A Hybrid SOM-SVM Approach for the Zebrafish Gene Expression Analysis

Wei Wu1,2, Xin Liu1, Min Xu1, Jin-Rong Peng1, and Rudy Setiono2*
1Laboratory of Functional Genomics, Institute of Molecular and Cell Biology, Singapore 138673; 2School of Computing, National University of Singapore, Singapore 117543.

Microarray technology can be employed to quantitatively measure the expression of thousands of genes in a single experiment. It has become one of the main tools for global gene expression analysis in molecular biology research in recent years. The large amount of expression data generated by this technology makes the study of certain complex biological problems possible, and machine learning methods are expected to play a crucial role in the analysis process. In this paper, we present our results from integrating the self-organizing map (SOM) and the support vector machine (SVM) for the analysis of the various functions of zebrafish genes based on their expression. The most distinctive characteristic of our zebrafish gene expression is that the number of samples of different classes is imbalanced. We discuss how SOM can be used as a data-filtering tool to improve the classification performance of the SVM on this data set.

Key words: self-organizing map, support vector machine, clustering, classification

Introduction

Biology used to be data-poor science. With more advanced techniques developed in recent years, biologists are now able to transform vast amounts of biological information into useful data. This makes it possible for scientists to study gene functions globally, and a new field, functional genomics, emerges. Specifically, functional genomics refers to the development and application of global (genome-wide or system-wide) experimental approaches to assess gene functions by making use of the information and reagents provided by structural genomics. It is characterized by high-throughput or large-scale experimental methodologies followed by statistical and computational analyses.

Microarray technology can be employed to monitor large amounts of genes' expression level in parallel. Here gene expression refers to the process to transcribe a gene's DNA sequence into RNA that serves as a template for protein production, and gene expression level indicates how active a gene is in a certain tissue, at a certain time, or under a certain experimental condition. The monitored gene expression level provides an overall picture of the genes being studied.

* Corresponding author.
E-mail: rudys@comp.nus.edu.sg

It also reflects the activities of the corresponding protein under certain conditions.

Previously, most of the gene expression analyses were done manually with very limited information derived from the experiments. The focus of a molecular biologist was on a few select genes or proteins. With the application of large-scale biological information quantification methods like microarray and DNA sequencing, the behavior of genes can now be studied globally. At present, there is an increasing demand for automatic analysis of the various relationships hidden behind large amounts of genes from their expression. To achieve this, machine learning algorithms such as the self-organizing map (SOM) for unsupervised data clustering and the support vector machine (SVM) for supervised data classification can be expected to play very important roles.

This paper reports the results of our analysis using SOM and SVM on the gene expression data set of zebrafish. The data set has been collected at the Institute of Molecular and Cell Biology (IMCB) in Singapore. Some samples in the data set have been classified as members of one of the following functional categories: “Enzyme for metabolism”, “Protein, DNA, and RNA biosynthesis”, “Muscle specific protein”, “Cellular signaling”, “Transcription factor”, and “Splicing”; while many others remain un-
labeled. The research question that we are aiming to answer through our experiment is whether filtering the data samples by an unsupervised clustering algorithm, namely SOM, would improve the classification accuracy of a supervised learning method, in this case, SVM. The main idea involves discarding atypical samples, as discovered by SOM before the SVM classifier is built. Our experimental results show that indeed, such data filtering can improve the predictive accuracy of SVM.

