ABSTRACT

Motivation: Incorporating biological prior knowledge into predictive models is a challenging data integration problem in analyzing high-dimensional genomic data. We introduce a hypergraph-based semi-supervised learning algorithm called HyperPrior to classify gene expression and array-based comparative genomic hybridization (arrayCGH) data using biological knowledge as constraints on graph-based learning. HyperPrior is a robust two-step iterative method that alternatively finds the optimal labeling of the samples and the optimal weighting of the features, guided by constraints encoding prior knowledge. The prior knowledge for analyzing gene expression data is that cancer-related genes tend to interact with each other in a protein–protein interaction network. Similarly, the prior knowledge for analyzing arrayCGH data is that probes that are spatially nearby in their layout along the chromosomes tend to be involved in the same amplification or deletion event. Based on the prior knowledge, HyperPrior imposes a consistent weighting of the correlated genomic features in graph-based learning.

Results: We applied HyperPrior to test two arrayCGH datasets and two gene expression datasets for both cancer classification and biomarker identification. On all the datasets, HyperPrior achieved competitive classification performance, compared with SVMs and the other baselines utilizing the same prior knowledge. HyperPrior also identified several discriminative regions on chromosomes and discriminative subnetworks in the PPI, both of which contain cancer-related genomic elements. Our results suggest that HyperPrior is promising in utilizing biological prior knowledge to achieve better classification performance and more biologically interpretable findings in gene expression and arrayCGH data.

Availability: http://compbio.cs.umn.edu/HyperPrior

Contact: kuang@cs.umn.edu

Supplementary information: Supplementary data are available at bioinformatics online.

1 INTRODUCTION

In the past decade, numerous cancer researchers have actively investigated high-throughput genomic data to reveal the molecular mechanisms underlying cancer development and progression. DNA copy number variations (CNVs) measured by array-based comparative genomic hybridization (arrayCGH), and microarray gene expressions are among the most widely studied high-throughput data (Sawyers, 2008). Microarray gene expressions provide a genome-wide quantification of messenger RNA abundance, while arrayCGH data quantify the events of amplification or deletion of large DNA segments on chromosomes. A large number of high-resolution arrayCGH datasets and gene expression datasets were generated to study many different cancers (Glinsky et al., 2004; Pole et al., 2006; Tonon et al., 2005; van’t Veer et al., 2002). In these studies, two major objectives were (i) to detect highly discriminative chromosomal copy number aberration regions or gene expression patterns as biomarkers of cancer-relevant phenotypes; and (ii) to build reliable predictive models based on the biomarkers for cancer sample classification. Although many interesting and promising findings were reported in these studies, concerns have been raised on the unstable and inconsistent results in cross-validations and cross-platform comparisons due to the relatively small sample sizes in the studies (Dupuy and Simon, 2007). To address the problem, researchers have proposed including other complementary genomic information such as pathways or functional annotations to aid model building and biomarker discovery. It is believed that the prior knowledge from complementary data can generate more robust models and more consistent discoveries across independent studies. For gene expression profiles, the availability of large protein–protein interaction networks (PPI networks), which contain information on gene functions, pathways and modularity of gene regulations, provides a desirable source of data for the purpose (Aragues et al., 2008; Chuang et al., 2007; Rapaport et al., 2007). In arrayCGH data, microarray comparative genomic hybridization measures copy number information distributed along the genome at different resolutions. This information typically includes thousands of spot intensities. Intuitively, neighboring spots on the chromosomes tend to be highly correlated because a DNA aberration can expand to neighboring intervals (Rapaport et al., 2008). However, designing a unified strategy to integrate gene expressions with protein–protein interactions or to integrate arrayCGH data with the chromosomal spatial information is still a challenging data integration problem, since standard classification and feature selection methods do not meet the complexity of a joint learning on two different data types.

