, a, c, k\}$, $\{g, b, c, k\}$, $\{a, c, k\}$, $\{b, c, k\}$, $\{c, k\}$, and $\{k\}$. Thus, we impose a grouping structure $\mathcal{G}$ containing distinct groups of $\tilde{G}$, that is, every group in $\mathcal{G}$ appears only once:

$$\mathcal{G} = (\mathcal{G}_1^{(1)}, \ldots, \mathcal{G}_{n_1}^{(1)}, \ldots, \mathcal{G}_1^{(K)}, \ldots, \mathcal{G}_{n_K}^{(K)}),$$

while a gene may appear in multiple groups, which is permitted in our formulation and computation.

Corresponding to $\mathcal{G}$, we construct our penalty as

$$\text{(1)} \qquad \left(\sum_{k=1}^{K} \sum_{t=1}^{n_k} \|\beta_{\mathcal{G}_t^{(k)}}\|_\infty \right).$$

The hinge loss penalized by (1) leads to our proposed DGC-SVM with PW grouping (DGC-SVM-PW), which is developed as an attempt to encourage selecting genes along the pathway (pathway selection):

$$\text{(2)} \qquad \min_{\beta_0, \beta} \left\{ \sum_{i=1}^{N} \left[1 - y_i \left(x_i^T \beta + \beta_0\right)\right]_+ + \lambda \left(\sum_{k=1}^{K} \sum_{t=1}^{n_k} \|\beta_{\mathcal{G}_t^{(k)}}\|_\infty \right) \right\},$$

where the subscript "+" denotes the positive part, i.e., $z_+ = max\{z, 0\}$, λ is a tuning parameter, and $\|\beta_{\mathcal{G}_t^{(k)}}\|_\infty = \max_{s \in \mathcal{G}_t^{(k)}} \{|\beta_s|/w_s\}$ with w_s as a weight function for gene s. For example, w_s can be $\sqrt{d_s}$ with d_s as the number of direct neighbors of gene s, or d_s, or simply 1 for all genes. The solution to (2) can be obtained through linear programming:

$$\text{(3)} \qquad \min_{\beta_0^+, \beta_0^-, \beta^+, \beta^-, \{M_{\mathcal{G}_t^{(k)}}\}} \left(\sum_{i=1}^{N} \xi_i + \lambda \sum_{k=1}^{K} \sum_{t=1}^{n_k} M_{\mathcal{G}_t^{(k)}} \right)$$

subject to

$$\text{(4)} \qquad \begin{aligned} & y_i \left(\beta_0^+ - \beta_0^- + x_i^T(\beta^+ - \beta^-)\right) \geq 1 - \xi_i, \\ & \xi_i = \left[1 - y_i \left(x_i^T \beta + \beta_0\right)\right]_+ \geq 0, \quad \forall i, \\ & \frac{\beta_s^+}{w_s} + \frac{\beta_s^-}{w_s} \leq M_{\mathcal{G}_t^{(k)}}, \quad \beta_s^+ \geq 0, \quad \beta_s^- \geq 0, \\ & s \in \mathcal{G}_t^{(k)}, \quad k = 1, \ldots, K, \quad t = 1, \ldots, n_k. \end{aligned}$$

In the above parametrization in (4), $M_{\mathcal{G}_t^{(k)}} = \max_{s \in \mathcal{G}_t^{(k)}} \{|\beta_s|/w_s\}$, and $\beta_s = \beta_s^+ - \beta_s^-$, in which β_s^+ and β_s^- denote the positive and negative parts of β_s. Note that, by our construction, some genes fall within multiple groups $\mathcal{G}_t^{(k)}$, which is permitted by linear programming.

2.3 Partial tree grouping

The PT grouping is devised to achieve hierarchical selection, that is, selecting a lower-level gene ensures the selection of its upper-level gene(s). In addition, selecting any gene in the DAG guarantees the inclusion of at least one center gene, which is desirable in view of the biological importance of any center gene. The DGC-SVM with PT grouping (DGC-SVM-PT) groups each node/leaf with all its downstream genes. Since a leaf has no downstream genes, the group derived from the leaf contains only one element, the leaf itself. For the above $\tilde{G}$, we have p groups in total, K of which contain only single elements derived from K leaves, and the rest $p - K$ of which are formed as

$$\mathcal{G}_q = \{\text{node } q \text{ and all its downstream genes}, \quad q = 1, \ldots, p - K\}.$$

For example, the simple network in Fig. 1 derives 26 groups, including $\mathcal{G}_1$ with all the 26 genes as well as 14 single-leaf groups. Here we impose a grouping structure as $\mathcal{G} = (\mathcal{G}_1, \ldots, \mathcal{G}_p)$. The formulation of DGC-SVM-PT is the same as its PW grouping counterpart in (2)–(4).

The DGC-SVM-PT is a direct application of the CAP family of [22] in the context of SVM. It has the hierarchical property that if any node/leaf at a lower-level is included

in the model, nodes at its upper-level will also be included. This property is important to our goal of capturing disease genes along pathways or within subnetworks, which offers the possibility to detect genes that have weak direct effect on the outcome but are critical in regulating multiple biological processes through connecting various functional groups of disease-relevant genes. In addition, this property guarantees the identification of a center gene of the network if any gene in the network is selected.

2.4 Choice of weight

DGC-SVM involves a weight function w. The choice of the weight depends on the goal of shrinkage, and governs variable selection and predictive performance.

A main motivation behind the proposed penalties is the *grouping* effect of the L_∞-norm. Because of the singularity of the penalty $\max(|a|, |b|)$ at $|a| = |b|$, by [9], the penalty encourages the shrinkage towards $|a| = |b|$, which can be achieved when the penalization parameter λ is large enough. For linear regression, this so-called grouping effect has been theoretically established by [3] and [14] for two-gene groups, and by [21] for a more general case with more than two genes in a group. Now consider network G and its grouping structure $\mathcal{G}$ derived from $\tilde{G}$. For simplicity, we assume that $\mathcal{G}$ contains only two-gene and one-gene groups. For these two-gene groups, the weighted penalty encourages $|\beta_{j_1}|/w_{j_1} = |\beta_{j_2}|/w_{j_2}$ where β_{j_1} and β_{j_2} belong to the same group. Here we examine three weight functions specifically: $w_s = 1$, $w_s = \sqrt{d_s}$, and $w_s = d_s$, where d_s is the degree of gene s, i.e., the number of direct neighbors of gene s. The new method encourages $|\beta_{j_1}| = |\beta_{j_2}|$ if $w_s = 1$, $\frac{|\beta_{j_1}|}{\sqrt{d_{j_1}}} = \frac{|\beta_{j_2}|}{\sqrt{d_{j_2}}}$ if $w_s = \sqrt{d_s}$, and $\frac{|\beta_{j_1}|}{d_{j_1}} = \frac{|\beta_{j_2}|}{d_{j_2}}$ if $w_s = d_s$. The same reasoning also applies to groups with more than two genes. Therefore, larger weights (from $w_s = 1$, $w_s = \sqrt{d_s}$, to $w_s = d_s$) favor genes with more direct neighbors to have larger coefficient estimates; in other words, larger weights relax the shrinkage effect for those "hub" genes that are connected to many genes and are known to be biologically more important. Because of this property, the choice of a large weight, as a simple strategy, enables us to alleviate the bias in the coefficient estimates from penalization and possibly improve predictive performance. The weight can be considered as a tuning parameter and estimated by cross-validation or an independent tuning data set although we will not pursue it further here. Since the proposed penalty is linear, linear programming is applicable to solve the associated optimization problem. We implemented the proposed method through a linear programming routine `lpsolve` in R.

3. SIMULATION

We numerically evaluated the new methods, DGC-SVM-PW and DGC-SVM-PT, in two simulation studies over a simple network and a more complex one. The DAG for a simple network is essentially a hierarchical tree where any two genes are connected by a unique path. In contrast, there exist multiple paths adjoining the same pair of genes in the complicated network. The grouping structure in either case is unique. We compare the performance of DGC-SVMs with that of STD-SVM, L1-SVM, and NG-SVM. All methods were implemented by the R package `lpsolve` except STD-SVM, which was obtained by the R package `e1071` (with linear kernel).

