Structure Feature Selection for Chemical Compound Classification

Hongliang Fei and Jun Huan

Abstract— **With the development of highly efficient cheminformatics data collection technology, classification of chemical structure data emerges as an important topic in cheminformatics. Towards building highly accurate predictive models for chemical data, here we present an efficient feature selection method. In our method, we first represent a chemical structure by its 2D connectivity map. We then use frequent subgraph mining to identify structural fragments as features for graph classification. Different from existing methods, we consider the spatial distribution of the subgraph features in the graph data and select those ones that have consistent spatial locations.**

We have applied our feature selection methods to several cheminformatics benchmarks. Our experimental results demonstrate a significant improvement of prediction as compared to the state-of-the-art feature selection methods.

I. INTRODUCTION

A new challenge for bioinformatics is to develop computational techniques to elucidate the roles of small organic molecules in biological systems. Such successful delineation will lead to better drugs, improved chemical tools to study biological systems, and more effective environmental preservation strategy. Traditionally the computational analysis of chemicals was done mainly within pharmaceutical companies for therapeutics discovery [6]. This situation, however, has been changed dramatically in the last few years. With the Chemical Genetics Initiative and the Molecular Library Initiative (started by NIH in 2002, [20], and 2004, [1], respectively), digitalized data about chemical structures and their biological activities (e.g. interactions with biological systems) grow exponentially fast.

A major challenges in building structure-activity relationship models for chemicals lies in the large number of structural features of the molecule structures. The objective of this paper is to derive an automated way to construct a low-dimensional vector representation for graph represented chemical structures through developing a highly effective feature selection methods. Finding a proper vector representation of graphs and graph represented chemicals may lead to more accurate models, reduced computational time, and better explanations of the real relationship between biomolecules and their functions, and hence worth a careful investigation.

Current solutions for feature selection problems can be roughly divided into two categories: feature extraction and feature selection [2]. Principle Component Analysis (PCA) projects data to a eigenvector space to reduce the dimensionality and hence to obtain a small number of features [12], [16]. Similar methods include Linear Discriminative Analysis LDA [23], Local Linear Embedding LLE [17] and Isomap [19]. Using Kernel PCA, investigators have designed algorithms to embed a graph to a vector space and achieved good empirical results in classification [21], [18].

Current feature selection methods include *feature filtering* methods that select individual features whose distribution correlates the distribution of the class labels. Such methods include term frequency thresholding, mutual information, information gain, χ^2, and Pearson Correlation as studied in [22]. In contrast to filtering method, which do not consider the dependency between features and may select redundant features, *wrapper methods* search through the feature subset space and identify highly informative features by using a classifier to score the subsets of features [14], [15].

Adapting existing feature extraction and feature selection methods to cheminformatics is non trivial. First, chemical structures are discrete structures. There is no obvious choices of features in chemical structures to start the feature selection method. Second, kernel functions map chemical structures to a Hilbert space implicitly and thus avoid the problem of direct feature extraction. Though theoretic appealing, limited progresses have been made in reality in applying graph kernel functions to extract useful features in cheminformatics. This is due to several reasons: (i) design a kernel function for molecular structures is not easy, (ii) the connection of kernel space and the original structure space is not clear and (iii) it is hard to explain the physical meaning of the identified features using kernel PCA techniques.

In this paper we have developed a novel strategy for feature selection in chemical structures. In our method, we first represent a chemical structure by its 2D connectivity map where nodes represent atoms and edges represent chemical bonds in a chemical. We then use frequent subgraph mining to identify structural fragment features. Our main thrust of the paper is that rather than using a simple postprocessing technique to select features, we consider the spatial distribution of the features and use that information to guild our feature selection process. We illustrate our intuition with the following example.

