JOURNAL OF COMPUTATIONAL BIOLOGY
Volume 19, Number 6, 2012
© Mary Ann Liebert, Inc.
Pp. 694–709
DOI: 10.1089/cmb.2012.0065

Network-Induced Classification Kernels for Gene Expression Profile Analysis

OFER LAVI,[1,3] GIDEON DROR,[2] and RON SHAMIR[1]

ABSTRACT

Computational classification of gene expression profiles into distinct disease phenotypes has been highly successful to date. Still, robustness, accuracy, and biological interpretation of the results have been limited, and it was suggested that use of protein interaction information jointly with the expression profiles can improve the results. Here, we study three aspects of this problem. First, we show that interactions are indeed relevant by showing that co-expressed genes tend to be closer in the network of interactions. Second, we show that the improved performance of one extant method utilizing expression and interactions is not really due to the biological information in the network, while in another method this is not the case. Finally, we develop a new kernel method—called NICK—that integrates network and expression data for SVM classification, and demonstrate that overall it achieves better results than extant methods while running two orders of magnitude faster.

Key word: algorithms.

1. INTRODUCTION

In the past decade, gene expression profiles based on DNA microarrays have been widely used to detect disease biomarkers. These profiles, measuring thousands of gene expression levels simultaneously, served as the basis for feature selection and classification methods and have been shown to provide better prognosis than prior models (Paik et al., 2006). However, the biomarker sets created by such methods have several drawbacks: Analysis often results in hundreds of genes, biological interpretation of the selected genes is difficult, and the overlap between the sets of genes selected as features in similar studies is very poor (Ein-Dor et al., 2005). In addition, genes selected in one dataset often do not perform well on other datasets (Chuang et al., 2007). This lack of robustness of biomarker selection was decisively demonstrated by Ein-Dor et al. (2005). To overcome this problem, Ein-Dor et al. suggested enlarging the sample size, or dividing the sample in advance into known homogeneous subsets based on some prior knowledge, and analyzing each subset separately (Sørlie et al., 2003).

We would like then to develop methods for detecting sets of biomarkers that (1) are more meaningful biologically and (2) are more stable across different studies. Such sets would be more useful for downstream biological research. The two goals do not always go hand in hand; for example, Hwang et al. (2008)

[1]Blavatnik School of Computer Science, Tel Aviv University, Tel Aviv, Israel.
[2]Yahoo! Research, Haifa, Israel.
[3]IBM Haifa Research Lab, Haifa, Israel.

provide a list of four genes that are highly predictive for breast cancer prognosis and also biologically meaningful, but they were not differentially expressed in other breast cancer data sets.

One possible way to improve marker selection is by using additional biological knowledge in addition to the expression data. Several types of prior knowledge are available, including GO and KEGG gene annotations (Ashburner et al., 2000; Kanehisa and Goto, 2000), collections of small-scale regulatory pathways (Tian et al., 2005), and large-scale protein-protein interaction (PPI) and metabolic networks (Jensen et al., 2009; Snel et al., 2000; Rual et al., 2005; Aranda et al., 2009; Kerrien et al., 2007).

Several studies integrate network knowledge into gene expression analysis: A spectral approach is taken by Rapaport et al. (2007) for the purpose of noise reduction based on network topology. Ideker and colleagues (Chuang et al., 2007; Lee et al., 2008b) substitute the use of expression levels of individual genes with an aggregate of the expression levels of a set of genes within a subnetwork, greedily searching for such subnetwork markers within the network. Kuang and colleagues (Hwang et al., 2008; Tian et al., 2009) add network data into a loss function using an optimization framework approach for gene expression profile classification, and Zhu et al. (2009) add it by introducing an alternative regularization term to an SVM classifier objective function.

In order to find candidate genes that may serve as strong leads for downstream research, Nitsch et al. (2009) offer to score a gene using its neighbors' scores. The authors aim at finding what they call *disease-causing genes* and do not look at subsequent learning tasks such as classification or clustering. Last, Wei and colleagues (Wei and Pan, 2008; Wei and Li, 2007) take a statistical approach built on a *mixture model*, assuming two populations of genes—differentially expressed (DE) and equally expressed (EE), integrating the network by assuming that genes that are neighbors in some pathway are more likely to belong to the same population.

In this work we introduce a novel kernel we call NICK—a Network-Induced Classification Kernel for SVM—encapsulating the protein network topology and the relations between the different features. NICK is derived analytically by integrating a co-expression assumption into the SVM framework and can be used within any kernel method or as a plain linear transformation of the data that integrates network information into the data. We compared the performance of NICK within SVM classification to that of linear kernel SVM, and to two additional existing methods on data from a number of gene expression case-control studies. NICK outperforms a linear kernel in most settings and is found to be up to 250 times faster than the best extent method, achieving better or similar classification performance.

2. RESULTS AND DISCUSSION

First, we test and validate the assumption that genes that are close on the network are likely to have similar expression. Second, to assess if network information is truly helpful, we test two current methods that combine expression and network data. Surprisingly, we show that the network is not really helpful in one of them. Lastly, we introduce NICK and compare its performance to other methods.

2.1. Large scale networks are informative to gene expression analysis

The basis of using biological networks to enhance biomarker selection is the assumption that genes that are closer on the network are likely to have more similar expression. This co-expression assumption is made, for example, by Rapaport et al. (2007) and was validated to some extent, for example by Jansen et al. (2002). We first sought to systematically test this assumption using the STRING network (Snel et al., 2000; Jensen et al., 2009). To this end, we partitioned the gene pairs into several distinct populations according to their distance in the network and compared the distribution of absolute Pearson correlations of expression among the populations. The Pearson correlation was calculated using the expression data of Wang et al. (2005), containing 286 expression profiles of 22,000 RNA transcripts each.

Overall, the mean correlation of adjacent genes ($r = 0.123$) is only slightly higher than that of distant (non-adjacent) genes ($r = 0.111$), but this difference is highly significant (p-value $< 7.24 \times 10^{-31}$, one-tail t-test). Moreover, by partitioning the pairs according to their distance, we found that the larger the distance between two nodes, the lower the correlation between their expression profiles (Table 1). On the other hand, adjacent pairs that are connected by multiple two edge-paths obtained higher mean correlation than other adjacent pairs.

Table 1. Mean Correlation in Expression Among Different Gene Pair Populations

Pair population (sample size)	*Mean correlation*	*Significance*
Adjacent-baseline (15340)	0.12302	N/A
Distant (19978)	0.11078	7.24×10^{-31}
2-nodes away (654)	0.11746	6.99×10^{-2}
3-nodes away (3171)	0.11527	1.33×10^{-5}
4-nodes away (7733)	0.11167	7.24×10^{-18}
5-nodes away (4453)	0.11115	1.68×10^{-14}
6-nodes away (1207)	0.11263	9.56×10^{-5}
7+ nodes away (2755)	0.10857	9.65×10^{-15}
Adjacent members in 1 3-clique (5625)	0.1303	1.53×10^{-5}
Adjacent members in 2 3-cliques (2683)	0.13555	1.84×10^{-7}
Adjacent members in 3 3-cliques (1485)	0.14518	3.13×10^{-11}
Adjacent members in 5 3-cliques (622)	0.15377	8.83×10^{-9}

The right column measures the probability that the samples of the population and of the baseline came from the same distribution (*t*-test).

We also compared the correlation distribution of each gene pair subpopulation to that of the adjacent pairs population (Fig. 1). The percentage of adjacent genes that exhibit high correlation values is higher than that of distant genes, and this percentage decreases with gene distance. The opposite is true for the low correlation range. Adjacent genes that are also highly connected, as measured by their membership in multiple 3-cliques, show even higher percentage in the high correlation range.