System and Methods

Data Set

The experimental data set we used consists of a large number of samples with low dimensions. This data set includes developmental microarray data of zebrafish obtained from the Laboratory of Functional Genomics at IMCB, Singapore (1). In the microarray experiment, there were altogether 11,552 Expression Sequence Tag (EST) clones printed onto the microarray glass slides. According to BLAST (Basic Local Alignment Search Tool) search, 4,506 of the 11,552 clones have matches to 728 distinct publicly deposited sequences. It means that the functions of these 4,506 clones are known, while those of the remaining clones are unknown. The relative expression of the 11,552 clones at the different developmental stages of zebrafish was monitored via microarray experiments. Based on the developmental morphology of the fish, six stages are distinguished: cleavage (E2), gastrula (E3), blastula (E4), segmentation (E5), pharyngula (E6), and hatching (E7). The experimental data set was constructed with 11,385 samples from 11,552 clones with 6 features corresponding to the 6 developmental stages. A total 2,581 out of 11,385 samples corresponding to the known clones have been classified to the following functional categories: “Enzyme for metabolism”, “Protein, DNA, and RNA biosynthesis”, “Muscle specific protein”, “Cellular signaling”, “Transcription factor”, and “Splicing”.

Within each category, several subtypes have been explicitly labeled with specific names. For example, under the “Transcription factor” category, there exist three main subtypes of interest, namely, “zinc finger protein”, “homeobox protein”, and “other”. Table 1 lists out the overall known protein types of the sample set. Although each functional protein category contains several subtypes, each subtype itself is composed of a big family of proteins. It is known that only those data points labeled with “ribosomal protein” and “muscle specific protein” could probably follow certain pattern of gene expression levels, since they are both produced in large amounts and concentrated at the later stages of the zebrafish development. Characteristics for the other types are not apparent. Moreover, the protein amount needed for the zebrafish development varies significantly for various types of proteins. This phenomenon causes the extremely uneven distribution among data samples. For example, the number of labeled samples for “splicing” and “transcription factor” is rather small (around 60) compared to other types such as the “ribosomal protein”, which numbers 630 samples.

Table 1 Classification of Labeled Data

| Protein class          | Label | No. of samples | Protein category                  |
|------------------------|-------|----------------|-----------------------------------|
| Ase                    | A     | 144            | Enzyme for metabolism             |
| Ase-dehydrogenase      | A1    | 10             | Enzyme for metabolism             |
| Ase-DNA polymerase     | A2    | 31             | Enzyme for metabolism             |
| Ase-oxidase            | A3    | 123            | Enzyme for metabolism             |
| Ase-protein phosphatase| A4    | 960            | Cellular signaling                |
| Ase-synthase           | A5    | 48             | Enzyme for metabolism             |
| Ase-transferase        | A6    | 258            | Enzyme for metabolism             |
| Muscle specific protein| M     | 245            | Muscle specific protein            |
| Ribosomal protein      | R     | 630            | Protein, DNA, and RNA biosynthesis|
| Splicing protein       | S     | 66             | Splicing                          |
| Transcription factor-other| T   | 55             | Transcription factor              |
| TF-zinc finger protein | T1    | 5              | Transcription factor              |
| TF-homeobox protein    | T2    | 6              | Transcription factor              |
| Total                  |       | 2,581          |                                   |
Machine learning methods: SOM and SVM

SOM is an unsupervised algorithm often used to visualize and to help interpret large high-dimensional data sets (2, 3), and is one of the methods that can be useful for clustering, which is an important step in analyzing microarray data (4–6). In typical applications such as visualization of process states, this is achieved by representing the central dependencies within the data on the map. The map consists of a regular grid of processing neurons. A model vector for a group of multidimensional observations, normally a vector consisting of features, is associated with each unit. The map attempts to represent all observations with optimal accuracy using a restricted set of models. After the map has been trained, the models become ordered on the grid so that similar models are close to each other and dissimilar models far from each other.