On the top portion of Figure 1, we show the structure of a chemical structure and its graph representation. At the bottom of the same figure we show a graph database of three graph represented chemicals and three subgraph patterns. We see that subgraph features F_1 and F_2 occur in every graph with a consistent relative spatial distribution. In contrast to F_1 and F_2, subgraph feature F_3 has quite different spatial distribution as compared to F_1 and F_2. In regular feature selection, F_1, F_2, and F_3 occur in the same set of graphs and

Hongliang Fei and Jun Huan are in Department of Electrical Engineering and Computer Science, University of Kansas, Lawrence, KS 66047 {hfei, jhuan}@ittc.ku.edu

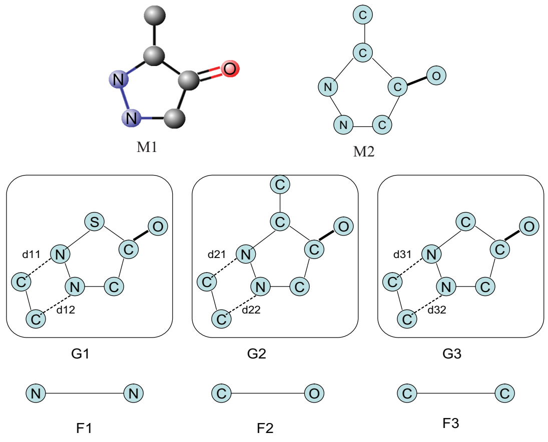

Fig. 1. The top: a chemical structure and the right is its corresponding graph representation; The middle and the lower: spatial distribution of three frequent fragments in a chemical structure graph database

hence may have the same classification power. Clearly this is not the case in this example. Based on intuition, we have designed an integrated approach of two existing approaches: graph kernels and subgraph mining by designing a feature selection method working on graph spectrum kernels to gain deeper understanding of graph data. Our comprehensive experimental study of the designed algorithms using real-world data sets revealed the power of the novel feature selection method.

We formally define our feature selection problem using graphs and graph features below:

A. Problem Statement

Given a set of graph represented chemicals $\mathcal{G}$, each graph in $\mathcal{G}$ has an associated labels c, and a set of subgraphs $\mathcal{F}$ extracted from $\mathcal{G}$, the **graph feature selection problem** is to select a subset of features $\mathcal{F}_s \subset \mathcal{F}$, to give a better classification accuracy for the graph data set.

B. Related Work

Extracting features, in the form of subgraphs, from graph data has been well studied in graph database mining methods. In this methods, the goal is to extract highly informative subgraphs from a set of graphs. Typically some filtering criteria are applied, among those the most widely used is the frequency of a subgraph. For example, Huan *et al.* develop a depth-first search algorithm: Fast Frequent Subgraph Mining (FFSM) [10]. This algorithm identified all connected subgraphs that occurs in a large fraction of graphs in a graph data set. Majority of the frequent subgraph feature extraction methods are unsupervised, meaning that there is no class labels information available (or such information are deliberately ignored) with a few exceptions. For example, an odd ratio is used to select subgraphs that is highly informative to build classifier in [9].

On the other hand, many existing feature selection methods are supervised, determining the relevance of a feature through computing the correlation of feature value distribution and class label distribution. Traditional feature filtering methods select features independent of any classifier. In contrast to filtering method, which do not consider the dependency between features and may select redundant features, wrapper methods search through the subset space and identify highly informative features by using a classifier to evaluate the classification power of subsets of features and identify optimal subsets [14].

Kernel methods are now widely used in supervised learning and feature selection as well. For example, in the method of Support Vector Machine Recursive Feature Elimination (SVM_RFE)[8], SVM_RFE selects features via a greedy backward feature elimination. SVM_RFE first build a linear classifier, it then uses the weight vector of the hyperplane constructed by the training samples to rank features. During each iteration, lower ranked features were removed and new hyperplane is constructed and so on so forth. The limitation of SVM_RFE is that it works only with linear kernel.

Spectral feature selection [24], as a filtering method, explored an uniformed frame for feature selection in both unsupervised and supervised learning. It first constructed an object graph, where each node is corresponding to an object of training data; then ranked features using graph spectral decomposition and selected a subset of features based on their rank. Since spectral feature selection is a filtering methods, the feature dependency information is ignored. Cao *et al.* recently developed a method for feature selection in the kernel space rather than the original feature space based on Maximum Margin concept. Without tracing back into original feature space, they could select features in Kernel space.

Maximum Margin Feature selection (MMRFS) [4] is a wrapper method. In this method MMRFS uses information gain to weigh the correlation between each feature and class labels. MMRFS then selects a feature with less redundancy and covering new training samples.

Though feature selection have been developed for a long time, none of the existing method considers the special characteristics of graph data and hence may not provide the optimal results for graph feature selection. The objective of this paper is to develop a highly effective graph feature selection methods.