In this article, we propose a hypergraph-based iterative learning algorithm called HyperPrior to integrate genomic data with general
connected by the hyperedges. Samples tend to have similar states of features and thus are highly weighted. To model the genomic data as a hypergraph, each feature is represented by two hyperedges corresponding to two states of the feature. In arrayCGH data, each spot intensity is represented by two hyperedges as a prior graph. The relations define the constraints for hypergraph-based learning. Note that each gene expression corresponds to two features (up/downregulations). The two features of a gene will have a relation with the two features of another gene interacting with the first gene. For example, gene 1 and gene 2 are both associated with both cancer and biomarker identification. The HyperPrior algorithm minimizes a cost function under a unified regularization framework. This framework elegantly takes biological prior knowledge (e.g. a correlation structure of spot regions or a protein–protein interaction network) as constraints on a hypergraph built from genomic data. HyperPrior is a natural extension of label propagation algorithms on hypergraphs (Agarwal et al., 2006; Zhou et al., 2006). This algorithm helps to handle the problem of learning optimal weighting of the hyperedges, while all other methods assume a uniform weighing. To model the genomic data as a hypergraph, each sample is denoted by a vertex and each feature is denoted by two hyperedges corresponding to two states of the feature. In arrayCGH data, each spot intensity is represented by two hyperedges labeled as ‘amplification’ and ‘deletion’ of the associated DNA segment, respectively. In gene expression data, each gene expression value is denoted by two hyperedges labeled as ‘upregulated’ and ‘downregulated’, respectively. The hyperedges categorize samples by the two different states of features in the genomic data (Fig. 1A). Our cluster assumption on the hypergraph is that the same type of samples tend to have similar states of features and thus are highly connected by the hyperedges. HyperPrior formulates optimization problems as learning labels and hyperedge weights together with the assignment of edge weights constrained by the biological relation between the genomic features (Fig. 1B). Specifically, HyperPrior attempts to find a weighting of hyperedges that balances both the two-class separation on the hypergraph and the consistency with the biological constraints. The assumption is that neighboring spots or genes interacting with each other are more likely to receive similar weights. The resultant weights on the genomic features are used to discover biologically interpretable biomarkers. Specifically, the highly weighted DNA aberration regions may suggest cancer-relevant DNA amplification and deletion events, and the highly weighted genes in densely connected subnetworks in a protein–protein interaction network may suggest relevant cancer pathways.

2 RELATED WORK

Integrating multiple genomic data types for building predictive models or selecting features has been an important research focus in bioinformatics since several years ago (Barutcuoglu et al., 2006; Tsuda et al., 2005). For example, Li et al. (2006) proposed a two-layered Bayesian network approach to integrate relations from gene expressions, biological literature and gene sequences into a genome-wide functional network. Zhao et al. (2008) proposed a novel method to integrate gene–gene (or mRNA) relations and sample relations for gene selection. This method focuses on discovering geometric patterns to select genes with biological relevance and statistical significance.

Under this category, several other general computational methods were also proposed to use prior knowledge in classifying arrayCGH data and gene expression data (Chuang et al., 2007; Li and Li, 2008; Rapaport et al., 2007, 2008). Rapaport et al. (2008) proposed a novel method to integrate gene–gene (or mRNA) relations and sample relations for gene selection. This method focuses on discovering geometric patterns to select genes with biological relevance and statistical significance.

In their approach, the integration of gene expressions and protein–protein interactions is achieved by two independent procedures: discriminative subnetworks are first identified from the PPI network and the subnetworks are then used as features to predict cancer metastasis. Rapaport et al. (2007) proposed a similar method, which first computes the spectral graph structure of a gene network, and then uses the spectral graph structure to smooth microarray gene
Hypergraph-based learning with prior knowledge

expressions used before for sample classification. Aragues et al. (2008) proposed a statistical method to score genes by several measures including their degree in a cancer-specific interaction network, their differential expressions in microarray data and their structural, functional and evolutionary properties. Li and Li (2008) proposed to add a graph Laplacian constraint to the L1-norm linear regression model in a general regularization framework. The graph Laplacian encodes a network of known KEGG pathways. The new model can be efficiently solved by methods for lasso-type problems.