A second co-expression assumption is used in the literature in the case of labeled data: a gene is termed *differentially expressed* if its expression level varies markedly between two labeled sets of samples (e.g., cases and controls). The assumption states (Wei and Li, 2007) that genes that are closer in the network will tend to have more similar differential expression pattern: they tend to change or not to change together

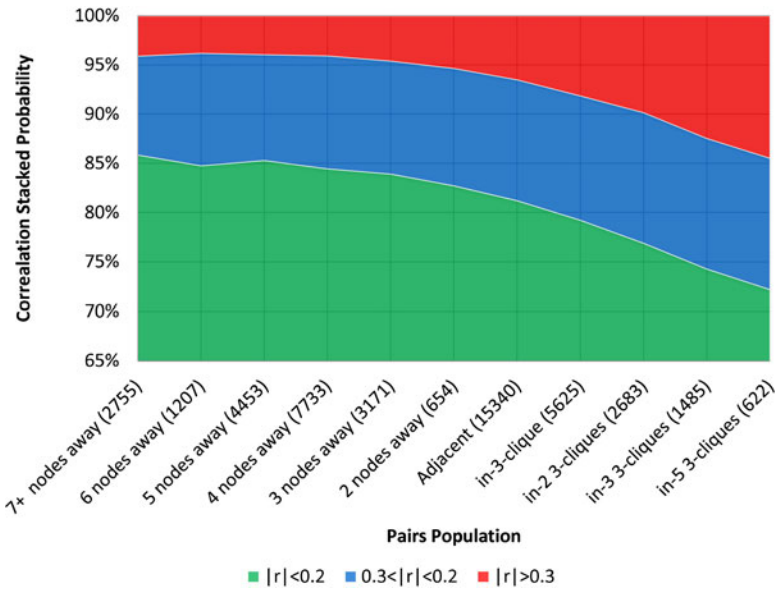

FIG. 1. Relation of gene pair expression correlation to the pair's physical closeness. The graph shows the distribution of correlation levels (in their absolute values) of gene pairs as a function of the pair population they belong to. The color indicates different levels of correlations. For each level and population, the Correlation Stacked Probability on the *y*-axis is the (stacked) probability that a pair exhibits the correlation level given its population. The probability for low correlation ($|r| < 0.2$, colored green) is higher in distant genes than in adjacent genes. The probability for high correlation ($|r| > 0.3$, colored red) is higher for adjacent genes than for distant genes. In parentheses, the number of pairs sampled in each population. For populations of gene-pairs that are also members of k 3-cliques, the greater the number of 3-cliques the pair shares, the higher the percentage of highly correlated pairs.

among the two sets of samples. This assumption follows the co-expression assumption: if close genes tend to co-express then if one gene is differentially expressed, then its neighbors will tend to be differentially expressed as well.

Often, labeled datasets are used to build a model which can later be used to classify expression profiles of unknown class. In such a model, an additional assumption (Hwang et al., 2008) is that genes that are close in the network will tend to have similar contribution to the classification model. Again, this assumption follows the co-expression assumption: if close genes tend to co-express they are also likely to have similar contribution to the classification model.

2.2. *Does the network make a difference?*

Chuang et al. (2007) reported on classification using subnetworks as features, combining expression and protein interaction data. The developed algorithm, PinnacleZ, showed improvement in comparison to selecting genes independently using *t*-test. We wanted to test whether this improvement was due to the added biological information in the protein network. For this test, we randomly permuted gene names in the network, and used the expression data together with the permuted network for feature selection and classification. The randomized networks preserve the topology of the original network, but dissociate any correlation they may have with the expression profiles. The feature selection and classification process was repeated 50 times for the true and randomly permuted networks, and results were quantified using AUC score.

The test was conducted on two breast cancer datasets (Wang et al., 2005; van de Vijver et al., 2002) using two classification algorithms. As seen in Figure 2, using the real network does not give results that are better on average than using a permuted one.

We also conducted a single run comparison on eight more datasets, using two different PPI networks—STRING (Jensen et al. [2009], April 2008, containing 6243 genes and 19102 edges) and IntAct (Kerrien et al. [2007], Aranda et al. [2009], June 2008, containing 9178 genes and 17609 edges), and Naive Bayes as classifier. The results (Fig. 3) show that both the true and permuted networks improved over the *t*-test classification, but the true network is not better than the permuted ones.

PinnacleZ starts from each gene as seed and uses the network neighbors (up to distance *d*) to greedily improve the subnetwork's predictive power. In view of the results above, the improvement in using the network over *t*-test, does not seem to be due to the biological content of the network. We believe that the improvement is due to the greedy search the algorithm performs and not due to the true network topology. The network topology merely limits the subset of genes that are reachable from every node in the greedy improvement step. If this subset is large enough, the greedy algorithm will find a combination of genes within this subset that will improve the classification results, regardless of the validity of the biological

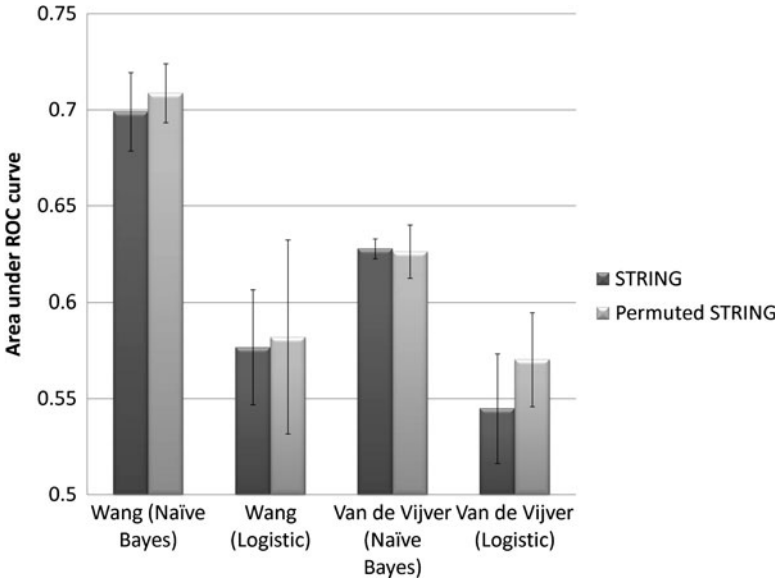

FIG. 2. PinnacleZ performance is indifferent to the underlying network. The figure presents the classification performance based on features selected by the PinnacleZ algorithm—AUC average and standard deviation of 50 runs of PinnacleZ using the STRING network and of 50 different permutations of the network. Results are shown for two different classification algorithms and two different datasets.

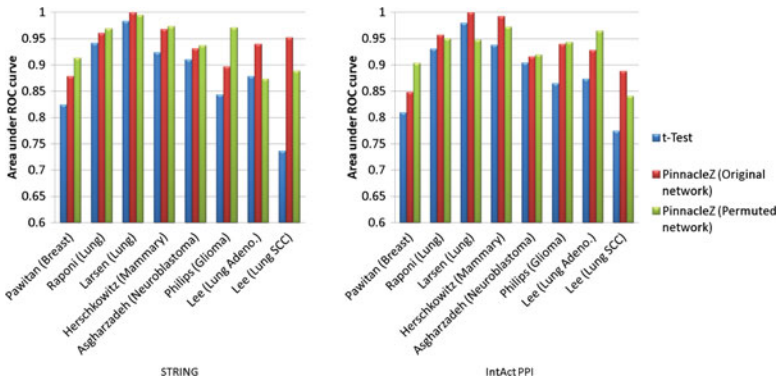

FIG. 3. Performance of PinnacleZ algorithm on the original and permuted networks. The figures present the classification performance of a Naive Bayes classifier, based on top 200 features selected by *t*-test and the PinnacleZ algorithm with the original network and with a randomized network. The test was repeated using two different networks: STRING (Jensen et al. 2009) and IntAct (Kerrien et al. 2007; Aranda et al., 2009) on eight different datasets (Pawitan et al., 2005; Raponi et al., 2006; Larsen et al., 2007; Herschkowitz et al., 2007; Asgharzadeh et al., 2006; Phillips et al., 2006; Lee et al., 2008a).

interactions in this subset. Hence, permuted networks, which allow a search space of roughly the same size but are not true biological interactions, perform equally well.