II. METHODOLOGY

Our structure based feature selection method has two steps: (1) feature extraction and (2) feature selection. In the feature extraction step, we mine frequent subgraphs in the training samples as features. We then apply a feature selection method, as outlined below and discussed in details in [7], to select a smaller set of features to build graph classification model.

A. Notation

In this paper, we use capital letters, such as G, for a single graph and upper case calligraphic letters, such as $\mathcal{G}$ = $G_1, G_2, \ldots, G_n$, for a set of n graphs. We assume each

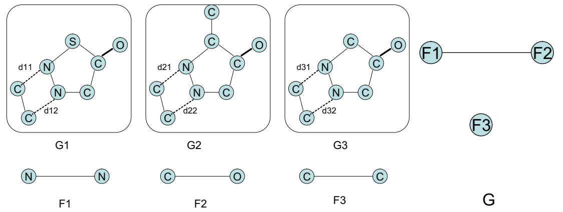

Fig. 2. Left: Three subgraphs and three embeddings of the subgraphs. This figure is duplicated from Figure 1 for clarity. Right: The feature consistency map for the three subgraphs shown in Figure 1

graph $G_i \in \mathcal{G}$ has an associated class label c_i from a label set C. We use $\mathcal{F} = F_1, F_2, \ldots, F_n$ for a set of n features. Given a set of n features $\mathcal{F}$ and a graph G, we create a feature vector for G, denoted by $G^{\mathcal{F}}$, indexed by features in $\mathcal{F}$ and with values indicate whether the related feature is present (1) or absent (0) in the graph G. In other words, $G^{\mathcal{F}} = (G^{F_i})_{i=1}^{n}$ and

$$G^F = \begin{cases} 1 & \text{if } F \subseteq G \\ 0 & \text{otherwise} \end{cases}$$

B. Feature Consistency Map

Here we present a way to measure the consistency in a set of features without the class label information. Such measurement is clearly unsupervised, meaning that we do not take class label distribution into consideration. Our main source of information is the spatial distribution of subgraph features. To quantify the information, we propose a data structure called feature consistency map, whose nodes are features and whose edges indicating the consistent relationship of features. Formal definition of the feature consistency map is presented below:

Definition 2.1: (**Feature Consistency Map**) A **feature consistency map** is a graph $G = (\mathcal{V}, \mathcal{E})$ where the vertex set $\mathcal{V}$ and arc set $\mathcal{E}$ are specified below:

- Each vertex in $\mathcal{V}$ represents a feature
- Two vertices are connected by an arc in $\mathcal{E}$ if the two vertices are "consistent"

To avoid confusion of the feature consistency map and a regular graph that we specified before, we use vertex and arc to denote commonly used term "node" and "edge". The key point in the definition is how we define the consistency of two features. In Figure 1, intuitively we see that F_1 and F_2 are consistent but F_1 and F_3 are not consistent. To make the intuition more clearly, we introduce the following definitions:

Definition 2.2: **Embedding Distance**: Given two frequent subgraphs features F_i and F_j and their embeddings of e_i e_j in a graph G, the embedding distance, denoted by $d^G(e_i, e_j)$ is defined as:

$$d^G(F_i, F_j) = \frac{\sum_{u \in e_i} \sum_{v \in e_j} d(u, v)}{|e_i| \times |e_j|} \quad (1)$$

Where $d(u, v)$ the shortest distance between node u and node v.

In Figure 2, subgraph feature F_1 has an embedding $\{N, N\}$ in G_1. F_2 has an embedding $\{C, O\}$ in G_1. Hence, the embedding distance between pattern F_1 and F_2, denoted by $d^{G_1}(F_1, F_2)$ is $\frac{2+3+3+2}{2*2}$=2.5.

Based on the embedding distances, we define two features are *consistent*, if the standard variance of the set of embedding distances is less than some threshold max_var. We refer interested readers for further details of the "consistency" relationship, including the handling of multiple occurrences of embeddings to [7].

Below we show how we use feature consistency map to perform feature selection.

C. Kernel-Target Alignment Framework

In this work we consider selecting features whose distribution is consistent with the distribution of the class labels. Towards that end, we compute the *object kernel matrix* ξ as defined below.

$$\xi = ((c_i == c_j))_{i,j=1}^{n} \quad (2)$$

where $(X == Y) = 1$ if $X = Y$ and otherwise 0. c_i is the class label of the ith object.