HyperPrior is different from the previous methods in both problem formulation and model implication. The regularization framework of HyperPrior is designed for simultaneously classifying samples and selecting features based on prior knowledge. HyperPrior explores the cluster structures in both the hypergraph and the prior graph in one unified learning framework. Thus, the learning problem is a combination of semi-supervised learning and variable selection. In Section 3.1.5, we will also show that the core of HyperPrior is to learn a “kernel” for a diagonal linear transformation of the data along with learning labels on samples. Thus, HyperPrior can be interpreted as a novel semi-supervised and relaxed wrapper-feature-selection method constrained by prior knowledge.

3 METHODS

In this section, we first introduce the HyperPrior algorithm and its regularization framework. We then describe how to model arrayCGH data with spatial prior knowledge and gene expression data with prior knowledge in a protein-protein interaction network by constrained hypergraphs.

3.1 HyperPrior algorithm

3.1.1 Notations

A hypergraph is a special graph that contains hyperedges. In a normal graph, each edge connects a pair of vertices, but in a hypergraph each edge can connect an arbitrary number of vertices in the graph. Let \( V = \{v_1, v_2, \ldots, v_n\} \) be a set of vertices and \( E = \{e_1, e_2, \ldots, e_m\} \) be a set of hyperedges defined on \( V \): for any hyperedge \( e \in E \), \( e = \{v_i, v'_i, \ldots, v''_i\} \), where \( \{v_i, v'_i, \ldots, v''_i\} \) is a subset of \( V \). A hyperedge \( e \) and a vertex \( v \) are called incident if \( v \in e \). A non-negative real number (a weight) can be assigned to each hyperedge by a function \( v \) (we can also be defined as a vector variable and we will use both notations interchangeably). The vertex function can also be defined as \( d(e) = \sum_{v \in e} h(v, e) \), which is the number of vertices incident with \( e \). The degree of a hyperedge \( e \) is defined as \( d(e) = \sum_{v \in e} h(v, e) \), which is the number of vertices incident with \( e \). We define diagonal matrices \( D_1 = diag(h(v_1), h(v_2), \ldots, h(v_n)) \) and \( D_2 = diag(h(e_1), h(e_2), \ldots, h(e_m)) \).

The hypergraph \( G(V, E) \) is a \( |V| \times |E| \) matrix with elements defined as \( h(v, e) = 1 \) (or a real value if \( H \) is weighted), when \( v \in e \) and 0 otherwise. The degree of a vertex \( v \) is defined as \( d(v) = \sum_{e \in E} h(v, e) \), which is the (weighted) sum of the weights of the hyperedges incident with \( v \). The degree of a hyperedge \( e \) is defined as \( d(e) = \sum_{v \in e} h(v, e) \), which is the number of vertices incident with \( e \). We define diagonal matrices \( D_1 = diag(h(v_1), h(v_2), \ldots, h(v_n)) \) and \( D_2 = diag(h(e_1), h(e_2), \ldots, h(e_m)) \).

Let \( G(V, E, w) \) be a weighted hypergraph to model the genomic data: each patient sample is denoted by a vertex in \( V \) and each hyperedge denotes one of the two states (1 or 0) of a genomic feature. The incidence matrix \( H \) between \( V \) and \( E \) are determined by the CNV log-ratios or gene expression intensities on the samples. We define a function to assign initial labels to \( V \) in the hypergraph \( G(V, E, w) \). If a vertex \( v \) is in the positive patient group, \( y(v) = 1 \); if it is in the negative patient group, \( y(v) = -1 \); and, if \( v \) is a test sample, \( y(v) = 0 \).

3.1.2 Hypergraph-based learning

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

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

where \( I \) is the identity matrix. Here, \( D_1 \) and \( D_2 \) are used for computing the normalization of the hypergraph Laplacian, and the unnormalized hypergraph Laplacian is \( diag(D_1w) - W \). Empirical results showed that the normalized form gives better classification performance for graph-based learning (Zhou et al., 2006). If the predicted labels on the vertices are consistent with the incidences with the hyperedges, the value of \( \Omega(f, w) \) should be minimized. For criteria (2), we directly calculate the squared-loss between the predicted labeling \( f \) and the original labeling \( y \) as follows,

\[
\sum_{v \in V} (f(v) - y(v))^2 = ||f - y||^2.
\]