3.1 A simple network

We applied the DGC-SVM to the simple network depicted in Fig. 1, where any two genes in its DAG are connected by a unique path. The simulation data sets were generated following the set-ups of [13]:

- Generate the expression level of center gene 1, $X_1 \sim N(0, 1)$.
- Assume node s and each of its downstream genes follow a bivariate normal distribution with means 0 and unit variances with correlation 0.7. Thus, the expression level of each downstream gene is distributed as $N(0.7X_s, 0.51)$.
- Generate outcome Y from a logistic regression model: $\text{Logit}\,(Pr(Y = 1|X)) = X^T\beta + \beta_0$, $\beta_0 = 2$, where X is a vector of the expression levels of all the genes, and β is the corresponding coefficient vector.

We considered three sets of informative genes. The effect of each informative gene on the outcome was equal to that of its upstream node divided by the square-root of the upstream node's degree. All the other genes were noninformative, which had no effect on the outcome. Three sets of true coefficients $\beta = (\beta_1, \beta_2, \ldots, \beta_{26})$ were specified in three scenarios:

1. PT setting: one of the tree branches of the hierarchical tree or DAG (gene 1, 2, 5, 6, 14, 15, and 16) was informative.
   $\beta = (5, \beta_1/\sqrt{3}, 0, 0, \beta_2/\sqrt{3}, \beta_2/\sqrt{3}, \underbrace{0, \ldots, 0}_{7}, \beta_5/\sqrt{2},$
   $\beta_6/\sqrt{3}, \beta_6/\sqrt{3}, \underbrace{0, \ldots, 0}_{10})$.
2. PW setting: pathway $\{1, 3, 7, 17\}$ was informative.
   $\beta = (5, 0, \beta_1/\sqrt{3}, 0, 0, 0, \beta_3/\sqrt{4}, \underbrace{0, \ldots, 0}_{9}, \beta_7/\sqrt{2},$
   $\underbrace{0, \ldots, 0}_{9})$.
3. PW setting: pathway $\{1, 2, 5, 14\}$ was informative.
   $\beta = (5, \beta_1/\sqrt{3}, 0, 0, \beta_2/\sqrt{3}, \underbrace{0, \ldots, 0}_{8}, \beta_5/\sqrt{2}, \underbrace{0, \ldots, 0}_{12})$.

In each scenario, we simulated 50, 50 and 10,000 observations for each training, tuning and test dataset. For each tuning parameter value, we obtained a classifier from the training data, applied it to the tuning data, and identified $\hat{\lambda}$ that yielded the minimal misclassification error over the

Table 1. Simulation results (averaged over 100 runs) for a simple network with $p = 26$ genes; 7, 4, and 4 informative genes were in the three scenarios respectively

Scenario	Method	Test Error (SE)	# False Negative (SE)	Model Size (SE)
1	STD	0.129 (0.003)	0.00 (0.00)	26.00 (0.00)
	L1	0.122 (0.003)	2.81 (0.14)	7.19 (0.36)
	NG ($w = 1$)	0.145 (0.004)	0.26 (0.05)	14.73 (0.47)
	NG ($w = \sqrt{d}$)	0.123 (0.004)	0.10 (0.04)	15.32 (0.50)
	NG ($w = d$)	0.108 (0.003)	0.12 (0.05)	14.47 (0.56)
	PW ($w = 1$)	0.152 (0.003)	0.84 (0.07)	15.57 (0.65)
	PW ($w = \sqrt{d}$)	0.136 (0.003)	0.93 (0.08)	16.90 (0.61)
	PW ($w = d$)	0.126 (0.003)	1.60 (0.13)	14.44 (0.61)
	PT ($w = 1$)	0.107 (0.003)	1.94 (0.14)	9.42 (0.53)
	PT ($w = \sqrt{d}$)	0.107 (0.004)	2.51 (0.13)	8.65 (0.53)
	PT ($w = d$)	0.110 (0.004)	3.08 (0.13)	7.39 (0.44)
2	STD	0.147 (0.003)	0.00 (0.00)	26.00 (0.00)
	L1	0.118 (0.004)	0.99 (0.07)	6.56 (0.33)
	NG ($w = 1$)	0.162 (0.004)	0.16 (0.05)	17.83 (0.52)
	NG ($w = \sqrt{d}$)	0.138 (0.003)	0.01 (0.01)	17.33 (0.52)
	NG ($w = d$)	0.125 (0.003)	0.04 (0.02)	16.09 (0.57)
	PW ($w = 1$)	0.174 (0.003)	0.32 (0.05)	18.93 (0.48)
	PW ($w = \sqrt{d}$)	0.163 (0.003)	0.42 (0.06)	16.31 (0.48)
	PW ($w = d$)	0.144 (0.003)	0.43 (0.06)	14.94 (0.47)
	PT ($w = 1$)	0.111 (0.003)	0.72 (0.08)	7.74 (0.49)
	PT ($w = \sqrt{d}$)	0.109 (0.003)	1.05 (0.08)	6.64 (0.52)
	PT ($w = d$)	0.113 (0.003)	1.32 (0.07)	6.07 (0.40)
3	STD	0.143 (0.002)	0.00 (0.00)	26.00 (0.00)
	L1	0.120 (0.003)	0.96 (0.07)	6.67 (0.34)
	NG ($w = 1$)	0.149 (0.003)	0.09 (0.04)	16.04 (0.53)
	NG ($w = \sqrt{d}$)	0.127 (0.003)	0.02 (0.01)	15.50 (0.51)
	NG ($w = d$)	0.117 (0.003)	0.06 (0.03)	13.25 (0.52)
	PW ($w = 1$)	0.165 (0.002)	0.33 (0.05)	17.98 (0.55)
	PW ($w = \sqrt{d}$)	0.147 (0.003)	0.34 (0.05)	16.96 (0.49)
	PW ($w = d$)	0.141 (0.003)	0.41 (0.06)	15.39 (0.51)
	PT ($w = 1$)	0.106 (0.003)	0.59 (0.07)	8.07 (0.53)
	PT ($w = \sqrt{d}$)	0.110 (0.003)	1.10 (0.08)	5.84 (0.34)
	PT ($w = d$)	0.113 (0.003)	1.28 (0.07)	5.71 (0.27)

tuning set. Then we used the classifier corresponding to $\hat{\lambda}$ to compute the misclassification error on the test data. The entire process was repeated 100 times (i.e., 100 independent runs). The means of the test misclassification errors, false negatives (the number of informative genes whose coefficients were estimated to be zero), model sizes (the number of genes whose coefficients were estimated to be nonzero), and their corresponding standard errors ($sd/\sqrt{run}$) are reported in Table 1.

Evidently, DGC-SVM-PT generated models as sparse as that obtained from L1-SVM, and gave the most accurate prediction among all the other methods. In addition, the center gene, gene 1, was detected in each run by DGC-SVM-PT. NG-SVM and DGC-SVM-PW yielded fewer false negatives due to the larger models produced by each method. The weight function $w = d$ improved the classification accuracy, slightly shrank the model size, and kept almost the same false negatives for NG-SVM and DGC-SVM-PW compared with the other two weight functions. In contrast, $w = 1$ worked better for DGC-SVM-PT. It reduced the false negatives while produced models of comparable predictive performance to that with $w = \sqrt{d}$ or $w = 1$. Therefore, DGC-SVM-PT with $w = 1$ was the winner. In addition, it also improved reproducibility. The most frequently-recovered pathways from each method (L1-SVM, NG-SVM $w = \sqrt{d}$, DGC-SVM-PW $w = \sqrt{d}$, and DGC-SVM-PT $w = \sqrt{d}$) are displayed in Fig. 2. Both DGC-SVM-PT and L1-SVM missed the leaves under each scenario. However, both identified the majority parts of the true pathways. Compared with all the other methods, DGC-SVM-PT detected the same pathway with a much higher frequency. Therefore, the identified pathways by this method were more reproducible.