Given a set of features $\mathcal{F}$, we define a *feature kernel matrix* $S_{\mathcal{F}}$ as

$$S_{\mathcal{F}} = (K(G_i^{\mathcal{F}}, G_j^{\mathcal{F}}))_{i,j=1}^{n} \quad (3)$$

In the formula, K can be any kernel function. For simplicity in our experimental study we use linear kernel $K(X, Y) = X \times Y$.

With the feature kernel function and object kernel function, we use the following formula to measure whether the feature kernel is "consistent" with the object kernel. Toward that end, we introduce a binary function $\cdot : M \times M \rightarrow \mathbb{R}$ to compute the inner product of two matrices as

$$X \cdot Y = trace(X^T \times Y) \quad (4)$$

where M is the set of all n by m matrices.

Based on the function $\cdot$, we define the similarity between two matrices is the inner product of the two matrices X and Y, normalized by the norm of the X and Y, or

$$\mathfrak{S}(X, Y) = X \cdot Y / (||X|| \times ||Y||). \quad (5)$$

where $||X|| = \sqrt[2]{X \cdot X}$.

Geometrically the similarity function $\mathcal{S}$ measures the cosine value of the angel between two kernel matrices. This measurement is first used in [5] to automatically select kernel functions. We adapt it here for the purpose of feature selection. Before we proceed to our feature selection method, we present an important data structure, which we call feature consistency map.

D. Feature Ranking and Forward Structure Based Feature Selection

Once we compute the Feature Consistency map G, we use a simple way to rank the features by sorting the features according to their degree (number of edges incident on the feature) in G in descending order. We sort features according to their node degree in the feature consistency map and select to top k features. We call this method SFS_Filtering.

In forward structure based feature selection, we sort the features using the same procedure from the feature filtering method. Different from the feature filtering method, we use the Equation 5 and select feature in the context of selected features. Specifically, we evaluate the similarity between the resulting kernel function (may contain several features) and the object kernel function and make sure when we select a feature, the similarity value monotonically increases.

The following is the algorithm for forward selection, where $\mathfrak{S}$ measures the similarity between two matrices, ξ is the object kernel matrix, and $\mathcal{F}$ is a set of subgraph features as we discussed before.

Algorithm 1 SFS_FS($\mathcal{F}, \xi$)

```
$\mathcal{F}_s = \emptyset, T_0 = 0$
sort $\mathcal{F}$ according to the node degree in the related feature consistency map
$n \leftarrow |\mathcal{F}|$
for $i = 1, \ldots, n$ do
    $T \leftarrow \mathfrak{S}(\mathcal{F}_s \cup F_i, \xi)$
    if $T > T_0$ then
        $\mathcal{F}_s \leftarrow \mathcal{F}_s \cup F_i$
        $T_0 \leftarrow T$
    end if
end for
return $\mathcal{F}_s$
```

Finally, we notice that we may augment a weight to each node in the feature consistency map. A straightforward way to assign a weight to a node is to compute the Pearson's Correlation Coefficient between the feature and the class labels in the training data set. Many other choices are available, such as mutual information [22] and odd ratio [11]. It is hard to enumerate all possible choices and we use Pearson Correlation simple empirically it gives us good results. Without special explaining, the feature consistency map that we use in this paper is always node-weighted where the weight of the node is computed using Pearson's Correlation Coefficients.

III. EXPERIMENT

In this section, a comprehensive study of the performance of our feature selection method is performed using 5 real-world chemical structure graph data sets. We compared our method with 3 state-of-the-art feature selection methods: SVM Recursive Feature Elimination (SVM_RFE) [8], Spectral Feature Selection [24], and Maximum Margin Feature selection (MMRFS) [4].

For each data set, we apply the FFSM algorithm [10] to extracting frequent subgraph patterns from the data sets and use the LibSVM package [3] to train a Support Vector Machine (SVM) for classification. Performance is measured via classification accuracy.

A. Data Sets

In the paper, We use data sets from drug virtual screening experiments [13]. In a data set, the target values are drugs' binding affinity to a particular protein. For each protein, the data provider selected 50 chemical structures that clearly bind to the protein ("active" ones). The data provider also listed chemical structures that are very similar to the active ones (judged with domain knowledge) but clearly do not bind to the target protein("negative" ones). This list is known as the "decoy" list. We randomly sample 50 chemical structures from the decoy list. Refer to [13] for further details regarding the nature of the data set. To reiterate, each of these 5 data sets contains 100 compounds with 50 positives and 50 negatives.