3.1.3 Incorporating prior knowledge

To introduce prior knowledge into the hypergraph-based learning, we assume that correlating genomic features should receive similar weights on their associated hyperedges. We define two different functions to encode the prior knowledge. The first function \( \Psi_1(w) \) is a network-Laplacian constraint (Li and Li, 2008). We define an indicator \( b_{e,j} \) to capture the pairwise relation between hyperedges \( e_1 \) and \( e_2 \). The indicator \( b_{e,j} \) is 0 if the two genomic features associated with \( e_1 \) and \( e_2 \) are correlated in the prior knowledge; otherwise 1. Let \( \Delta \) be the correlation matrix with \( \Delta_{j,k} = b_{e,j} \). The second function \( \Psi_2(w) \) is a negative hypergraph Laplacian constraint on weights \( w \) for the hyperedges that are correlated with \( e_1 \) and \( e_2 \). We require the weight of \( e_1 \) to be close to the average weight of \( e_2 \) as follows,

\[
\sum_{j \in E} w_{e,j} = ||w_{e,j}||. \]

3.1.4 Alternating optimization

After the prior knowledge is introduced, the learning task is to minimize the sum of the three cost terms, which is

\[
\Phi(f, w) = \Omega(f, w) + \mu ||f - y||^2 + \rho \Psi(w).
\]
subject to

\[ \omega(e) \geq 0 \quad \text{for } \forall e \in E \]

\[ \sum_{e \in E} \omega(e) c(v(e)) \leq d(v) \quad \text{for } \forall v \in V \]

where \( \mu \) and \( \phi \) are positive real numbers and \( \Psi(w) = \Phi_1(w) + \Psi_0(w) \). The intuition of adding \( \sum_{e \in E} \omega(e) c(v(e)) \) as another set of constraints is to maintain the hypergraph structure. This set of constraints can guarantee each \( d(v) \) is fixed as a constant such that \( \mathbb{D}(f, w) \) is always a linear function of \( w \) when \( f \) is fixed. In the unnormalized form of the hypergraph Laplacian, the constraint \( \sum_{e \in E} \omega(e) c(v(e)) \leq d(v) \) is not required, and a simple lower bound \( \sum_{e \in E} \omega(e) = (c(v) > 0) \) can be used.

The objective function \( \Psi(w) \) in Equation (4) is cubic in \( (f, w) \). However, the formulation contains two sub-problems, both of which are quadratic convex programming problems independently of \( f \) and \( w \). Specifically, if we fix \( f \) to be a specific weight \( w \), satisfying the constraints \( w(e) \geq 0 \) for \( \forall e \in E \) and \( \sum_{e \in E} \omega(e) c(v(e)) \leq d(v) \), the minimization of \( \Psi(w) \) is convex in \( w \). The derivation of the two convex optimization problems is described in Supplementary Sections 1.1 and 1.2. A local optimal solution can be found by solving the two optimizations alternatively by iteration (Beaudel and Hathaway, 2003), under the assumption that \( f \) and \( w \) can be independently optimized. The assumption does not guarantee a global optimal solution. The alternating optimization can be solved by an iterative algorithm proposed by us in Hwang et al. (2008), described in Supplementary Figure 1.

The time complexity of HyperPrior includes solving two minimization problems: fixing \( w \) to learn \( f \) and vice versa. The first problem can be solved by network propagation in \( O(k_1 V^2 E) \), where \( k_1 \) is the round of propagations (Zhou et al., 2006). The value of \( k_1 \) mainly depends on the eigenvalues of the Laplacian matrix. The second problem is a standard convex quadratic programming problem, which can be solved in polynomial time \( O(E^3) \), where \( E \) is a real number. Thus, the time complexity of HyperPrior is \( O(k_1 V^2 E + E^3) \), where \( k_1 \) is the number of iterations of alternating optimization. Usually, \( k_1 \geq 2 \) or 3 in our experiments.