3.2 A complicated network

Next, we explored the complicated network originating from gene 1 as displayed in Fig. 3. For this network, there

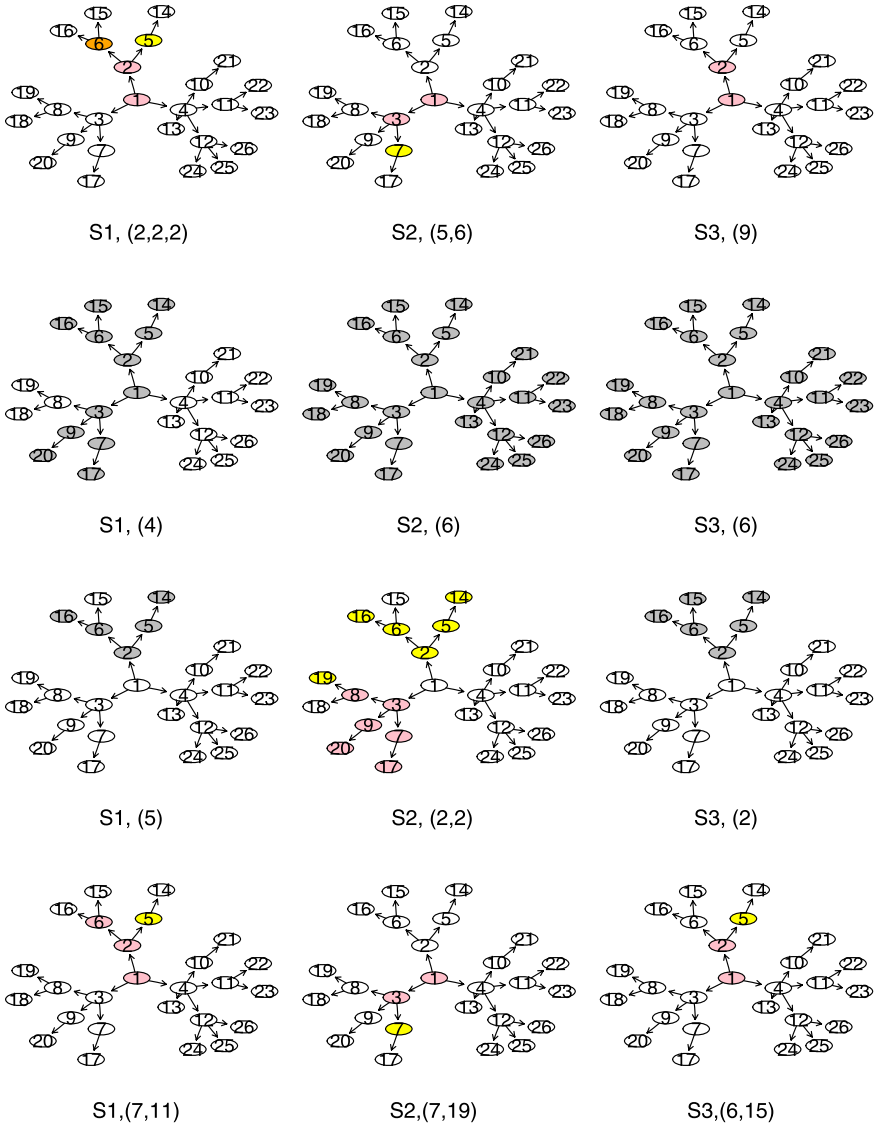

Figure 2. Most frequently recovered pathways. Rows from top to bottom: L1-SVM, NG-SVM, DGC-SVM-PW, and DGC-SVM-PT. Columns from left to right: scenario 1 (S1) $\{1, 2, 5, 6, 14, 15, 16\}$*, scenario 2 (S2)* $\{1, 3, 7, 17\}$*, and scenario 3 (S3)* $\{1, 2, 5, 14\}$*. Frequencies of the recovered pathways are included in parentheses.*

exists one pair of genes connected by more than one path. Therefore, the DAG derived from the complicated network does not form a tree. For example, gene 32 has both gene 23 and gene 3 at its upstream. In addition, genes at the same level are connected, such as genes 22 and 23, and genes 33 and 34. By definition, genes with no downstream genes are considered as leaves. Even though gene 22 is connected with gene 23, gene 22 is considered as a leaf because gene 23 is at the same level as gene 22. Likewise, genes 33 and 34 are both treated as leaves. Therefore, unlike the simple network, the DAG defined by the complicated network contains directed edges characterizing upper-lower relationships but ignores undirected edges describing lateral connections. Genes 1, 2, and 3 are assumed to be disease genes that affect the outcome weakly but importantly. Our goal is to identify disease genes, including those that play a critical role in mediating other genes in multiple biological processes even if their direct effects on the outcome are weak.

The simulated data were generated similarly as for the simple network. Here we considered two scenarios: (1) genes 1, 2, and 3 had weak effects ($\beta_1 = \beta_2 = \beta_3 = 0.1$), and leaf gene 32 had a strong effect ($\beta_{32} = 10$); (2) the three disease genes had the same effect as in scenario (1) whereas leaf gene 34 had a strong effect ($\beta_{34} = 10$).

Table 2 suggests that L1-SVM generated sparse models yielding the most accurate prediction under both scenarios. However, on average, it missed about 2.5 out of 4 informative genes and identified only around 0.5 out of 3 disease

Table 2. Simulation results (averaged over 100 runs) for a complicated network with $p = 13$ genes. Four informative genes were in each scenario

Scenario	Method	Test Error (SE)	# False Negative (SE)	Model Size (SE)	# Disease Genes (SE)
1	STD	0.131 (0.002)	0.00 (0.00)	13.00 (0.00)	3.00 (0.00)
	L1	0.089 (0.003)	2.59 (0.07)	3.04 (0.25)	0.41 (0.07)
	NG ($w = 1$)	0.214 (0.005)	0.00 (0.00)	13.00 (0.00)	3.00 (0.00)
	NG ($w = \sqrt{d}$)	0.210 (0.004)	0.00 (0.00)	13.00 (0.00)	3.00 (0.00)
	NG ($w = d$)	0.216 (0.005)	0.00 (0.00)	13.00 (0.00)	3.00 (0.00)
	PW ($w = 1$)	0.109 (0.003)	1.10 (0.14)	9.58 (0.36)	1.90 (0.14)
	PW ($w = \sqrt{d}$)	0.115 (0.003)	0.86 (0.12)	9.70 (0.34)	2.14 (0.12)
	PW ($w = d$)	0.117 (0.003)	0.46 (0.09)	10.65 (0.25)	2.54 (0.09)
	PT ($w = 1$)	0.146 (0.004)	0.00 (0.00)	10.79 (0.20)	3.00 (0.00)
	PT ($w = \sqrt{d}$)	0.145 (0.004)	0.00 (0.00)	10.86 (0.18)	3.00 (0.00)
	PT ($w = d$)	0.145 (0.004)	0.00 (0.00)	10.62 (0.18)	3.00 (0.00)
2	STD	0.137 (0.003)	0.00 (0.00)	13.00 (0.00)	3.00 (0.00)
	L1	0.095 (0.003)	2.46 (0.08)	3.75 (0.29)	0.54 (0.08)
	NG ($w = 1$)	0.217 (0.005)	0.00 (0.00)	13.00 (0.00)	3.00 (0.00)
	NG ($w = \sqrt{d}$)	0.213 (0.004)	0.00 (0.00)	13.00 (0.00)	3.00 (0.00)
	NG ($w = d$)	0.215 (0.004)	0.00 (0.00)	13.00 (0.00)	3.00 (0.00)
	PW ($w = 1$)	0.101 (0.004)	2.12 (0.13)	5.52 (0.44)	0.88 (0.13)
	PW ($w = \sqrt{d}$)	0.099 (0.003)	1.56 (0.14)	6.79 (0.44)	1.44 (0.14)
	PW ($w = d$)	0.107 (0.004)	1.16 (0.13)	8.03 (0.40)	1.84 (0.13)
	PT ($w = 1$)	0.151 (0.004)	0.00 (0.00)	9.62 (0.25)	3.00 (0.00)
	PT ($w = \sqrt{d}$)	0.150 (0.004)	0.00 (0.00)	9.97 (0.23)	3.00 (0.00)
	PT ($w = d$)	0.152 (0.005)	0.00 (0.00)	9.98 (0.22)	3.00 (0.00)