In a similar test on the HyperGene algorithm (Hwang et al., 2008), none of the randomized networks outperformed the real network, resulting a significant performance decrease (p-value $< 10^{-13}$, t-test) when substituting the real network with a permuted one. On the van't Veer dataset (van 't Veer et al., 2002), average AUC scores for HyperGene with 50 randomized networks ranged between 0.7024 and 0.8095, while 5-fold CV average score using the real network used by Hwang et al. (2008) was 0.845 for SVM and 0.893 for the HyperGene algorithm. In this case, it seems that the topology of the network does play a role in the improvement achieved.

2.3. NICK

We developed a novel method for integrating network information into the classification process. Our method, called NICK, builds a kernel that is based on the whole network, taking into account both distance and connectivity level between every two nodes. We summarize the method here briefly.

We modified the original SVM (Vapnik, 1999) objective function to reflect the assumption that close genes in the network should contribute similarly to the classification. We assume the network is a simple undirected graph $G = (V, E)$ with a set of nodes V and a set of edges E (each edge is represented by a pair of nodes (i, j) where $i, j \in V$). Our modified SVM problem is defined as:

$$\min_{w,\, w_0} \left\{ \frac{1}{2} ||\mathbf{w}||^2 + \frac{1}{2} \beta \sum_{(j,\, k) \in E} (w_j - w_k)^2 \right\}$$

subject to

$$(\mathbf{w}^T \mathbf{x}_i + w_0) \cdot y_i \geq 1 \qquad \text{for } i = 1, \ldots, n$$

where $\mathbf{x}_i$ is a vector of gene expression values representing the i'th sample and y_i is the i'th sample's label, $y_i \in \{-1, 1\}$, and each gene (feature) i corresponds to a node in the network. We seek a vector of weights $\mathbf{w}$, one weight per feature, that are regularized by the term $\frac{1}{2}\beta \sum_{(j,\, k) \in E} (w_j - w_k)^2$ so that the difference between weights of adjacent nodes will be minimized. This term is similar to the one used in Hwang et al. (2008) in a non-SVM formulation. $\beta \geq 0$ is a trade-off parameter where larger values of β give a stronger effect of the network on the model. The formulation with $\beta = 0$ is equivalent to the standard SVM.

This problem is a quadratic programming problem whose solution is equivalent to that of SVM with a new kernel. A derivation of a slightly more general problem is described in detail in Methods. The equivalence of our modified SVM problem to the standard SVM allows us to use any theory, algorithms and tools for solving the SVM problem in order to solve our problem as well.

TABLE 2. DATASETS USED IN THE EXPERIMENTAL RESULTS

Dataset	*Name*	*Reference*	*Cancer type*	n
GSE5123	Larsen	Larsen et al. (2007)	Lung	51
GSE4573	Raponi	Raponi et al. (2006)	Lung	130
van 't Veer	van 't Veer	van 't Veer et al. (2002)	Breast	117
E-TABM158	Chin	Chin et al. (2006)	Breast	118
GSE2034	Wang	Wang et al. (2005)	Breast	286
GSE3141	Nevins	Bild et al. (2006)	Lung	111
VanDeVijver	van de Vijver	van de Vijver et al. (2002)	Breast	295
GSE4922; GSE1456	Ivshina	Ivshina et al. (2006)	Breast	99
Pawitan	Pawitan	Pawitan et al. (2005)	Breast	159

n is the number of samples in the study.

The kernel matrix, denoted **Q**, can be expressed in terms of the *Laplacian matrix* **B** of the graph as $\mathbf{Q} = (\mathbf{I} + \beta\mathbf{B})^{-1}$, where **I** is the identity matrix. We show that the kernel can further decompose by means of *Cholesky decomposition* to a transformation matrix. Briefly, this transformation constructs a set of meta-features where each meta-feature is associated with a single feature (the *pivot*), and is a linear combination of other features within the pivot's connected component.

The kernel may be used in problems other than SVM that utilize kernel methods, and since it does not depend on the sample labels, it can be applied to unsupervised kernel methods. The transformation matrix can also be applied to other data analysis problems that do not rely on kernels.

Interestingly, the matrix **Q**, which we analytically derived from our regularized SVM formulation, was investigated for its algebraic properties and was applied in the fields of chemistry and electronic engineering (Golender et al., 1981; Merris, 1997, 1998; Chebotarev, 2008; Chebotarev and Shamis, 2006).

2.4. Improving classification performance using NICK

We tested the method on nine case-control gene expression datasets of breast and lung tumors. The datasets are listed in Table 2. All datasets relate to cancer prognosis, aiming at differentiating tumors of patients with good prognosis from those with poor prognosis, as reflected by survival time, or metastasis free period after the expression profile was taken. As a reference network, we used STRING (Jensen et al. [2009], April 2008), containing 6243 genes (nodes) and 19102 interactions (edges).

Results can be seen in Figure 4. For two datasets (Larsen et al., 2007; Chin et al., 2006), the baseline AUC was under 0.5, and thus they were excluded. For each dataset, we compared the CV AUC score of the

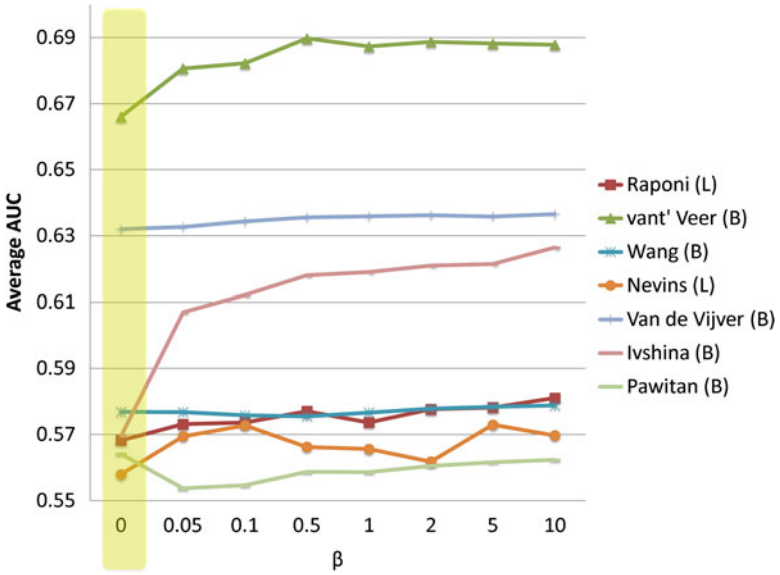

FIG. 4. Classification performance comparison. The figure displays average Area Under ROC curve measurements for SVM classification of the different datasets, using the NICK kernel with different values of β. The yellow shaded area where $\beta = 0$ serves as a baseline and is equivalent to standard SVM. B, breast cancer; L, lung cancer.

TABLE 3. COMPARISON OF CLASSIFICATION PERFORMANCE USING TRUE AND RANDOMIZED NETWORKS

Dataset	*Real network*	*Randomized networks*	*Real network rank*
van 't Veer	0.687	0.652 ± 0.0128	1
van de Vijver	0.636	0.618 ± 0.012	1
Ivshina	0.619	0.564 ± 0.012	1
Raponi	0.574	0.563 ± 0.027	11
Nevins	0.566	0.547 ± 0.025	5
Wang	0.576	0.58 ± 0.029	35

For each dataset, the table presents the AUC score achieved with the real network versus the average and range of AUC score achieved with 50 randomized networks. The last column shows the rank of the real network AUC score among the total 51 networks (50 randomized and 1 real).

baseline SVM ($\beta = 0$) with the AUC score with different values of β. Out of the seven datasets, five (Raponi, van t' Veer, Nevins, Van de Vijver, and Ivshina) showed an improvement with all values of β, one (Wang) showed a mixed result, and one (Pawitan) showed performance decrease for all values of β. In order to test for significance, we conducted a pairwise *t*-test for each dataset, keeping the same cross validation folds, comparing the AUC for differnet positive values β against the baseline SVM ($\beta = 0$). A total of 49 tests (7 datasets, 7 different values of β) were done. Thirty-eight tests showed improvement in the AUC score, 13 of them (in the datasets of Ivshina, Van 't veer and Nevins) found to be significant (FDR < 0.05). On the other hand, none of the 11 tests that showed performance decrease were statistically significant. One dataset showed significant improvement through all values of β (Ivshina). All significance tests were corrected for multiple testing, accounting for the multiple datasets, and multiple values of β. Figure 4 also shows that for most datasets showing improvement when using NICK, increasing β beyond 1 had minor effect. We thus used $\beta = 1$ as the default value in subsequent tests.