The fitting of the model vectors $m$ is usually carried out by a sequential regression process, where $t = 1, 2, \ldots$ is the step index: for each sample $x(t)$, first the winner index $c$ (best match) is identified by the condition,

$$
\forall i, ||x(t) - m_c(t)|| \leq ||x(t) - m_i(t)||.
$$

Normally, the distance between samples and model vectors is calculated as the Euclidean distance. When this is done, all model vectors or a subset of them that belong to the nodes centered around node $c = c(x)$ are updated as $m_i(t + 1) = m_i(t) + h_{c(x),i}(x(t) - m_c(t))$. Here $h_{c(x),i}$ is the “neighborhood function”, a decreasing function of the distance between the $i^{th}$ and $c^{th}$ nodes on the map grid. Gaussian function is commonly used as the neighborhood function. This regression is usually repeated over the available samples. After sufficient cycles of the regression process, neurons tend to locate to the same position within the map grid. At this stage, SOM for the data set is trained and ready for the next task, which usually involves a closer analysis of the map.

The analysis can be done by studying the density of the data on SOM maps. The density or cluster structure of the data can be visualized by the so-called U-matrix computed from an SOM map. The U-matrix visualizes the distance between model vectors of neighboring neurons as different shades of gray. The density of the model vectors reflects the density of the data items, and the place where the distance between model vectors is smaller in clustered areas and larger in sparser areas.

The cluster found by the U-matrix can be further analyzed by one of the following three methods:
1. plotting class distribution;
2. plotting model vectors (In this case, the expression profiles are represented by the map units);
3. plotting the distribution of the original data variables, i.e. gene expression features.

Before we proceed with any of these methods, neurons in the U-matrix need to be classified and labeled. In our experiments, majority voting is chosen for the purpose of class labeling. Each neuron could possibly have several original data points that vote this neuron as the BMU (Best Matching Unit). Majority voting only labels the neuron with the class having the most instances.

SOM attempts to learn functionally significant classifications of genes in an unsupervised fashion, while SVM adopts the opposite approach (7, 8). SVM, as applied to gene expression data, begins with a set of genes with known classification, and finds an optimal separating hyperplane between members and non-members of a given class in an abstract space. Whereas unsupervised methods determine how a set of genes clusters into functional groups, SVM determines what expression characteristics of a given gene make it a part of a given functional group. This supervised method uses complex learning models that exploit the specific characteristics of the given functional group. As a supervised method, sufficient prior knowledge must be provided to SVM in terms of class membership (9, 10).

The idea of SVM is that each vector could be thought of as a point in $m$-dimensional space. SVM finds the maximum margin hyperplane, that is, the hyperplane that maximizes the minimum distance from the hyperplane to the closest training point, in the feature space that is normally of higher dimension than the original input space (11). Data samples in the original space are usually linearly non-separable, but in new higher dimensional feature space, they are more likely to be linearly separable. The transformation of data samples into higher dimensional space is achieved by defining a kernel function.

In the linearly non-separable case, there is no hyperplane that can separate all samples. In this circumstance, a set of slack variables $\{\xi_i\} (\xi_i \geq 0, i = 1, \ldots, n)$ is introduced, and the separation hyperplane is redefined as

$$
y_i(w^T x_i + b) \geq 1 - \xi_i \quad i = 1, \ldots, n,
$$

and the optimization problem becomes
minimize \( w^T w + C \sum_{i=1}^{n} \xi_i \)
subject to \( y_i(w^T x_i + b) \geq 1 \quad i = 1, \ldots, n, \)

where \( x_i \) represents the data for sample \( i \), \( y_i \) is \(-1\) or \(+1\), representing the class membership of sample \( i \), and the parameter \( C \) balances the generalization ability represented in the first term and the separation ability indicated in the second term of the objective function. The above linear program can be converted to its dual problem that does not involve the slack variables:

maximize \( W(\alpha) = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j x_i^T x_j \)
subject to \( \sum_{i=1}^{n} y_i \alpha_i = 0 \), \( 0 \leq \alpha_i \leq C \), \( i = 1, \ldots, n. \)