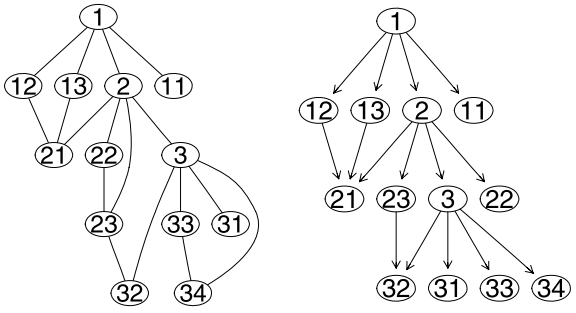

Figure 3. Left: a complicated network originating from gene 1. Right: DAG defined from the network.

genes. In contrast, NG-SVM yielded models that contained every gene. Even though DGC-SVMs led to larger test errors than L1-SVM, they detected most of the informative and disease genes. In particular, DGC-SVM-PT detected all the informative and disease genes, where the results are insensitive to the choice of weight. Figure 4 displays the pathways most frequently identified by each method ($w = \sqrt{d}$ when a weight was involved) except the standard SVM in each scenario. It is evident that L1-SVM captured the gene that exerted the largest influence on the response with the highest frequency (genes 32 and 34 in scenarios 1 and 2 respectively). NG-SVM involved all genes and hence that did not distinguish the informative from noise genes. After a close examination, we find that the DGC-SVM-PT selected all the informative genes as well as all the three disease genes in each run even though the three disease genes affected the response weakly. Although this method generated larger models with less accurate prediction compared with L1-SVM, it exactly achieved our goal of identifying all the disease genes along a pathway, including those with only weak effects.

4. APPLICATIONS TO MICROARRAY DATA

To evaluate their performance in the real world, we applied the proposed DGC-SVMs to two microarray data sets related to breast cancer metastasis [8, 20] and Parkinson's disease [16] (Gene Expression Omnibus: GSE6613; `http://www.lncbi.nlm.nih.gov/geo/`).

4.1 Breast cancer metastasis

The breast cancer metastasis (BC) data set contains expression levels of 8,141 genes from 286 patients, 106 of whom developed metastasis within a 5-year follow-up after surgery. The data set includes three tumor suppressor genes, *TP53*, *BRCA1*, and *BRCA2*, which are known for preventing uncontrolled cell proliferation, and for playing a critical role in repairing the chromosomal damage. The malfunction of these genes leads to an increased risk of breast cancer. Together with the expression data set, the protein-protein interaction (PPI) network previously used by [8] was adopted as our prior biological information, which was obtained by assembling a pooled data set comprising 57,235 interactions among 11,203 proteins and curation of the literature. We

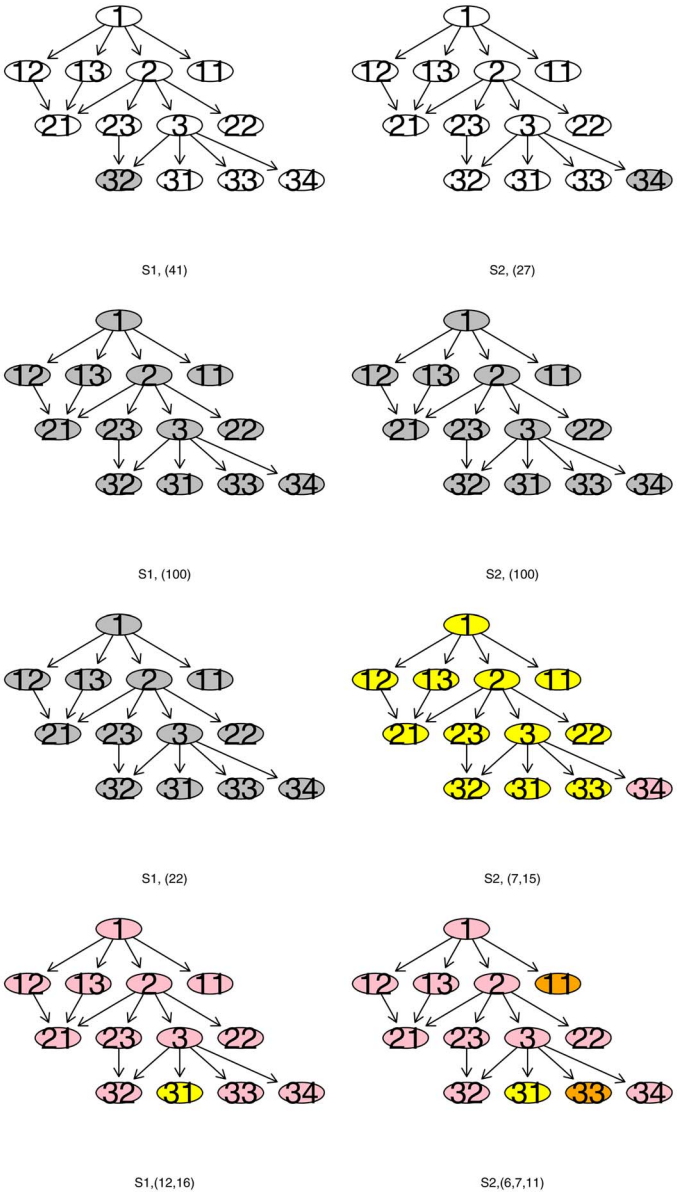

Figure 4. Most frequently recovered pathways. Rows from top to bottom: L1-SVM, NG-SVM, DGC-SVM-PW, and DGC-SVM-PT. Columns from left to right: scenario 1 (S1) $\{1, 2, 3, 32\}$*, and scenario 2 (S2)* $\{1, 2, 3, 34\}$*. Frequencies of the recovered pathways are included in parentheses.*

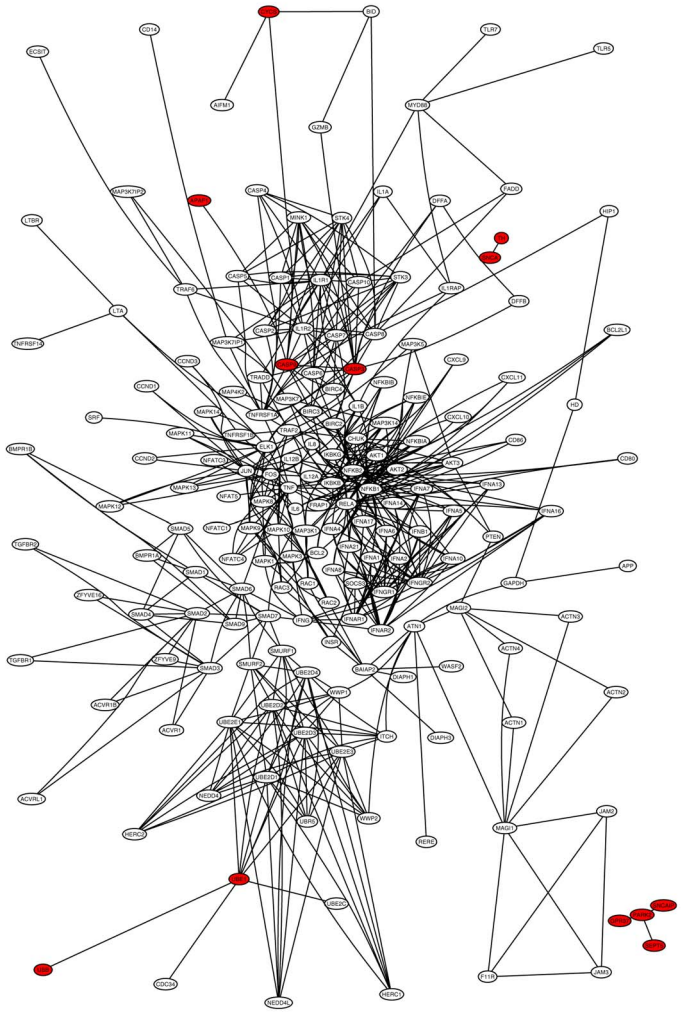

*Figure 5. PD-net: subnetworks grown from the 12 PD disease genes (*UBB, UBE1, CASP9, CASP3, APAF1, CYCS, PARK2, GPR37, SEPT5, SNCAIP, SNCA, *and* TH*); 181 genes in total. PD genes are in red.*