In order to test whether the improvement is indeed due to the network data, we further tested the NICK performance with randomized networks, as we did with HyperGene and PinnacleZ. We generated 50 different randomized networks and ran the NICK algorithm (with $\beta = 1$) using each randomized network on the five datasets the algorithm showed improvement on, and on the one that showed mixed results. For each dataset, we measured the average AUC score, comparing it to the average AUC result obtained with the original STRING network. Table 3 summarizes the results. Figure 5 shows the distribution of AUC scores for four of the datasets. On three datasets, the score with the real network ranked above all scores achieved with random networks, and for the remaining two it ranked 5 and 11. For comparison, on the Wang dataset, where NICK gave mixed results, the real network is ranked 35 among the total 51 networks (Fig. 5d).

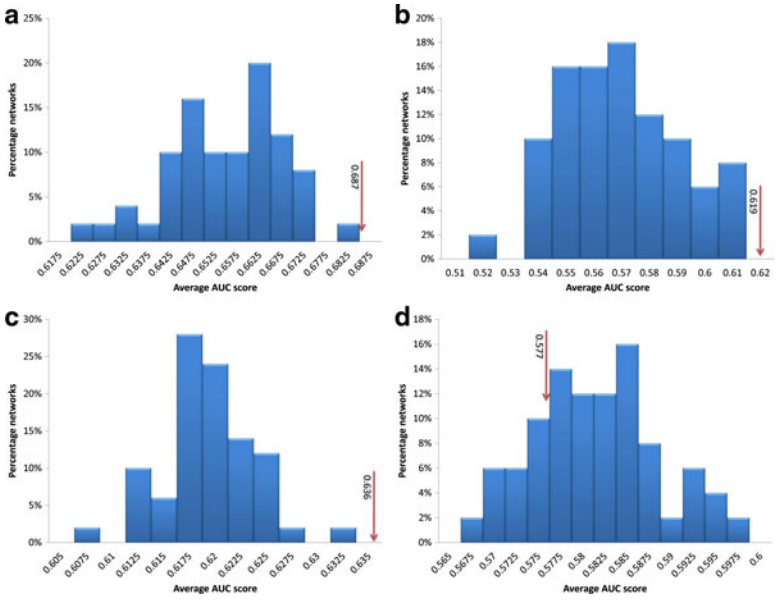

FIG. 5. Distribution of AUC scores in random networks versus real network. Results for the van 't Veer **(a)**, Ivshina **(b)**, van de Vijver **(c)**, and Wang **(d)** datasets. Each plot shows a histogram of AUC scores obtained by running the algorithm with 50 different randomized STRING networks. The red arrow denotes the average score across folds obtained by running the algorithm with the real network. (a–c) Datasets that the method exhibited improvement on. In these cases, the real network is ranked above all randomized network runs. (d) On a dataset where the method did not show improvement, the real network is ranked 35 among the 50 randomized networks.

TABLE 4. PERFORMANCE COMPARISON

Dataset	*No. of Genes*	*Classifier*			
		NetProp	*HyperGene*	*SVM*	*NICK*
Ivshina	25	0.6289	*0.4893*	0.6336	**0.6665**
	50	**0.6756**	*0.5352*	0.6725	0.6327
	100	0.6238	*0.6015**	**0.6366**	0.6103
	250	0.6034	**0.6114***	0.6098	*0.5904*
	500	0.6184	*0.5606**	0.6084	**0.6266**
Wang	25	0.6466	*0 .6456*	**0.6592**	0.6562
	50	0.6398	*0.6123*	0.6713	**0.6732**
	100	0.6609	*0.5958*	0.6886	**0.6918**
	250	0.6782	*0.6007*	**0.6937**	0.6861
	500	**0.6792**	*0.5805**	0.6584	0.6623
van de Vivjer	25	0.7225	0.7224	*0.7186*	**0.7257**
	50	**0.7268**	0.7196	0.7114	*0.6788*
	100	**0.7316**	*0.6977*	0.7262	0.7208
	250	0.743	*0.6757*	0.755	**0.7564**
	500	0.7456	*0.7023*	0.7508	**0.7578**
van 't Veer	25	**0.8452**	0.7738	*0.7381*	0.7857
	50	**0.8333**	*0.7976*	0.8095	0.8214
	100	**0.8452**	*0.7143*	0.8214	0.8333
	250	0.8214	*0.8095*	**0.8333**	0.8214
	500	*0.8214*	**0.869**	0.8333	0.8333

Area under ROC curve results of four algorithms with four breast cancer datasets. The table shows the AUC average of fivefold cross validation for Ivshina, Wang, and van de Vijver datasets, and an AUC for a single run on the original training and test set from van 't Veer based on data compiled by Hwang et al. Numbers in bold (italics) indicate the highest (lowest) score among the four algorithms in each row.

*Incomplete runs of the HyperGene algorithm aborted by the quadratic programming solver.

2.5. Comparison to other methods

We compared the performance of NICK to the HyperGene algorithm (Hwang et al., 2008) and to two algorithms that do not use network information: a linear kernel SVM as used by NICK, and NetProp, a network propagation algorithm (Zhou et al., 2006) as used by HyperGene (for HyperGene we used a MATLAB®; implementation kindly provided to us by the authors). For the comparison, we used four breast cancer datasets: van 't Veer (van 't Veer et al., 2002), van de Vijver (van de Vijver et al., 2002), Ivshina (Ivshina et al., 2006), and Wang (Wang et al., 2005).

We compared the algorithms with feature sets of different sizes ranging from 25 to 500 genes. Table 4 and Figure 6 summarize the results. In order to limit the running time of HyperGene, its authors set a threshold of 10,000 iterations for each internal optimization routine. In some cases, the quadratic programming solver exceeded the above threshold during an internal iteration of the HyperGene algorithm and thus failed to find an optimal solution before optimization process was finished. The HyperGene score in these cases could be low due to the incomplete optimization.

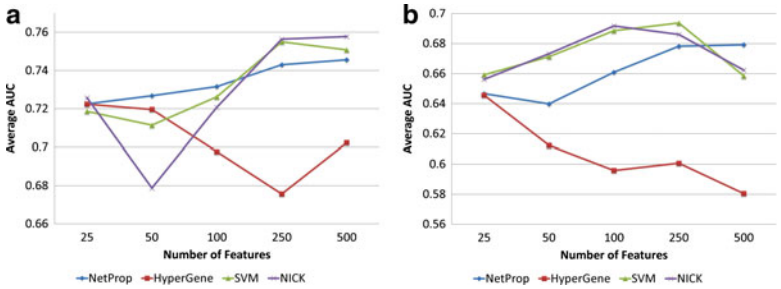

FIG. 6. Performance comparison. The figure presents the data in Table 4 comparing four different algorithms on two breast cancer datasets: **(a)** van de Vijver. **(b)** Wang. For full details, see Table 4.