After removing Hydrogen atoms in our graph representation of chemicals, as commonly done in the cheminformatics field, we follow the same procedure [10] to use a graph to model a chemical structure: a node represents an atom and an edge represents a chemical bond. The characteristics of data sets is shown in Table II.

TABLE II
DATA SET: THE SYMBOL OF THE DATA SET. S: TOTAL NUMBER OF SAMPLES IN THE DATA SET. P: TOTAL NUMBER OF POSITIVE SAMPLES, N: TOTAL NUMBER OF NEGATIVE SAMPLES, $\overline{V}$: AVERAGE NUMBER OF NODES IN THE DATA SET, $\overline{E}$: AVERAGE NUMBER OF EDGES IN THE DATA SET

Data set	S	P	N	$\overline{V}$	$\overline{E}$
A1A	100	50	50	26	28
CDK2	100	50	50	22	25
COX2	100	50	50	22	24
FXa	100	50	50	27	29
PDE5	100	50	50	26	28

B. Experimental Setup

For each data set, we first represent chemical structures by its 2D connectivity map. Using the FFSM algorithm [10] with $min\_support = 50\%$ and with at least 5 nodes and no more than 10 nodes, we mine frequent subgraphs. We treat each subgraph as a feature and create a binary feature vector for each graph in the data set, indexed by the mined subgraphs, with values indicate the existence (1) or absence (0) of the related subgraphs. All feature selection methods that we compared with are based on the same feature sets.

We implement our own version of the spectral feature selection method, which is a filtering method with no additional parameters. SVM_RFE executable is obtained as the one included in the spider machine learning toolbox http://www.kyb.tuebingen.mpg.de/bs/people/spider/. H. Cheng kindly provided us with MMRFS executable, and we use the default parameter (coverage threshold δ=1 and $min\_sup = 0.5$). To compute the feature consistency map that is used in the Pattern_SFS method, we set max_var to be 0.5.

To have a fair comparison, we select a fixed number of k features using each method and compare the classification accuracy of the selected features. In our experiments, we set $k = 25$. Empirical study shows that there is no significant classification accuracy change if we replace the fixed value 25 with a relatively wide range of values.

TABLE I

AVERAGE AND STANDARD DEVIATION FOR PRECISION AND RECALL OF FOUR METHODS ON 5 DATA SETS. STARS (*) DENOTE THE HIGHEST PRECISION OR CALL AMONG ALL COMPETING METHODS FOR A DATASET.

	Pattern_SFS		MMRFS		Spectral_FS		SVM_RFE	
Data set	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall
A1A	0.890±0.012*	0.939±0.033	0.652±0.029	0.842±0.093	0.879±0.023	0.951±0.039*	0.857±0.027	0.882±0.040
CDK2	0.843±0.065	0.707±0.080	0.945±0.057*	0.613±0.076	0.875±0.040	0.724±0.072	0.819±0.020	0.809±0.040*
COX2	0.884±0.034	0.700±0.030	0.825±0.065	0.662±0.066	0.932±0.039*	0.670±0.015	0.812±0.024	0.804±0.033*
FXa	0.932±0.029*	0.904±0.054	0.667±0.020	0.958±0.081*	0.899±0.055	0.845±0.051	0.838±0.037	0.823±0.027
PDE5	0.901±0.042*	0.859±0.023*	0.702±0.030	0.675±0.070	0.881±0.044	0.823±0.063	0.867±0.014	0.847±0.012

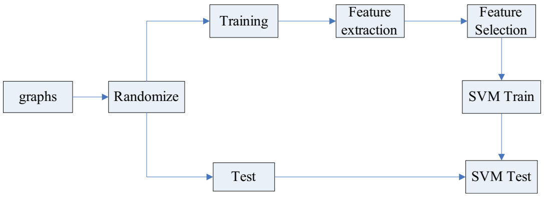

Fig. 3. Experimental workflow for a single cross-validation trial.