considered a subnetwork consisting of the direct neighbors of the three tumor suppressor genes, denoted as BC-1nb-net, where expression levels of a total of 294 genes that belong to BC-1nb-net were available. In our analysis, due to their prominent roles, *TP53*, *BRCA1*, and *BRCA2* were considered as center genes of BC-1nb-net.

For evaluation, we randomly split the data into training, tuning, and testing set with 95, 95 and 96 observations respectively. The expression level of each gene was normalized to have mean 0 and standard deviation 1 across samples. Given any value from a prespecified set of wide-ranging values for the tuning parameter λ, we computed the classifier $\hat{f}_\lambda$ on the training set and applied $\hat{f}_\lambda$ to tuning set. The value of λ yielding the minimal misclassification error on the tuning data was chosen as $\hat{\lambda}$. Then we applied the classifier $\hat{f}_{\hat{\lambda}}$ on the test set for evaluation. We compared the performance of DGC-SVM with that of STD-SVM, L1-SVM, and NG-SVM in terms of misclassification error, selection of mutant genes (genes with available mutation frequencies), and model sparsity, averaged over 50 runs. Since genes with large mutation frequencies are more likely to malfunction, disturbing the aggregate activity of the network, a method that can detect more such mutant genes may be considered to perform better. The mutation frequencies of 227 genes are available [8]. Among the 294 genes in BC-1nb-net, 40

Table 3. BC-1nb-net: 294 genes including 40 cancer genes (CA) and 7 cancer genes with mutation frequencies larger than 0.10 (CA-LMF); misclassification error, number of selected CA; number of selected CA-LMF, number of selected genes, and their standard errors (SE in parentheses) obtained by averaging over 50 runs

Method	Error (SE)	# CA-LMF	# CA	# Genes
STD	0.396 (0.006)	7.00 (0.00)	40.00 (0.00)	294.00 (0.00)
L1	0.384 (0.007)	0.46 (0.10)	3.92 (0.46)	26.48 (3.01)
NG ($w = 1$)	0.385 (0.008)	0.58 (0.10)	3.82 (0.45)	29.38 (2.91)
NG ($w = \sqrt{d}$)	0.388 (0.006)	0.64 (0.11)	4.90 (0.60)	31.82 (2.95)
NG ($w = d$)	0.404 (0.006)	0.86 (0.11)	6.42 (0.58)	36.88 (2.45)
PW ($w = 1$)	0.380 (0.007)	0.78 (0.15)	4.86 (0.55)	42.32 (4.27)
PW ($w = \sqrt{d}$)	0.382 (0.006)	0.62 (0.12)	4.48 (0.62)	30.50 (3.64)
PW ($w = d$)	0.396 (0.006)	0.86 (0.12)	3.96 (0.49)	23.80 (2.89)
PT ($w = 1$)	0.373 (0.006)	1.76 (0.16)	14.90 (1.09)	88.18 (6.31)
PT ($w = \sqrt{d}$)	0.395 (0.006)	1.84 (0.18)	12.52 (0.61)	63.88 (2.12)
PT ($w = d$)	0.396 (0.005)	1.58 (0.16)	7.20 (0.60)	31.76 (2.64)

genes had mutation frequencies (denoted as cancer genes), 7 of which (*ABL1*, *JAK2*, *p53*, *PTEN*, *p14ARF*, *PTCH*, and *RB*) had a mutation frequency larger than 0.10 (denoted as cancer genes with large mutation frequency).

As indicated in Table 3, all the methods had similar predictive accuracies even though DGC-SVM-PT with $w = 1$ performed slightly better. However, an improvement in detecting clinically relevant genes was seen evidently by incorporating the prior network information. Compared with L1-SVM, NG-SVM with $w = d$ detected almost twice as many frequently mutant cancer genes and more than 1.5 times cancer genes with models less than 1.5 times larger. DGC-SVM-PW with $w = d$ generated a more sparse model with 3 genes less than L1-SVM while detected the same number of cancer genes and almost twice as many frequently mutant cancer genes as L1-SVM. The advantage of DGC-SVM-PT with $w = d$ in gene selection was evident. It captured almost 3.5 times as many frequently mutant cancer genes and almost twice cancer genes by a model only 1.2 times larger than L1-SVM. The proposed DGC-SVM prevailed the other methods in identifying clinically important genes.

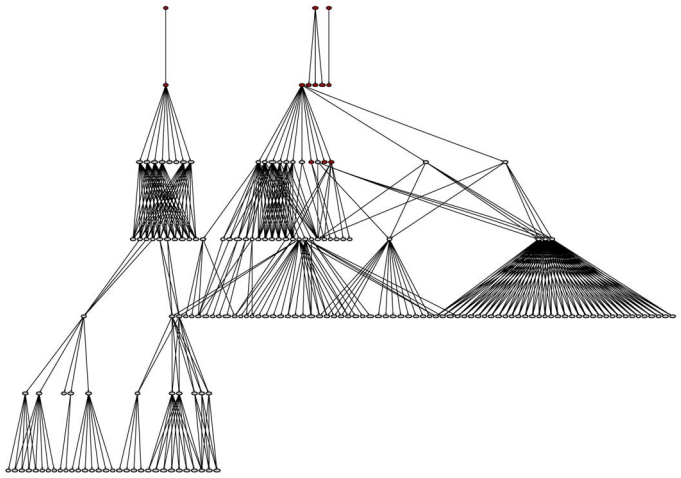

*Figure 6. DAG transformed from PD-net: 12 PD disease genes (*UBB, UBE1, CASP9, CASP3, APAF1, CYCS, PARK2, GPR37, SEPT5, SNCAIP, SNCA, *and* TH*); 181 genes in total. PD genes are in red.*

4.2 Parkinson's disease

The Parkinson's disease (PD) data set includes disease status and expression levels of 22,283 genes from 105 patients with 50 cases and 55 controls [16]. The gene network information was obtained from two sources: (1) the network in [13], which combines 33 KEGG regulatory pathways and contains 1,523 genes and 6,865 edges; (2) the Parkinson's disease KEGG pathway (PD-KEGG, `http://www.genome.ad.jp/kegg/pathway/hsa/hsa05020.html`), which uncovers the interactions of 27 PD disease genes as of November 2008. A total of 12 out of the 27 PD disease genes are contained in the network of [13]: *UBB*, *UBE1*, *CASP9*, *CASP3*, *APAF1*, *CYCS*, *PARK2*, *GPR37*, *SEPT5*, *SNCAIP*, *SNCA*, and *TH*. We focused our analysis on a subnetwork expanded from the 12 disease genes with the network of [13], denoted as PD-net (Fig. 5), which consists of four components: (1) the 6th-order-neighbor-subnetwork of *UBB* (A direct neighbor of *UBB* is defined as a 1st-order-neighbor; a direct neighbor of a 1st-order-neighbor of *UBB* as a 2nd-order-neighbor; and so on.); (2) the 3rd-order-neighbor-subnetwork of *CASP9*; (3) the isolated four-gene-subnetwork including *PARK2*, *GPR37*, *SEPT5*, and *SNCAIP*; and (4) the isolated two-gene-subnetwork including *SNCA* and *TH*. A total of 181 genes belong to PD-net. Note that *PARK2*/*GPR37*/*SEPT5*/*SNCAIP* and *SNCA*/*TH* form two islands respectively. The DAG of PD-net (Fig. 6) was obtained by merging the DAG of *UBB* and that of *CASP9* at the common node *SMAD7*. Note the two islands in Fig. 6.