### 3.1.5 Model interpretation

In essence, the regularization framework of HyperPrior is a semi-supervised and relaxed wrapper-feature-selection method, which performs feature selection based on prior knowledge while classifying samples. A dissection of Equation (1) can explain the role of \( W \) in the learning framework. Given a (weighted) hypergraph incidence matrix \( H \), we define its normalized adjacency matrix as \( H = D^{-1/2} HD^{-1/2} \). In the standard hypergraph-based learning framework (Agarwal et al., 2006), a linear kernel \( K_{lin}(V, V) = HWH \) is chosen to construct a similarity graph between objects used for semi-supervised learning in the normalized graph Laplacian \( I - K_{lin}(V, V) \), which is \( (I - D^{-1/2} HD^{-1/2})^{-1} \). Instead of fixing \( W \) to be the identity matrix, the HyperPrior framework treats \( W^{-1/2} \) as a diagonal linear transformation matrix for the original feature space. HyperPrior learns a \( W \) used with linear kernel \( K_{lin}(V, V) = (HWH)^{-1} \). The HyperPrior framework derives an optimal \( W \) to generate the best labeling of the samples based on the prior knowledge given by \( \Psi(w) = \text{diag}(W) \). In general, \( W \) can be any linear transformation matrix. However, to make the learning problem tractable, we restrict \( W \) to be a diagonal matrix with positive weights of the features on the diagonal. If we further restrict the values in \( w \) to be binary \( \{0, 1\} \), \( W \) is a projection matrix and \( K_{lin} \) is a kernel for feature selection. Thus, with a binary \( w \), the regularization framework is a model that performs wrapped feature selection based on prior knowledge for graph-based learning models. However, it is non-deterministic polynomial-time hard (NP-hard) to solve the integer programming problem. We relax \( w \) to be positive real values in optimization. It is still a challenging problem to find an efficient algorithm for feature selection: an \( l_1 \)-norm regularizer \( w \) can be included and the new formulation can still be solved by the same method. However, the additional regularizer will also introduce one more hyper-parameter to tune.

The algorithm proposed by Shapire et al. (2007) was similarly motivated to search for a good projection \( W \) for data matrix \( H \). But instead of learning the \( W \), they directly derived the \( W \) from the eigen-decomposition of the graph Laplacian of the PPI network as a data preprocessing step. The HyperPrior framework attempts to solve both semi-supervised learning and wrapped feature selection together with an objective function on both \( f \) and \( w \). Compared with the last model introduced by Li and Li (2008), the joint learning of \( f \) and \( w \) in our framework creates a harder non-convex problem. However, the semi-supervised learning might give better generalization on the test samples.

#### 3.1.6 Inductive learning

Although HyperPrior is designed for semi-supervised learning, it is convenient to use it for inductive learning as well (Chapelle et al., 2006). Given an optimal weighting \( w^* \) and optimal labeling \( f^* \) learned by HyperPrior, for a new test sample \( t \), we minimize the objective function (4) only with respect to this new label \( f(t) \), that is

\[
\min_{f(t)} \frac{1}{2} \sum_{S \in V} \left( \frac{f(t)}{\sqrt{v(t)}} \right)^2 + \frac{\mu}{2} ||f(t)||^2 + \text{constant}
\]

where \( S_{t,j} = \sum_{v \in S} \frac{\delta_i v(t) v_S(t -) v_S(t +) v \omega(e) c(v(e))}{\sqrt{v(t)}} \). The analytical solution to this optimization problem can be calculated in \( O(n) \) as follows,

\[
\frac{f(t)}{\sqrt{v(t)}} = \frac{\sum_{S \in V} S_{t,j}}{\sum_{S \in V} S_{t,j}}
\]
We first generated a set of vertices with 80% of the vertices in two arrayCGH datasets and two gene expression datasets. A large curated protein–protein interaction network constructed by Chuang et al. (2007) was used as prior knowledge for classifying the gene expression datasets.