After feature selection, we use SVM with RBF kernel and default parameters (C=1, γ=0.5) to obtain accuracy in all the experiments. We perform standard 5-fold cross validation to derive training and testing samples. For each cross validation, we compute precision as (TP/(TP+FP)), recall as (TP/(TP+FN)), and accuracy as (TP+TN/S) where TP stands for true positive, TN stands for true negative and S stands for the total number of samples. For each data set, we repeat the 5-fold cross validation 10 times and report the average precision, recall, and accuracy. Figure 3 gives an overview of our experimental setup.

C. Experimental Results

1) Performance: In this section, we show the performance of our method compared with three additional methods mentioned before on five real-world datasets. The accuracy is shown in Figure 4 and the precision and recall are shown in Table I.

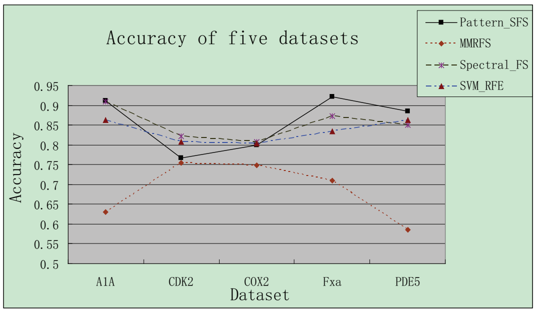

Fig. 4. Comparing the classification accuracy of 4 feature selection methods on 5 data sets.

From the figure, it is clear that Pattern_SFS outperforms MMRFS in all the 5 data sets, outperforms the SVM_RFE method and the spectral feature selection method in 3 out of the 5 data sets. Overall, Pattern_SFS is the best method in 3 out of the 5 tested data sets. The results confirm our hypothesis that considering the spatial distribution of subgraph features results in better selection of discriminative features.

2) Comparison of Variations of Structure Based Feature Selection: Here we compare seven variations of the basic Structure Based Feature Selection methods, including:

- forward feature selection (SFS_FS),
- backward feature elimination (SFS_BE),
- filtering (SFS_Filter),
- forward feature selection with un-weighted feature consistency map (SFS_FS_NPC),
- filtering with un-weighted feature consistency map (SFS_Filter_NPC).

Besides, we present results without any feature selection (Pattern_all) and results with features selected by Pearson Correlation Coefficient Selection (PCCS). The results are shown in Table III. From the table, we observe that Pattern_All often achieves the worst result, which demonstrates that redundant features will result in over-fitting problem and diminish the classification accuracy. Although PCCS takes correlation between a single feature and label, it neglects dependence of features and hence usually do not select the optimal feature subsets. The performance of PC weighted forward selection is the best overall. In the paper, we use PC weighted forward selection to compare with other three state-of-the art methods.

Overall, our structure based feature selection method is effective and achieves good accuracy. Since feature selection can be viewed as a data preprocessing step, any other feature selection method can be combined with our framework and our method is applicable to any current start of art classifiers.

IV. CONCLUSIONS

In this paper, we presented a novel feature selection method for chemical classification. By using structural fragment as features and ranking features based on their spatial distribution and their contributions to classification, we have designed a feature selection method (and several variations) called structure based feature selection method. Compared with current state-of-the-art methods as evaluated on 5 real world data sets, our method outperforms the 3 state-of-the-art methods on majority of the tested data sets. In the future, we plan to extend structure feature selection to kernel space.

TABLE III

CLASSIFICATION ACCURACY OF DIFFERENT IMPLEMENTATIONS OF THE STRUCTURE BASED FEATURE SELECTION METHODS.

Data set	Pattern_All	SFS_Filte	SFS_FS	SFS_BE	SFS_Filter_NPC	SFS_FS_NPC	PCCS
A1A	0.626	0.809	0.912*	0.488	0.812	0.821	0.898
CDK2	0.668	0.679	0.766	0.535	0.666	0.776	0.850*
COX2	0.756	0.817*	0.802	0.475	0.783	0.793	0.788
FXa	0.714	0.892	0.921*	0.576	0.891	0.918	0.865
PDE5	0.602	0.904*	0.885	0.574	0.902	0.888	0.849

V. ACKNOWLEDGMENTS

This work has been supported by the Kansas IDeA Network for Biomedical Research Excellence (NIH/NCRR award #P20 RR016475) and a NIH grant #R01 GM868665. HF is partially supported by the Office of Naval Research through Award Number N00014-07-1-1042, Oak Ridge National Laboratory (ORNL) via Award Number 4000043403, and the KU Transportation Research Institute (KUTRI).