The data set was randomly split into training, tuning, and test sets with 40, 20, and 45 observations respectively. The

Table 4. Parkinson's disease: misclassification error, number of selected disease genes, number of selected genes, and their standard errors (SE in parentheses) obtained by averaging over 50 runs

Method	Error (SE)	# Disease Genes	# Genes
STD	0.448 (0.011)	12.00 (0.00)	181.00 (0.00)
L1	0.490 (0.008)	1.50 (0.19)	16.86 (1.58)
Final	-	5	70
NG ($w = 1$)	0.500 (0.011)	3.32 (0.39)	15.58 (1.72)
Final	-	8	83
NG ($w = \sqrt{d}$)	0.501 (0.009)	0.66 (0.19)	12.00 (1.34)
Final	-	12	102
NG ($w = d$)	0.493 (0.008)	0.00 (0.00)	14.10 (1.44)
Final	-	5	133
PW ($w = 1$)	0.481 (0.010)	4.80 (0.43)	13.10 (1.52)
Final	-	9	97
PW ($w = \sqrt{d}$)	0.493 (0.009)	3.26 (0.41)	13.22 (1.70)
Final	-	9	103
PW ($w = d$)	0.493 (0.007)	1.52 (0.28)	14.70 (1.88)
Final	-	7	97
PT ($w = 1$)	0.407 (0.011)	4.56 (0.41)	42.34 (4.64)
Final	-	6	108
PT ($w = \sqrt{d}$)	0.445 (0.013)	2.40 (0.18)	45.04 (3.80)
Final	-	6	113
PT ($w = d$)	0.456 (0.011)	2.42 (0.15)	55.56 (4.08)
Final	-	5	117

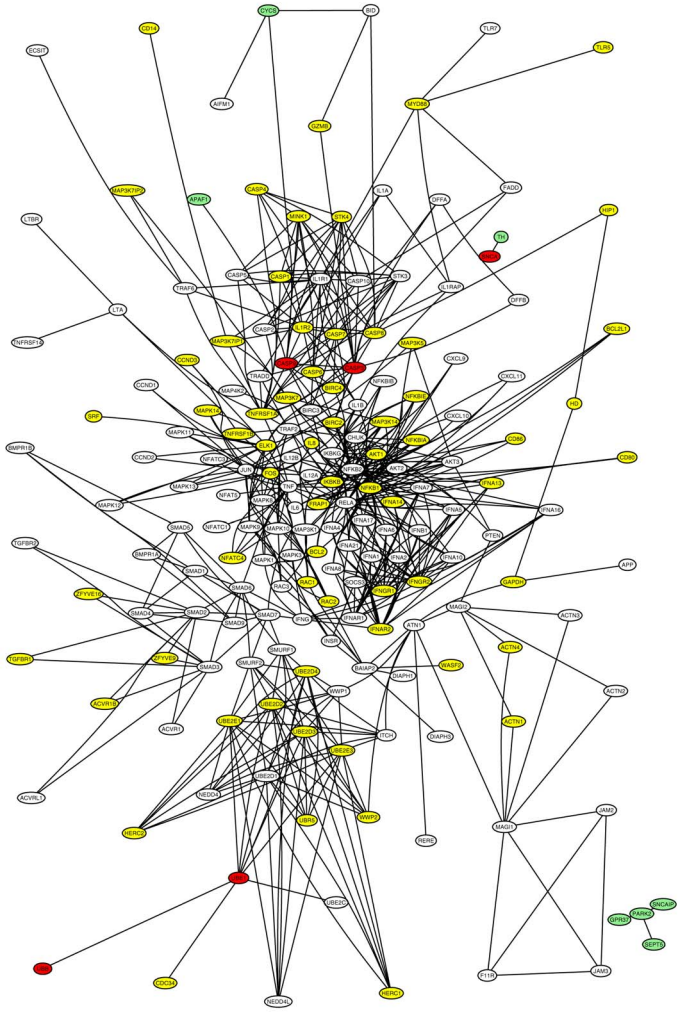

*Figure 7. L1-SVM final model: selected PD genes (*UBB, UBE1, CASP9, CASP3, *and* SNCA*) in red; missed PD genes (*APAF1, CYCS, PARK2, GPR37, SEPT5, SNCAIP, *and* TH*) in green; other selected genes in yellow; unselected genes in white; 181 genes in total.*

expression level of each gene was normalized to have mean 0 and standard deviation 1 across samples. The performance of each method was evaluated on the test set by the misclassification error, the selection of PD genes, and model sparsity averaged over 50 runs. Again five methods were compared: STD-SVM, L1-SVM, NG-SVM, DGC-SVM-PW ($w = 1$, $w = \sqrt{d}$, or $w = d$), and DGC-SVM-PT ($w = 1$, $w = \sqrt{d}$, or $w = d$). To obtain the final model for each method, we used a new tuning data set by combining, in each run, the previous tuning and test data, leading to a sample size as large as 65. Here $\hat{\lambda}$ was identified on the new tuning set from a wide range of prespecified values. Over 50 runs, the misclassification errors were averaged corresponding to each tuning parameter value. The value $\hat{\lambda}$ leading to the minimal average error was used to fit the final model to all the data (105 observations). Note that the misclassification error from the final model was likely to be biased due to reuse of the data for training/tuning and test; the purpose of fitting the final model was to yield a set of selected genes.

As indicated in Table 4, the methods that incorporate the prior gene network information improved gene selection while maintaining predictive accuracy comparable to that of the STD-SVM and L1-SVM. L1-SVM generated models with an average of 16.86 genes including 1.50 PD genes. NG-SVM with $w = 1$ detected more than twice as many PD genes with models having 1.28 less genes than L1-SVM. The improvement in gene selection of DGC-SVM-PW with $w = 1$ was more significant. It produced models with 3.76 less genes while detected more than three times as many PD genes as L1-SVM. Both the methods had a predictive accuracy at a comparable level to that of L1-SVM. DGC-SVM-PT with $w = 1$ improved both prediction accuracy and gene selection. It reduced the misclassification error by 17% and also selected three times as many PD genes with a model size not so much larger than that of L1-SVM. Overall, DGC-SVM-PT with $w = 1$ outperformed the other methods.

The selected genes of the final model obtained from each method (L1-SVM, NG-SVM with $w = 1$, DGC-SVM-PW with $w = 1$, and DGC-SVM-PT with $w = 1$) are displayed in the networks in Figs 7–10. L1-SVM generated the sparsest final model and missed the most PD genes. NG-SVM neglected around a half of the nodes and a half of the leaves.