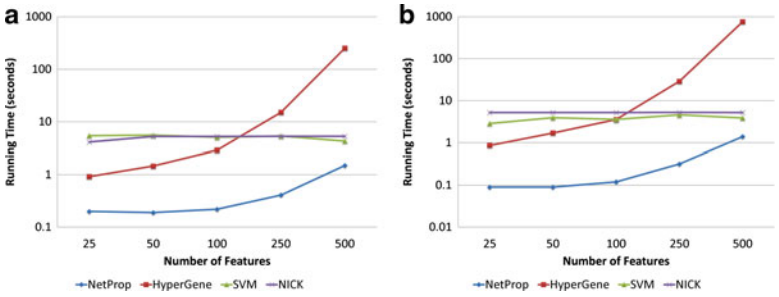

FIG. 7. Running time comparison. The figure shows running times of the four algorithms on different datasets with different number of features. Time is displayed in log scale (seconds). **(a)** van de Vijver. **(b)** Wang.

NICK ranks first in 7 of the 20 cases tested, the net propagation algorithm ranks first in 7 others, SVM ranks first in 4 cases, and HyperGene ranks first in two cases. HyperGene ranks last in 15 of the 20 cases. Notably, NICK's average rank is 1.5, compared to 2.25 ,3.5 and 2.75 for NetProp, HyperGene, and SVM, respectively. While the gaps between the best and second best scores are sometimes very small, the gap between the best and worst scores is often quite large. Note also that the number of features giving the highest score is not the same in different datasets.

We ran the four algorithms on a single core of 2-quad core Intel Xeon 5160 at 2.33 Ghz with 16GB memory running 64 bit Linux using MathWorks MATLAB version 7.2. Figure 7 shows a comparison of running times. Times include preprocessing and training on a single fold. Clearly, NICK shows a dramatic advantage over HyperGene and NetProp when the number of features grows. Both HyperGene and NICK require some preprocessing of the data using matrix operations. Following this preprocessing, NICK simply runs plain linear SVM, while HyperGene runs an iterative process solving a number of quadratic programming problems. For example, using 500 features on the Ivshina dataset, with a network of 2, 000 nodes and 9, 914 edges, NICK takes less than 5 seconds to run a single fold, which is about 250 times faster than HyperGene. NICK and SVM take roughly constant time, while HyperGene and NetProp show running times growing exponentially with the feature set size.

2.6. *Discussion*

Due to the difficulty of classifying disease expression profiles, it was suggested to integrate prior knowledge encapsulated in gene networks into the analysis process. The basic assumption behind this suggestion is that gene networks contain added information about gene expression, which can assist in the classification. Our analysis validates this assumption experimentally, showing that network proximity correlates with higher level of co-expression. On the other hand, we showed that not all extant algorithms truly make use of this network information in their analysis, and sometimes the same improvement over choosing independent genes as features is obtainable using randomized networks.

In this study, we introduced NICK, a kernel based on the network topology. In addition to its use in kernel methods, it can also be used as a linear transformation of the input in settings that do not involve kernels. Given a graph (or any non-negative similarity matrix), obtaining the NICK kernel matrix is straightforward. The presented decomposition of the kernel matrix allows for reducing the original problem to the standard SVM problem by simply performing a linear transformation on the data. After the transformation, any SVM implementation can be used. The method does not involve any search procedure over the network. It is very fast, and scales well with the network size and the number of features. Compared to the HyperGene algorithm and to basic SVM, NICK usually shows better classification quality.

Although SVM is a supervised classification algorithm, which is trained on labeled data, eventually the kernel and transformation do not depend on the class labels of the samples. In fact, they depend only on the network itself, while the data and labels information is restricted to the constraints of the optimization problem. Interpretation of the transformation matrix is quite straightforward. It is very clear how a meta-feature is constructed from its neighbor features, and the network topology is directly reflected in the weights of original features in the meta-feature. Since the matrices are independent of the data, the transformation and kernel can be used towards unsupervised methods such as clustering or unsupervised feature selection and extraction as well. In particular, methods that use $||\mathbf{w}||_2^2$ regularization, such as Ridge regression (Hoerl and Kennard, 1970) can justifiably use our regularization term or kernel.

NICK has several limitations. One obvious limitation (which holds for any PPI-based approach) is the network quality. PPI networks are known to be incomplete and error-prone (Chua et al., 2006). In addition, most network edges originated from *in vitro* experiments, which may differ from in-vivo conditions. Also, semantically, networks are compiled from pairwise relations, and it is hard to interpret paths and topology of the whole network, as different conditions may yield different sets of edges.

NICK is based on the assumption that close genes should contribute similarly to the classification model. In some cases, the opposite may be true (e.g., when one protein suppresses the function of another protein). Also, the leap from mRNA co-expression to protein interactions in the network level is not trivial, and as we have shown, the two signals are linked in a highly significant way, but linkage is not very strong.

The NICK transformation matrix has some limitations. The first is due to the global nature of the transformation. A single meta-feature can be a weighted average of all features in its connected component, so even distant features contribute to the meta-feature's value, which may not reflect the true biology. Also, the transformation matrix is triangular, which poses two problems in interpreting it. The first is that the meta-feature corresponding to the k-th feature (column) includes original features $\{k, k+1, \ldots, n\}$ in its connected component, and as we advance in the columns of the matrix, meta-features corresponding to the columns include less and less features. Hence, meta-features are highly overlapping (in terms of original features they contain) with a large variability in size. This makes every meta-feature by itself hard to interpret biologically. Second, the transformation depends on the order of the nodes in the initial adjacency matrix.

Finally, the transformation does not reduce the dimension of the data, neither by selecting a subset of the original features, nor by extracting a small number of new meta-features. The number of meta-features is identical to the number of original features. Hence, it does not directly allow feature selection.

2.6.1. When is the network informative? We compared NICK to two algorithms that use network data for classification purposes. In one of them (PinnacleZ), the biological information in the network apparently did not contribute to the performance improvement, while in the other (HyperGene) it did. Differences among the methods (e.g., in network size, edge definition, and the algorithm that utilizes the network) may explain this phenomenon and require further study.

The lower performance improvement on some datasets than on others is not fully explainable yet. It could be due to different measurement technology (notably, the two datasets that obtained the best results were profiled using custom Rosetta cDNAs), due to difference in sample purity in different cancer types, or due to uneven representation within the network of the pathways involved in different cancers. In fact, we observed positive—yet mild—correlation between the AUC score obtained for a dataset and the level of coexpression of neighboring genes in the networks on that dataset ($r^2 = 0.36$), which indeed hints to possible impact of the network on the classification accuracy. This question requires further study on additional datasets. Nevertheless, in datasets that did not exhibit improvement the network did not worsen the results. Remarkably, when performance improved, the improvement was achieved even with low relative weight (β) for the network information.

3. METHODS

3.1. Testing network informativeness

Let **x** and **y** be vectors of expression measurements of two genes over a given set of samples. To measure the level of co-expression of the two genes, we used Pearson correlation between **x** and **y**. To account for both negative and positive correlation we used the absolute value of the Pearson correlation.

In order to test for network informativeness with respect to expression data, we first grouped pairs of genes into different populations according to the pairs' connectivity within the network. As a baseline, we sampled random pairs from the population of all gene pairs that are neighbors in the network, comparing the distribution of correlations to non-neighbors (distant) pairs. We also looked at more specific populations of pairs according to their distance: pairs that are 3, 4, 5, and 6 nodes away, and pairs that are 7 or more nodes away, including nodes that have no path between them.