In all experiments, we compared the classification performance of HyperPrior with the hypergraph-based learning algorithm (Zhou et al., 2006), support vector machines (SVMs) with linear kernel and RBF kernel [Matlab Bioinformatics Toolbox (V3.0)], \( L_1 \)-SVM and fused SVM (Rapaport et al., 2008) were included as additional baselines in the experiments on the arrayCGH datasets. The linear lasso model by Li and Li (2008) and the graph-Laplacian-transform method by Rapaport et al. (2007) were included as additional baselines in the experiments on the gene expression datasets. The classification performance of all the methods were evaluated by leave-one-out (LOO) accuracy or the area under the curve (AUC) of receiver operating characteristics: the normalized area under a curve plotting the number of true positives against the number of false positives by varying the threshold on the decision values (Gribkov and Robinson, 1996).

4.1 Simulations

To mimic the noisy nature of the genomic data, we tested HyperPrior-LP on artificial hypergraphs with many noisy hyperedges. In all experiments, we labeled 50% of the vertices for training and held out the other 50% of the vertices for testing. We randomly generated hypergraphs with a large number of non-informative hyperedges connecting randomly selected vertices and a certain number of special hyperedges, each of which alone is not very informative but is highly informative in combination.

We first generated a set of vertices with 80% of the vertices in one case and 20% of the vertices in the other case. The set was then split into five weak informative hyperedges with an equal number of vertices. The informative hyperedges were generated to simulate the concerted behavior of genomic features, which are often non-informative unless combined as a module. The prior knowledge was introduced as the pairwise constraints between the informative hyperedges. Some other random constraints between non-informative hyperedges were also introduced as noise.

The algorithms were tested on 100 hypergraphs generated as described above. The average AUC of the baselines and HyperPrior-LP with different percentages of informative hyperedges are reported in Figure 2. Because the results are similar for different choices of \( \rho \) and \( \alpha = \mu / (1 + \mu) \) parameters, we only plot the case with \( (\alpha, \rho) = (0.5, 1) \). It is clear in the plot that, when the prior knowledge gives useful information about interactions between informative hyperedges, the performance of our algorithm is significantly better than SVMs and the hypergraph-based algorithm with uniform weights. Since in this simulation, only very high-order combination of the hyperedges could provide good classification performance, SVMs performed poorly in all cases.

4.2 Classification of arrayCGH data

We tested HyperPrior on two arrayCGH datasets used by Rapaport et al. (2008). The first dataset contains arrayCGH profiles of 57 bladder tumor samples and the second one contains arrayCGH profiles of 78 melanoma tumor samples. Following the data preprocessing procedure in Rapaport et al. (2008), we removed the probes in sexual chromosomes and tested three tumor classification problems: bladder tumors by grade (12 tumors of Grade 1 versus 45 tumors of higher grades) and by stage (16 tumors of Stage T1 versus 32 tumors of Stage T2+), and melanoma tumors by metastases (35 tumors that developed metastases within 24 months versus 43 that did not). A weighted incidence matrix \( W \) was used in the bladder cancer dataset because the results were more stable. We performed a cross-validation by a LOO procedure for the three classification problems. The number of misclassified samples by all the methods are reported in Table 1. On the bladder cancer dataset, HyperPrior-LP and HyperPrior-NB all achieved the same error rate. Overall, HyperPrior-LP and HyperPrior-NB performed better than the baseline methods that do not utilize the spatial prior knowledge, while the two algorithms gave competitive performance against fused SVM, which also utilizes the same spatial prior knowledge.