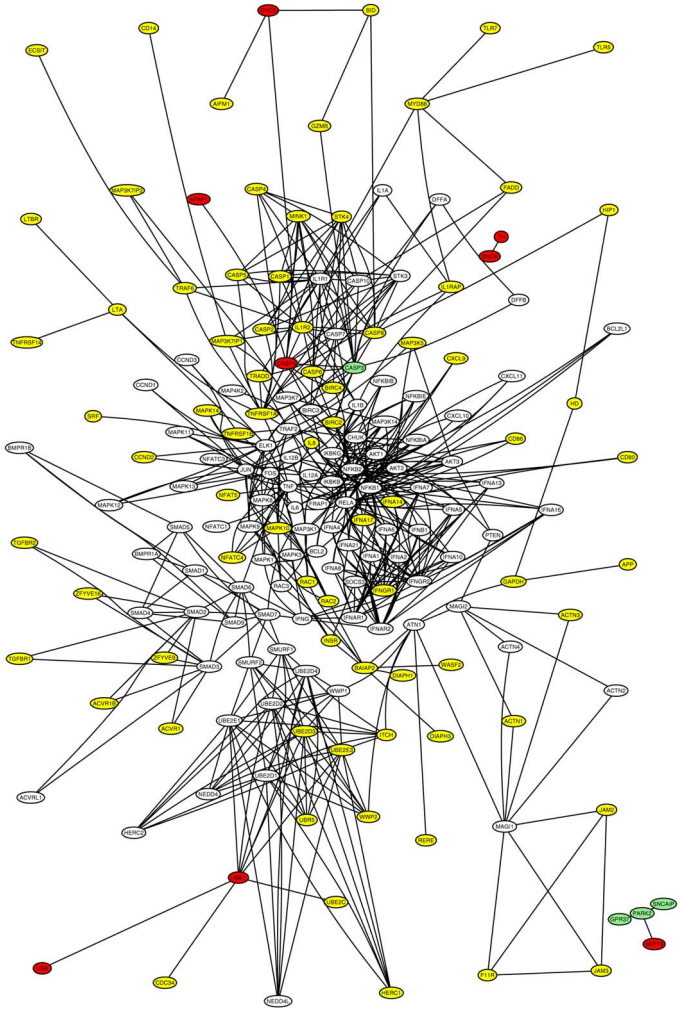

*Figure 8. NG-SVM with $w = 1$ final model: selected PD genes (*UBB, UBE1, CASP9, APAF1, CYCS, SEPT5, SNCA, *and* TH*) in red; missed PD genes (*CASP3, PARK2, GPR37, *and* SNCAIP*) in green; other selected genes in yellow; unselected genes in white; 181 genes in total.*

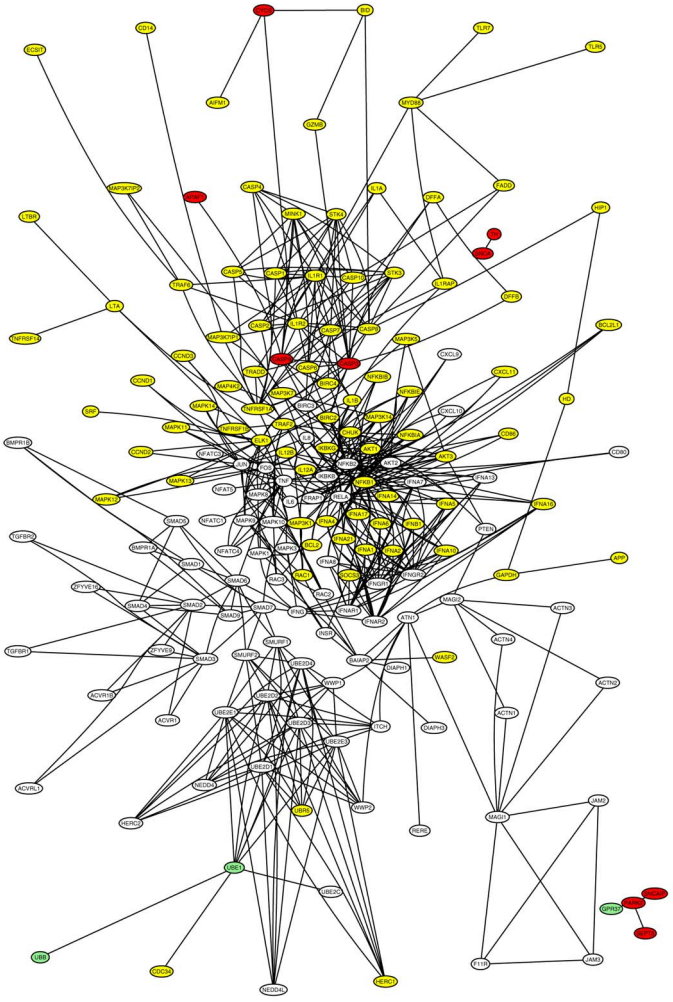

*Figure 9. DGC-SVM-PW with $w = 1$ final model: selected PD genes (*CASP9, CASP3, SNCA, CYCS, APAF1, TH, PARK2, SEPT5, *and* SNCAIP*) in red; missed PD genes (*UBB, UBE1, *and* GPR37,*) in green; other selected genes in yellow; unselected genes in white; 181 genes in total.*

According to DGC-SVM-PT, nodes located at higher level are more likely to have nonzero estimates. The majority of nodes were detected and also both center genes of the two islands were included by this method. DGC-SVM-PW identified the most PD genes among the four final models even though it neglected most part of the subnetwork derived from *UBB*.

5. DISCUSSION AND CONCLUSION

The availability of various repositories of gene networks and the accumulating knowledge on genes central to diseases make it possible to use these two sources of prior biological information to construct microarray-based classifiers. Employing such a network-based perspective not only sheds insight on deciphering the complexity within the network modules but also offers the possibility of detecting genes that play a critical role in mediating multiple biological processes but have weak direct effects on the outcome by themselves. Such genes are often ignored by expression analysis without the network information as pointed out by [8].

This paper proposes a penalty that encourages gene selection along pathways or within subnetworks. The identification of any gene leads to the search for disease genes along gene pathways or within subnetworks all the way toward genes central to the disease. The penalty enhances investigation of relationships within collectives of genes that contain both the selected genes and the central genes. By converting the undirected gene network into DAG, we imposed an upper-lower hierarchy on the gene network depending on

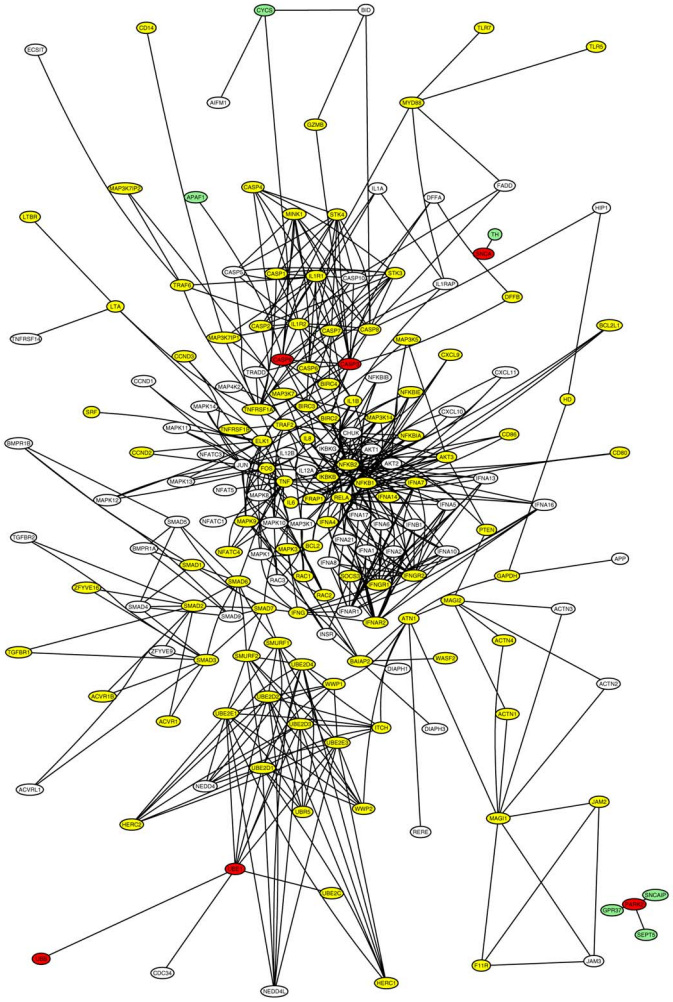

*Figure 10. DGC-SVM-PT with $w = 1$ final model: selected PD genes (*UBB, UBE1, PARK2, CASP3, SNCA, *and* CASP9*) in red; missed PD genes (*APAF1, CYCS, GPR37, SEPT5, SNCAIP, *and* TH*) in green; other selected genes in yellow; unselected genes in white; 181 genes in total.*

each gene's distance to the assumed network center, typically gene(s) central to the disease. The DAG facilitated the definition of gene groups according to one of two different ways of grouping, one based on grouping genes along linear pathways, and a second on grouping genes with all their downstream genes. The penalty term was constructed from the L_∞-norm being applied to each group. Combined with the hinge loss, we obtained our proposed method: DGC-SVM-PW and DGC-SVM-PT. The former attempts to realize selection along linear pathways in the network. The latter ensures the selection of an upper-level gene if any of its downstream genes is selected, thus realizing hierarchical selection. Due to this property, genes that have more downstream genes and are closer to the center are very likely to be detected. In addition, selection of any gene guarantees the inclusion of at least one central gene. According to the simulation studies, DGC-SVM detected more disease genes even if they potentially affected the outcome only weakly. Two real data applications showed that the new method prevailed STD-SVM and L1-SVM in capturing clinically relevant genes while making either more accurate or comparable prediction on the outcome. We conclude that DGC-SVM has the potential to be an effective classification tool for microarray data.