Distance is a highly local measure and does not take into account the existence of multiple paths between nodes. We were also interested in a connectivity measure that will take into account multiple connections, under the assumptions that due to the noisy nature of large-scale networks, multiple paths strengthen the

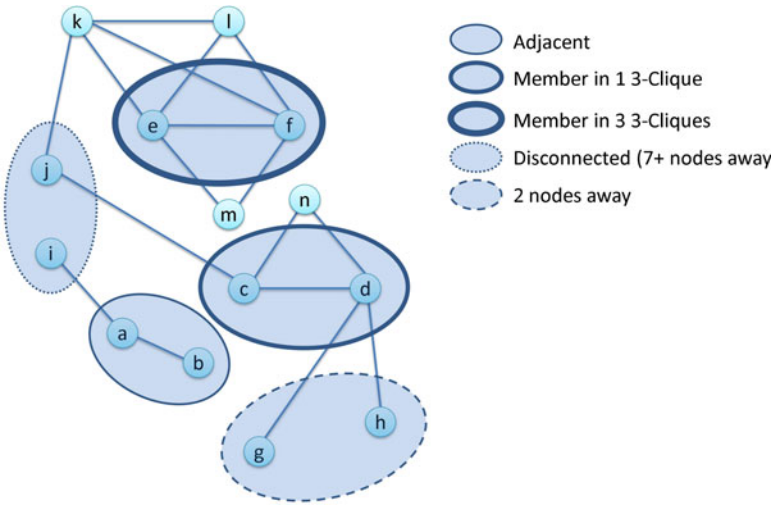

FIG. 8. Toy example of gene pair populations. Among the adjacent genes in the network are (a,b), (c,d), and (e,f). Pair **(g,h)** is 2 nodes away, and pair **(i,j)** is not connected at all and thus considered as 7+ nodes away. Pair **(c,d)** is considered more connected than pair **(a,b)**, as it is also a member of a 3-clique formed by node **n**. This pair is described as adjacent member in one 3-clique. Pair **(e,f)** has an even higher connectivity, and is described as adjacent which is also a member in three 3-cliques, using the three 3-cliques formed with the three nodes marked **k**, **l**, and **m**.

confidence in the relation between two nodes in the network. To this end, we looked at few additional gene-pairs populations: adjacent genes to those that are also members in one or more 3-cliques. We gathered those that are members in 2, 3, 4, and 5 3-cliques. A toy example illustrating the different gene-pair populations samples can be seen in Figure 8.

We compared the mean absolute Pearson correlation within each population to the baseline population using *t*-test.

3.2. Testing for network impact

We wanted to test whether the performance of different algorithms is influenced by the real network's topology. To this end, we ran the algorithms using both real and random networks. We randomized the networks by permuting the network node's names such that it will maintain its topology, but lose any correlation that it might have had with the expression data. For each algorithm, we repeated the test with 50 different network permutations, and with 50 runs of the original network. We reported the average AUC score using fivefold cross validation (Kohavi, 1995) using the real network, and an average AUC score using fivefold cross validation for each network permutation.

There are two elements of randomness in In PinnacleZ (Chuang et al., 2007) that required us to run the original algorithm multiple times for comparison. The first is the significance tests: Although the algorithm is deterministic and will always find the same subnetwork starting from a specific seed, the calculation of significance level of the resulting subnetwork is based on sampling and hence may be different every time. The second source of randomness is due to the different folds used to measure the classification performance.

3.3. Derivation of NICK

For simplicity we start from the standard linearly separable SVM formulation of Vapnik (1999):

$$\min_{\mathbf{w},\, w_0} \left\{ \frac{1}{2} ||\mathbf{w}||^2 \right\} \tag{1}$$

subject to

$$(\mathbf{w}^T \mathbf{x}_i + w_0) \cdot y_i \geq 1 \qquad \text{for any } i = 1, \ldots, n$$

Here, $\mathbf{x}_i$ is the gene expression vector (or feature vector) representing the i'th sample and y_i is the i'th sample's label, $y_i \in \{-1, 1\}$. The number of coordinates of $\mathbf{x}_i$ will be denoted by p.

Let $\mathbf{A}$ be a symmetric $p \times p$ matrix with non-negative entries, where $\mathbf{A}_{i,j}$ stands for the similarity level between genes i and j and $\mathbf{A}_{i,i} = 0$. In order for the weights to be closer for genes that are more similar, we wish to minimize the following mean square pairwise difference expression:

$$\frac{1}{2}\sum_{j=1}^{p}\sum_{k=j+1}^{p}(\mathbf{A}_{j,k}(w_j - w_k)^2)$$

We add this expression to the objective function, introducing a non-negative tradeoff parameter $\beta \geq 0$:

$$\min_{\mathbf{w}, w_0}\left\{\frac{1}{2}||\mathbf{w}||^2 + \frac{1}{2}\beta\sum_{j=1}^{p}\left(\sum_{k=j+1}^{p}(\mathbf{A}_{j,k}(w_j - w_k)^2)\right)\right\} \tag{2}$$

Let $\widetilde{A}$ be a $p \times p$ diagonal matrix with the sum of row j of $\mathbf{A}$ at $A_{j,j}$, $\widetilde{\mathbf{A}}_{j,k} = \delta_{j,k}\sum_l \mathbf{A}_{j,l}$. Here $\delta_{j,k}$ is the Kronecker delta with $\delta_{j,k} = 1$ if $j = k$ and $\delta_{j,k} = 0$ otherwise.

Following Beineke and Wilson (2004), the matrix notation of Equation 2 is:

$$\min_{\mathbf{w}, w_0}\left\{\frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{1}{2}\beta\mathbf{w}^T(\tilde{\mathbf{A}} - \mathbf{A})\mathbf{w}\right\} \tag{3}$$

Note that for a simple adjacency matrix based on a graph, where $\mathbf{A}_{i,j} = 1$ if i and j are adjacent and $\mathbf{A}_{i,j} = 0$ if they are not, $\widetilde{A}$ is a diagonal matrix with $\widetilde{A}_{i,i}$ being the degree of node i, and $\mathbf{B} = \widetilde{\mathbf{A}} - \mathbf{A}$ is known as the *Laplacian Matrix* of the graph (Cvetkovic et al., 1998). The newly added term captures the assumption that close genes are more likely to have similar expression and thus to similarly contribute to the learned classification model.

A solution to the original SVM quadratic programming problem is obtained by transforming the optimization problem to the dual form. We introduce Lagrange multipliers $\alpha_1, \ldots, \alpha_n$, $\alpha_i \geq 0$, one for each constraint (corresponding to a single sample point). The primal Lagrangian is:

$$\begin{aligned} L_P &= \frac{1}{2}\mathbf{w}^T\mathbf{w} + \frac{1}{2}\beta\mathbf{w}^T\mathbf{B}\mathbf{w} - \sum_{i=1}^{n}\alpha_i((\mathbf{x}_i^T\mathbf{w} + w_0)y_i - 1) \\ &= \frac{1}{2}\mathbf{w}^T(\mathbf{I} + \beta\mathbf{B})\mathbf{w} - \sum_{i=1}^{n}\alpha_i((\mathbf{x}_i^T\mathbf{w} + w_0)y_i - 1) \end{aligned} \tag{4}$$

In order to reach the same solution of (3), we need to find a saddle point of L_P where it is minimized with respect to $\mathbf{w}$, w_0 and maximized with respect to the Lagrange multipliers $\alpha_1, \ldots, \alpha_n$. We differentiate L_P with respect to w_0 to get:

$$\frac{\partial L_P}{\partial w_0} = \sum_{i=1}^{n}\alpha_i y_i$$

and set it equal to 0 to get:

$$\sum_{i=1}^{n}\alpha_i y_i = 0 \tag{5}$$

We differentiate L_P with respect to $\mathbf{w}$ to get:

$$\frac{\partial L_P}{\partial \mathbf{w}} = (\mathbf{I} + \beta\mathbf{B})\mathbf{w} - \sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i$$

and again set it equal to 0 to get:

$$\mathbf{w} = (\mathbf{I} + \beta\mathbf{B})^{-1}\sum_{i=1}^{n}\alpha_i y_i \mathbf{x}_i \tag{6}$$