The weights assigned by HyperPrior-LP with the optimal parameters are plotted separately for amplification events and deletion events along the chromosomes in Supplementary Figure 3. In the bladder cancer dataset, the highly weighted regions of deletion show good agreement with the results reported by Blaveri et al. (2005) in chromosomes 2, 4, 7 and 11, but there is no significant overlap in highly weighted amplification regions. In the melanoma cancer dataset, many of the highly weighted regions of amplification events (chromosomes 7, 8, 17 and 20), and deletion events (chromosomes 4, 5, 8, 11, 14 and 15) show strong consistency with those identified by Onken et al. (2006). It is evident in the plots that only scarce chromosomal regions are highly weighted.
We then evaluated interaction network was used as prior knowledge (Chuang et al., 2002). A large curated human protein–protein interaction network was used as prior knowledge (Chuang et al., 2007). This network contains 57 235 interactions integrated from yeast two-hybrid experiments, predicted interactions from orthology and co-citation, and other literature reviews. The details of the quantization and normalization of the datasets are described in the original papers. The classification task is to classify patients who developed metastasis or were free of metastasis in 5 years after prognosis. As suggested by van’t Veer et al. (2002), 231 genes were selected on a training set of 78 patients and the remaining 19 patients were held out as the test set in the van’t Veer et al. dataset. A LOO cross-validation was then applied to the 78-patients training dataset to select parameters for classifying the 19-patients test set. The detailed results of cross-validation are given in Supplementary Tables 1-4. In the experiments on the van de Vijver et al. dataset, we used for classification two subsets of hypothetical cancer susceptibility genes: 326 genes from Ingenuity and 1464 genes from Sloan Kettering cancer gene list (http://cbio.mskcc.org/CancerGenes/). We randomly run 5-fold cross-validation multiple times on the van de Vijver et al. dataset and measured the average AUC. Note that within each experiment of a 5-fold cross-validation, another 4-fold cross-validation was applied on the training set to determine the best parameters for HyperPrior and the baseline algorithms to test the held-out set. The classification results in Table 2 show that both HyperPrior-LP and HyperPrior-NB performed better than SVMs and the method by Rapaport et al. (2007) in all the experiments. On both datasets, HyperPrior achieved ∼2% improvement on the average AUCs. Although this improvement seems marginal, pairwise comparisons show that HyperPrior outperformed SVMs and the method by Rapaport et al. (2007) significantly with t-test (Supplementary Tables 5 and 6). The method by Li and Li (2008) performed similarly as HyperPrior (0.695 versus 0.697) in the experiments with 326 genes on the van de Vijver et al. dataset, but this method did not perform well in the other two experiments. The hypergraph-based algorithm achieved slightly worse results compared with HyperPrior in the experiments with 1464 genes on the van de Vijver et al. dataset, but in the other experiments, the results were statistically worse. To demonstrate that HyperPrior is capable of identifying true cancer susceptibility genes, we compared the highly weighted genes by HyperPrior on the van de Vijver et al. dataset with known breast cancer causative genes reported in the overview section of breast cancer (MIM 114480) in Online Mendelian Inheritance in Man (May 2007; http://www.ncbi.nlm.nih.gov/omim/). While correlation coefficients gave very low rankings to the 16 known breast cancer causative genes in the dataset, HyperPrior-LP in two different settings (ρ = 1 and 0.1) assigned high ranks to most of the genes, with 14 out of 16 genes ranked in the top 300 genes (Supplementary Table 7). Notable examples of the biomarker genes are tumor protein p53 (TP53), estrogen receptor 1 (ESR1), v-Ha-ras Harvey rat sarcoma viral oncogene homolog (HRAS) and v-Ki-ras2 Kirsten rat sarcoma viral oncogene homolog (KRAS).

### Table 1. Classification performance on arrayCGH data

<table>
<thead>
<tr>
<th>LOO errors</th>
<th>SVM (linear)</th>
<th>SVM (RBF)</th>
<th>L1-SVM</th>
<th>Fused SVM</th>
<th>Hypergraph</th>
<th>HyperPrior-LP</th>
<th>HyperPrior-NB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bladder tumors (by grade)</td>
<td>9</td>
<td>9</td>
<td>12</td>
<td>7</td>
<td>11</td>
<td>6</td>
<td>6</td>
</tr>
<tr>
<td>Bladder tumors (by stage)</td>
<td>9</td>
<td>9</td>
<td>13</td>
<td>7</td>
<td>9</td>
<td>5</td>
<td>6</td>
</tr>
<tr>
<td>Melanoma tumors</td>
<td>10</td>
<td>10</td>
<td>8</td>
<td>7</td>
<td>7</td>
<td>7</td>
<td>7</td>
</tr>
</tbody>
</table>

This table shows the number of misclassified samples in the LOO cross-validation on the bladder cancer dataset with two different labeling schemes (by tumor grade or by cancer stage) and the melanoma cancer dataset.