Even though its strength in improving gene selection has been established, DGC-SVM has some weaknesses. First, specification of the center of a network is somewhat arbitrary. As a matter of fact, different specifications result in different grouping structures and thus different penalties. Second, how to define DAG in presence of multiple center genes requires further investigation. We suggested an approach that is admittedly somewhat arbitrary. Third, DGC-SVM-PW is computationally intensive for a large network since a large number of groups are generated. Fourth, use of the weight function may improve gene selection at the expense of reduced predictive accuracy or vice versa; it is not guaranteed to better both at the same time. A further investigation is necessary to develop a simpler and more effective way to incorporate a prior gene network.

ACKNOWLEDGEMENTS

YZ and WP were partially supported by NIH grants HL65462 and GM081535; XS supported by NIH grant GM081535 and NSF grants IIS-0328802 and DMS-0604394. The Minnesota Supercomputing Institute (MSI) provided computing resources. We thank Dr Hongzhe Li and Dr Trey Ideker for providing the KEGG network and PPI network data respectively. We appreciate the helpful comments from the reviewers.

Received 18 February 2009

REFERENCES

[1] Alfarano, C., Andrade, C. E., Anthony, K., Bahroos, N., Bajec, M., Bantoft, K., Betel, D. et al. (2005). The Biomolecular Interaction Network Database and related tools 2005 update. *Nucleic Acids Res Database Issue* **33** D418–D424.

[2] Benson, M., Carlsson, L., Guillot, G., Jernas, M., Langston, M. A., Rudemo, M., and Andersson, B. (2006). A network-based analysis of allergen-challenged CD4+ T cells from patients with allergic rhinitis. *Genes and Immunity* **7** 514–521.

[3] Bondell, H. D. and Reich, B. J. (2008). Simultaneous regression shrinkage, variable selection, and supervised clustering of predictors with OSCAR. *Biometrics* **64** 115–123.

[4] Børresen-Dale, A. L. (2003). TP53 and breast cancer. *Human Mutation* **21** 292–300.

[5] Brown, M. P. S., Grundy, W. N., Lin, D., Cristianini, N., Sugnet, C. W., Furey, T. S., Ares, M., and Haussler, D. (2000). Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proceedings of the National Academy of Sciences of the United States of America* **97** 262–267.

[6] Calvano, S. E., Xiao, W. Z., Richards, D. R., Felciano, R. M., Baker, H. V., Cho, R. J., Chen, R. O., Brownstein, B. H., Cobb, J. P., Tschoeke, S. K., Miller-Graziano, C., Moldawer, L. L., Mindrinos, M. N., Davis, R. W., Tompkins, R. G., and Lowry, S. F. (2005). A network-based analysis of systematic inflammation in humans. *Nature* **437** 1032–1037.

[7] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning* **20** 273–297.

[8] Chuang, H. Y., Lee, E. J., Liu, Y. T., Lee, D. H., and Ideker, T. (2007). Network-based classification of breast cancer metastasis. *Molecular Systems Biology* **3** doi: 10.1038/msb4100180.

[9] Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association* **96** 1348–1360. MR1946581

[10] Furey, T., Cristianini, N., Duffy, N., Bednarski, D. W., Schummer, M., and Haussler, D. (2000). Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics* **16** 906–914.

[11] Kanehisa, M., Goto, S., Kawashima, S., Okuno, Y., and Hattori, M. (2004). The KEGG resource for deciphering the genome. *Nucleic Acids Res Database Issue* **32** D277–D280.

[12] Liu, M. W., Liberzon, A., Kong, S. W., Lai, W. R., Park, P. J., Kohane, I. S., and Kasif, S. (2007). Network-based analysis of affected biological processes in type 2 diabetes models. *PLoS Genetics* **3** 958–972.

[13] Li, C. and Li, H. (2008). Network-constrained regularization and variable selection for analysis of genomic data. *Bioinformatics* **24** 1175–1182.

[14] Pan, W., Xie, B., and Shen, X. (2009). Incorporating predictor network in penalized regression with application to microarray data. *Biometrics*, to appear. Available at `http://www.biostat.umn.edu./rrs.php` as Research Report 2009-001, Division of Biostatistics, University of Minnesota.

[15] Peri, S., Navarro, J. D., Kristiansen, T. Z., Amanchy, R., Surendranath, V., Muthusamy, B., Gandhi, T. K. et al. (2004). Human Protein Reference Database as a discovery resource for proteomics. *Nucleic Acids Res Database Issue* **32** D497–D501.

[16] Scherzer, C. R., Eklund, A. C., Morse, L. J., Liao, Z., Locascio, J. J., Fefer, D., Schwarzschild, M. A., Schlossmacher, M. G., Hauser, M. A., Vance, J. M., Sudarsky, L. R., Standaert, D. G., Growdon, J. H., Jensen, R. V., and Gullans, S. R. (2007). Molecular markers of early Parkinson's disease based on gene expression in blood. *Proceedings of the National Academy of Sciences of the United States of America* **104** 955–960.

[17] Soussi, T. and Béroud, C. (2003). Significance of TP53 mutations in human cancer: a critical analysis of mutations at CpG dinucleotides. *Human Mutation* **21** 192–200.

[18] Vapnik, V. (1995). *The Nature of Statistical Learning Theory*. Springer, New York. MR1367965

[19] Wang, L. and Shen, X. (2007). On L1-Norm multiclass support vector machines: Methodology and theory. *Journal of the American Statistical Association* **102** 583–594. MR2370855

[20] Wang, Y., Klijin, J. G., Zhang, Y., Sieuwerts, A. M., Look, M. P., Yang, F., Talantov, D., Timmermans, M., Meijer-van Gelder, M. E., Yu, J., Jatkoe, T., Berns, E. M., Atkins, D., and Foekens, J. A. (2005). Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer. *Lancet* **365** 671–679.

[21] Wu, S., Shen, X., and Geyer, C. (2008). Adaptive regularization through entire solution surface. *Biometrika*, to appear.

[22] Zhao, P., Rocha, G., and Yu, B. (2008). The composite absolute penalties family for grouped and hierarchical variable selection. *The Annals of Statistics*, to appear.

[23] Zhu, J., Rosset, S., Hastie, T., and Tibshirani, R. (2003). 1-norm support vector machines. *Neural Information Processing Systems*.

[24] Zhu, Y., Shen, X., and Pan, W. (2009). Network-based support vector machine for classification of microarray samples. *BMC Bioinformatics* **10** S21.

[25] Zou, H. and Yuan, M. (2008). The F_∞-norm Support Vector Machine. *Statistica Sinica* **18** 379–398. MR2416909

Yanni Zhu
Division of Biostatistics
School of Public Health
University of Minnesota

Wei Pan
Division of Biostatistics, MMC 303
School of Public Health
University of Minnesota
Minneapolis, Minnesota 55455–0392, U.S.A.
E-mail address: weip@biostat.umn.edu
url: www.biostat.umn.edu/rrs.php

Xiaotong Shen
School of Statistics
University of Minnesota