Notice that $\mathbf{I} + \beta\mathbf{B}$ is a positive definite matrix by construction, hence its inverse is well defined and unique. By substituting $\mathbf{w}$ into (4), we get the following dual optimization problem:

$$\max_{\alpha}\left\{\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\alpha_i\alpha_j y_i y_j \mathbf{x}_i^T(\mathbf{I} + \beta\mathbf{B})^{-1}\mathbf{x}_j\right\} \tag{7}$$

subject to

$$\sum_{i=1}^{n} \alpha_i y_i = 0 \qquad \text{for any } i = 1, \ldots, n$$

$$\alpha_i \geq 0 \qquad \text{for any } i = 1, \ldots, n$$

As the matrix $\mathbf{I} + \beta\mathbf{B}$ is both positive definite and symmetric, $(\mathrm{I} + \beta\mathbf{B})^{-1}$ can be decomposed using *Cholesky decomposition* (Golub and Van Loan, 1996): there exists an lower-triangular matrix $\mathbf{L}$ such that $(\mathrm{I} + \beta\mathbf{B})^{-1} = \mathbf{L}\mathbf{L}^T$. Plugging $\mathbf{L}\mathbf{L}^T$ into (7) yields:

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i^T \mathbf{L})(\mathbf{L}^T \mathbf{x}_j) \tag{8}$$

Remarkably, this expression has exactly the same form as the dual problem for the standard SVM, with $\mathbf{x}'_i = \mathbf{L}^T \mathbf{x}_i$. It is possible, then, to perform a linear transformation of the sample vectors $\mathbf{x}_i$ using $\mathbf{L}$ to obtain a set of transformed samples where $\mathbf{x}_i^{new} = \mathbf{x}_i^T \mathbf{L}$. Now we can run the regular SVM optimization procedure in order to learn a model of the transformed samples. In order to classify a new unseen sample, we should first transform it in the same manner, using $\mathbf{L}$ and then use the trained model to classify it. We note that although we derived the result (8) for the linearly separable case, one gets an identical expression (albeit with slightly different constraints on the Lagrange multipliers α_i), in the soft margin setting (Cortes and Vapnik, 1995).

3.3.1. Integration of different information sources. We comment that the same formalism described above can naturally integrate different sources of information about relations between genes. Indeed, given several gene networks (e.g. PPI network, metabolic networks, signaling networks, and similarities based on GO annotation), one needs to represent each network as a matrix with nonnegative elements, $\mathbf{A}^s$, such that $\mathbf{A}^s_{ij}$ represents the ''strength'' of the relations between genes i and j in network s. Now, similarly to Eq. 2, one needs to solve the following optimization problem,

$$\min_{\mathbf{w},\, w_0} \left\{ \frac{1}{2} ||\mathbf{w}||^2 + \frac{1}{2} \sum_s \beta_s \sum_{j=1}^{p} \left(\sum_{k=j+1}^{p} \left(\mathbf{A}^s_{j,k} (w_j - w_k)^2 \right) \right) \right\} \tag{9}$$

β_s are hyper parameters that control the relative contribution of the different networks. It is easy to see that the dual of Eq. 9 has the form of Eq. 8 so it can easily be solved using standard SVM tools.

Notice that to ensure that Eq. 9 is positive semi-definite, only undirected networks, associated with symmetric matrices $\mathbf{A}^s$, can be incorporated into the current formalism.

3.4. Dimension reduction

Applying our algorithm to gene expression data required dimension reduction as the data is characterized by a large feature dimension p with respect to the number of samples n (Guyon et al., 2002). We thus preprocessed our data throughout our experiments by first selecting p genes with highest variance among the samples, regardless of their labeling. We then constructed a subgraph G of the protein interaction network, containing only proteins corresponding to the selected p genes. We used $p = 2,000$ throughout our experiments and calculated our kernel and transformation matrix $\mathbf{L}$ based on G, preserving much of the original network's topology, based on genes that are relevant to the expression data.

We then selected the top k differentially expressed genes ranked by t-test. Similar to the choice of Chuang et al. (2007), we used $k = 200$ when comparing the algorithm performance with different values of β. When comparing to other methods we used different values for k ranging from 25 to 500.

Each row (and column) of $\mathbf{L}$ is associated with one gene we term as the *pivot* gene. For a given sample, $\mathbf{L}$ transforms p original feature values into p meta-feature values, each of which is a weighted average of other features' values. Note that if we generate the meta-features according to the order of the rows of $\mathbf{L}$, the i'th meta-feature would turn out to be a linear combination of exactly $p - i + 1$ features (due to the triangular form of $\mathbf{L}$). Instead of using all p meta-features, we only used k meta-features, each of which is a based on all p original features. We did this by zeroing all the rows in $\mathbf{L}$ that are not associated with the k chosen

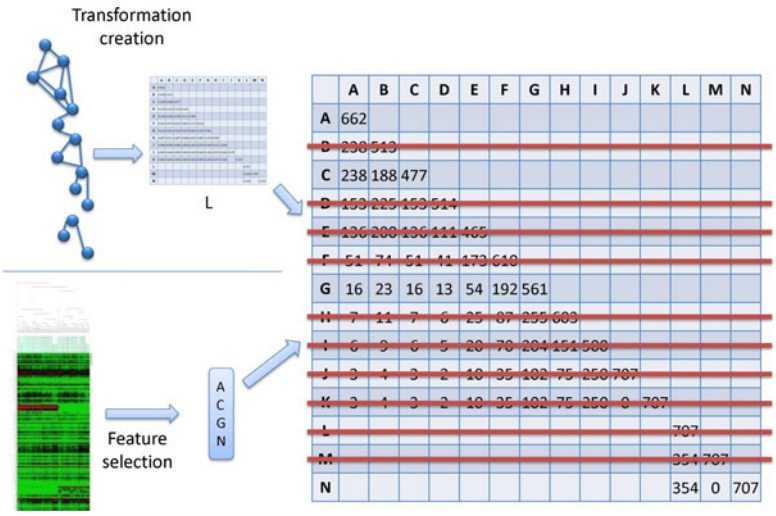

FIG. 9. Illustration of adaptation process of **L** following feature selection.

genes as illustrated in Figure 9. We named the restricted matrix $\mathbf{L}'$. During cross validation, feature selection and restriction of **L** was conducted separately for each training fold.

3.5. Model training and testing

For each dataset, we took the original expression data and transformed each expression vector $\mathbf{x}_i$ into $\mathbf{x}_i^{new}$ *by* $\mathbf{x}_i^{new} = \mathbf{x}_i^T \mathbf{L}'$. We used the transformed data for training and testing a standard linear kernel SVM estimating the performance of the output classifier by measuring the average AUC using five-fold cross validation. We repeated the process for different values of β ranging from 0.05 to 10.

ACKNOWLEDGMENTS

We would like to thank TaeHyun Hwang from Rui Kuang's laboratory (University of Minnesota) and Han-Yu Chuang from Trey Ideker's laboratory (University of California, San Diego) for permission and help in running their algorithms. This research was supported in part by the GENEPARK project which is funded by the European Commission within its FP6 Programme (contract EU-LSHB-CT-2006-037544) and by the Israel Science Foundation (grant 802/08).

DISCLOSURE STATEMENT

No competing financial interests exist.

REFERENCES

Aranda, B., Achuthan, P., Alam-Faruque, Y., et al. 2010. The IntAct molecular interaction database in 2010. *Nucleic Acids Res.* 38 (Suppl. 1) D525–D531.

Asgharzadeh, S., Pique-Regi, R., Sposto, R., et al. 2006. Prognostic significance of gene expression profiles of metastatic neuroblastomas lacking MYCN gene amplification. *J. Natl. Cancer Inst.* 98, 1193–1203.

Ashburner, M., Ball, C. A., Blake, J.A., et al. 2000. Gene ontology: tool for the unification of biology. *Nat. Genet.* 25, 25–29.

Beineke L.W., and Wilson, R.J., ed. 2004. *Topics in Algebraic Graph Theory. Volume 102 of Encyclopedia of Mathematics and its Applications.* Cambridge University Press, New York.

Bild, A.H., Yao, G., Chang, J.T., et al. 2006. Oncogenic pathway signatures in human cancers as a guide to targeted therapies. *Nature* 439, 353–357.

Chebotarev. P. 2008. Spanning forests and the golden ratio. *Discr. Appl. Math.* 156, 813–821.