### 4.3 Classification of gene expressions

We then evaluated HyperPrior on two breast cancer gene expression datasets, the van’t Veer et al. (2002) dataset with 97 samples and the van de Vijver et al. dataset with 295 samples (van de Vijver et al., 2002). A large curated human protein–protein interaction network was used as prior knowledge (Chuang et al., 2007). This network contains 57 235 interactions integrated from yeast two-hybrid experiments, predicted interactions from orthology and co-citation, and other literature reviews. The details of the quantization and normalization of the datasets are described in the original papers. The classification task is to classify patients who developed metastasis or were free of metastasis in 5 years after prognosis. As suggested by van’t Veer et al. (2002), 231 genes were selected on a training set of 78 patients and the remaining 19 patients were held out as the test set in the van’t Veer et al. dataset. A LOO cross-validation was then applied to the 78-patients training dataset to select parameters for classifying the 19-patients test set. The detailed results of cross-validation are given in Supplementary Tables 1-4. In the experiments on the van de Vijver et al. dataset, we used for classification two subsets of hypothetical cancer susceptibility genes: 326 genes from Ingenuity and 1464 genes from Sloan Kettering cancer gene list (http://cbio.mskcc.org/CancerGenes/). We randomly run 5-fold cross-validation multiple times on the van de Vijver et al. dataset and measured the average AUC. Note that within each experiment of a 5-fold cross-validation, another 4-fold cross-validation was applied on the training set to determine the best parameters for HyperPrior and the baseline algorithms to test the held-out set. The classification results in Table 2 show that both HyperPrior-LP and HyperPrior-NB performed better than SVMs and the method by Rapaport et al. (2007) in all the experiments. On both datasets, HyperPrior achieved ∼2% improvement on the average AUCs. Although this improvement seems marginal, pairwise comparisons show that HyperPrior outperformed SVMs and the method by Rapaport et al. (2007) significantly with t-test (Supplementary Tables 5 and 6). The method by Li and Li (2008) performed similarly as HyperPrior (0.695 versus 0.697) in the experiments with 326 genes on the van de Vijver et al. dataset, but this method did not perform well in the other two experiments. The hypergraph-based algorithm achieved slightly worse results compared with HyperPrior in the experiments with 1464 genes on the van de Vijver et al. dataset, but in the other experiments, the results were statistically worse. To demonstrate that HyperPrior is capable of identifying true cancer susceptibility genes, we compared the highly weighted genes by HyperPrior on the van de Vijver et al. dataset with known breast cancer causative genes reported in the overview section of breast cancer (MIM 114480) in Online Mendelian Inheritance in Man (May 2007; http://www.ncbi.nlm.nih.gov/omim/). While correlation coefficients gave very low rankings to the 16 known breast cancer causative genes in the dataset, HyperPrior-LP in two different settings (ρ = 1 and 0.1) assigned high ranks to most of the genes, with 14 out of 16 genes ranked in the top 300 genes (Supplementary Table 7). Notable examples of the biomarker genes are tumor protein p53 (TP53), estrogen receptor 1 (ESR1), v-Ha-ras Harvey rat sarcoma viral oncogene homolog (HRAS) and v-Ki-ras2 Kirsten rat sarcoma viral oncogene homolog (KRAS).
5 DISCUSSION

We introduce a hypergraph-based semi-supervised learning framework for sample classification and biomarker selection in arrayCGH and gene expression data with prior knowledge. We evaluated the algorithms with rigorous cross-validation and thorough parameterizations to show that the algorithms achieved similar results. This observation suggests that both cost functions are viable choices to incorporate prior knowledge. We also used a weighted distance between the spots on chromosomes as correlation in the spatial prior graph for the arrayCGH datasets. But our preliminary results showed no improvement with the introduction of the more complex modeling. In an additional experiment, we also tested whether the high ranking of the known breast cancer genes by HyperPrior was a biased output caused by the high connectivity of the known cancer genes in the PPI. We introduced random edges into the PPI network to make each gene have a degree that is at least half of the maximum degree in the network. We obtained similar top genes with the randomized network (Supplementary Table 8). This result indicates that HyperPrior indeed can utilize clusters in the PPI as useful prior knowledge for biomarker selection.

Conflict of Interest: none declared.

REFERENCES