REFERENCES

[1] C. Austin, L. Brady, T. Insel, and F. Collins. Nih molecular libraries initiative. *Science*, 306(5699):1138–9, 2004.

[2] B. Cao, D. Shen, J.-T. Sun, Q. Yang, and Z. Chen. Feature selection in a kernel space. In *ICML '07: Proceedings of the 24th international conference on Machine learning*, pages 121–128, New York, NY, USA, 2007. ACM.

[3] C. Chang and C. Lin. Libsvm: a library for support vector machines, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[4] H. Cheng, X. Yan, J. Han, and C.-W. Hsu. Discriminative frequent pattern analysis for effective classification. In *Proceedings of the 23rd IEEE International Conference on Data Engineering (ICDE)*, 2007.

[5] N. Cristianini, J. Shawe-Taylor, and A. Elisseeff. On kernel-target alignment, 2001.

[6] C. Dobson. Chemical space and biology. *Nature.*, 432(7019):824–8, 2004.

[7] H. Fei and J. Huan. Structure feature selection for graph classification. In *Proceedings of the ACM 17th Conference on Information and Knowledge Management (CIKM)*, 2008.

[8] I. Guyon, J. Weston, S. Barnhill, and V. Vapnik. Gene selection for cancer classification using support vector machines. *Machine Learning*, 46:389–422, 2002 January.

[9] J. Huan, W. Wang, D. Bandyopadhyay, J. Snoeyink, J. Prins, and A. Tropsha. Mining protein family specific residue packing patterns ¿from protein structure graphs. In *Proceedings of the 8th Annual International Conference on Research in Computational Molecular Biology (RECOMB)*, pages 308–315, 2004.

[10] J. Huan, W. Wang, and J. Prins. Efficient mining of frequent subgraph in the presence of isomorphism. In *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM)*, pages 549–552, 2003.

[11] J. Huan, W. Wang, A. Washington, J. Prins, R. Shah, and A. Tropsha. Accurate classification of protein structural families based on coherent subgraph analysis. In *Proceedings of the Pacific Symposium on Biocomputing (PSB)*, pages 411–422, 2004.

[12] I. Jolliffe. *Principal Component Analysis.* Springer; 2nd ed. edition, 1986.

[13] R. Jorissen and M. Gilson. Virtual screening of molecular databases using a support vector machine. *J. Chem. Inf. Model.*, 45(3):549–561, 2005.

[14] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[15] F. Li, Y. Yang, and E. P. Xing. From lasso regression to feature vector. In *Advances in Neural Information Processing Systems*, 2005.

[16] A. M. Martinez and A. C. Kak. PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):228–233, 2001.

[17] S. T. Roweis and L. K. Saul. Nonlinear Dimensionality Reduction by Locally Linear Embedding. *Science*, 290(5500):2323 – 2326, 2000.

[18] B. Schölkopf and A. J. Smola. *Learning with Kernels.* the MIT Press, 2002.

[19] J. B. Tenenbaum, V. de Silva, and J. C. Langford. A Global Geometric Framework for Nonlinear Dimensionality Reduction. *Science*, 290(5500):2319 – 2323, 2000.

[20] N. Tolliday, P. A. Clemons, P. Ferraiolo, A. N. Koehler, T. A. Lewis, X. Li, S. L. Schreiber, D. S. Gerhard, and S. Eliasof. Small molecules, big players: the national cancer institute's initiative for chemical genetics. *Cancer Research*, 66:8935–42, 2006.

[21] S. Yan, D. Xu, B. Zhang, and H.-J. Zhang. Graph embedding: A general framework for dimensionality reduction. In *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2*, pages 830–837, Washington, DC, USA, 2005. IEEE Computer Society.

[22] Y. Yang and J. O. Pedersen. A comparative study on feature selection in text categorization. In D. H. Fisher, editor, *Proceedings of ICML-97, 14th International Conference on Machine Learning*, pages 412–420, Nashville, US, 1997. Morgan Kaufmann Publishers, San Francisco, US.

[23] H. Yu and J. Yang. A direct LDA algorithm for high-dimensional data with application to face recognition. *Pattern Recognition*, 34(10):2067–2070, 2001.

[24] Z. Zhao and H. Liu. Spectral feature selection for supervised and unsupervised learning. In *Proceedings of the 24th International Conference on Machine Learning (ICML)*, 2007.