Chebotarev P., and Shamis, E. 2006. The matrix-forest theorem and measuring relations in small social groups. *CoRR*, abs/math/0602070.

Chin, K., DeVries, S., Fridlyand, J., et al. 2006. Genomic and transcriptional aberrations linked to breast cancer pathophysiologies. *Cancer Cell* 10, 529–541.

Nian Chua, H., Sung, W.K., and Wong, L. 2006. Exploiting indirect neighbours and topological weight to predict protein function from protein–protein interactions. *Bioinformatics* 22, 1623–1630.

Chuang, H.-Y.Y., Lee, E., Liu, Y.-T.T., et al. 2007. Network-based classification of breast cancer metastasis. *Mol. Syst. Biol.* 3.

Cortes C., and Vapnik, V.1995. Support-vector networks. *Mach. Learn.* 20, 273–297.

Cvetkovic, D.M., Doob, M., Sachs, H., et al. 1998. *Spectra of Graphs: Theory and Applications,* 3rd ed. Vch Verlagsgesellschaft Mbh. Berlin.

Ein-Dor, L., Kela, I., Getz, G., et al. 2005. Outcome signature genes in breast cancer: is there a unique set? *Bioinformatics* 21, 171–178.

Golender, V.E., Drboglav, V.V., and Rosenblit, A.B. 1981. Graph potentials method and its application for chemical information processing. *J. Chem. Information Comput. Sci.* 21, 196–204.

Golub G.H., and Van Loan, C.F. 1996. *Matrix Computations*, 3rd ed. Johns Hopkins University Press, Baltimore.

Guyon, I., Weston, J., Barnhill, S., et al. 2002. Gene selection for cancer classification using support vector machines. *Mach. Learn.* 46, 389–422.

Herschkowitz, J., Simin, K., Weigman, V., et al. 2007. Identification of conserved gene expression features between murine mammary carcinoma models and human breast tumors. *Genome Biol.* 8, R76.

Hoerl A.E., and Kennard, R.W. 1970. Ridge regression: biased estimation for nonorthogonal problems. *Technometrics* 12, 55–67.

Hwang, T., Tian, Z., Kuang, R., et al. 2008. Learning on weighted hypergraphs to integrate protein interactions and gene expressions for cancer outcome prediction. *Proc 8th IEEE Int. Conf. Data Mining (ICDM '08)* 293–302.

Ivshina, A.V., George, J., Senko, O., et al. 2006. Genetic reclassification of histologic grade delineates new clinical subtypes of breast cancer. *Cancer Res.* 66, 10292–10301.

Jansen, R., Greenbaum, D., and Gerstein, M. 2002. Relating whole-genome expression data with protein-protein interactions. *Genome Res.* 12, 37–46.

Jensen, L.J., Kuhn, M., Stark, M., et al. 2009. String 8–a global view on proteins and their functional interactions in 630 organisms. *Nucleic Acids Res.* 37, 412–416.

Kanehisa M., and Goto, S. 2000. KEGG: Kyoto Encyclopedia of Genes and Genomes. *Nucleic Acids Res.* 28, 27–30.

Kerrien, S., Alam-Faruque, Y., Aranda, B., et al. 2007. IntAct—open source resource for molecular interaction data. *Nucleic Acids Res.* 35, 561–565.

Kohavi, R. 1995. A study of cross-validation and bootstrap for accuracy estimation and model selection. *IJCAI* 1137–1143.

Larsen, J.E., Pavey, S.J., Passmore, L.H., et al. 2007. Expression profiling defines a recurrence signature in lung squamous cell carcinoma. *Carcinogenesis* 28, 760–766.

Lee, E.S., Son, D.S., Kim, S.H., et al. 2008a. Prediction of recurrence-free survival in postoperative non small cell lung cancer patients by using an integrated model of clinical information and gene expression. *Clin. Cancer Res.* 14, 7397–7404.

Lee, E., Chuang, H.Y., Kim, J.W., et al. 2008b. Inferring pathway activity toward precise disease classification. *PLoS Comput. Biol.* 4, e1000217.

Merris, R. 1997. Doubly stochastic graph matrices. *Publ. Elektrotech. Fak. Univ. Beograd.* 8, 64–71.

Merris, R. 1998. Doubly stochastic graph matrices. II. *Linear Multilinear Algebra* 45, 275–285.

Nitsch, D., Tranchevent, L.C., Thienpont, B., et al. 2009. Network analysis of differential expression for the identification of disease-causing genes. *PLoS ONE* 4, e5526.

Paik, S., Tang, G., Shak, S., et al. 2006. Gene expression and benefit of chemotherapy in women with node-negative, estrogen receptor-positive breast cancer. *J. Clin. Oncol.* 24, 3726–3734.

Pawitan, Y., Bjohle, J., Amler, L., et al. 2005. Gene expression profiling spares early breast cancer patients from adjuvant therapy: derived and validated in two population-based cohorts. *Breast Cancer Res.* 7, R953–R964.

Phillips, H.S., Kharbanda, S., Chen, R., et al. 2006. Molecular subclasses of high-grade glioma predict prognosis, delineate a pattern of disease progression, and resemble stages in neurogenesis. *Cancer Cell* 9, 157–173.

Rapaport, F., Zinovyev, A., Dutreix, M., et al. 2007. Classification of microarray data using gene networks. *BMC Bioinform.* 8, 35.

Raponi, M., Zhang, Y., Yu, J., et al. 2006. Gene expression signatures for predicting prognosis of squamous cell and adenocarcinomas of the lung. *Cancer Res.* 66, 7466–7472.

Rual, J.F., Venkatesan, K., Hao, T., et al. 2005. Towards a proteome-scale map of the human protein–protein interaction network. *Nature* 437, 1173–1178.

Snel, B., Lehmann, G., Bork, P., et al. 2000. String: a web-server to retrieve and display the repeatedly occurring neighbourhood of a gene. *Nucleic Acids Res.* 28, 3442–3444.

Sørlie, T., Tibshirani, R., Parker, J., et al. 2003. Repeated observation of breast tumor subtypes in independent gene expression data sets. *Proc. Nat. Acad. Sci USA* 100, 8418–8423.

Tian, L., Greenberg, S.A., Kong, S.W., et al. 2005. Discovering statistically significant pathways in expression profiling studies. *Proc. Nat. Acad. Sci USA* 102, 13544–13549.

Tian, Z., Hwang, T., and Kuang, R. 2009. A hypergraph-based learning algorithm for classifying gene expression and arraycgh data with prior knowledge. *Bioinformatics* 25, 2831–2838.

van de Vijver, M.J., He, Y.D., van 't Veer, L.J., et al. 2002. A gene-expression signature as a predictor of survival in breast cancer. *N. Engl. J. Med.* 347, 1999–2009.

van 't Veer, L.J., Dai, H., van de Vijver, M.J., et al. 2002. Gene expression profiling predicts clinical outcome of breast cancer. *Nature* 415, 530–536.

Vapnik, V. 1999. *The Nature of Statistical Learning Theory (Information Science and Statistics)*, 2nd ed. Springer, New York.

Wang, Y., Klijn, J.G., Zhang, Y. et al. 2005. Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer. *Lancet* 365, 671–679.

Wei, P., and Pan, W. 2008. Incorporating gene networks into statistical tests for genomic data via a spatially correlated mixture model. *Bioinformatics* 24, 404–411.

Wei Z., and Li, H. 2007. A Markov random field model for network-based analysis of genomic data. *Bioinformatics* 23, 1537–1544.

Zhou, D., Huang, J., and Schölkopf, B. 2006. Learning with hypergraphs: clustering, classification, and embedding. *Adv. NIPS* 19, 1601–1608.

Zhu, Y., Shen, X., and Pan, W. 2009. Network-based support vector machine for classification of microarray samples. *BMC Bioinform.* 10, S21.

Address Correspondence to:
Ofer Lavi
Blavatnik School of Computer Science
Tel Aviv University
Tel Aviv, 69978 Israel

E-mail: oferl@il.ibm.com