The optimal coefficient of the separating hyperplane can be computed from the optimal values of the dual variables as follows:

\[ w^* = \sum_{i=1}^{n_s} \alpha_{s_i} y_{s_i} x_{s_i}, \]

where \( n_s \) is the number of support vectors, and \( s_i \) \((i = 1, \ldots, n_s)\) is the index corresponding to those support vectors. To identify support vectors, we turn to the Karush-Kuhn-Tucker conditions of the linear program:

\[ \begin{cases} 
\alpha_i [y_i (w^T x_i + b) - 1 + \xi_i] = 0 & i = 1, \ldots, n, \\
\xi_i (\alpha_i - C) = 0 
\end{cases} \]

According to these conditions, all the sample vectors with positive Lagrange multipliers are support vectors; and the slack variable is non-zero only when its corresponding Lagrange multiplier equals to \( C \). The value of \( b^* \) can be determined by choosing any support vector \( x_i \) with Lagrange multiplier \( 0 < \alpha_i < C \):

\[ b^* = 1 - w^* T x_i, \quad \text{if} \quad y_i = +1, \]
\[ b^* = w^* T x_i - 1, \quad \text{if} \quad y_i = -1. \]

### Integrating SOM with SVM

In real circumstances, we obtained a large number of clones from microarray experiments, where only a small set of clones could be identified manually according to their functional class types. Therefore, we always get a gene expression matrix where only some of its rows could be explicitly labeled.

Current SVM method only extracts information from the training data set that is entirely labeled. However, for a data set that is partly labeled, SVM method simply ignores any information from the data set that is unlabeled. As described before, SOM is an unsupervised method. Without any prior knowledge, it can recognize the data clustering structures and detect possible functional classes within the data set. This unsupervised nature helps to grasp class information from the whole data set including both labeled and unlabeled samples. Incorporating this information could probably help SVM to learn the class characteristics better, and hence produce better performance in the big picture.

Another reason for combining SOM and SVM methods is because non-linear SVM determines the hyperplane that maximizes the minimum distance from the hyperplane to the closest training point in the feature space. The final hyperplane could be overly complex, which in turn results in deterioration in the classification accuracy. SOM could be applied to identify and remove probably noise samples before the data set is used to train an SVM.

Instead of directly applying SVM to the training set, we first analyze the available training samples using SOM. From the SOM map, for each tightly coupled class, we extract typical data points corresponding to this class. After this step, we can obtain a smaller but more representative set for each class. We expect SVM to build up better classifier from this filtered training set. The detailed steps of the SOM and SVM integrated approach are described below.

### Input

Training set \( T = [X; Y] \), where the matrix \( X \) and the vector \( Y \) are as follows:

\[
X = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1n} \\ x_{21} & x_{22} & \cdots & x_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ x_{m1} & x_{m2} & \cdots & x_{mn} \end{bmatrix}, \quad Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_m \end{bmatrix}
\]

Here \( x_{ij} \) represents the expression value of the \( i^{th} \) gene in the \( j^{th} \) hybridization, where \( 1 \leq i \leq m \) and \( 1 \leq j \leq n \). In each row \( x_i \) is the vector for the \( i^{th} \) gene. The element \( y_i \) represents the class label for the \( i^{th} \) gene. For example, \( y_2 = 3 \) indicates that the second gene is labeled 3, which corresponds to the ase-oxidase. The entries of vector \( y \) could be null, since the genes are possibly unlabeled for some gene expression data.
Hybrid SOM-SVM Approach

Output

$X'$, a subset of $X$, which is expected to better represent the classes.

Steps

1. Train the SOM map with the matrix $X$, and a set of $k$ neurons $M = \{m_1, m_2, \ldots, m_k\}$ on the map is obtained after the training.
2. For a neuron $m_i \in M$, find the set of genes for which $m_i$ is their BMUs. Count the frequency of each class type for this set of genes. And $f_{ij}$ is the frequency of class $C_j$ for neuron $m_i$.
3. For each class $C_j$, let $N_j$ be the set of all neurons $n_j$ where $f_{ij} \geq t$, in which $t$ is a threshold.
4. For the set $N_j$, find the set of genes $S_j$ for which neurons in $N_j$ are their BMUs, and with class label $C_j$.
5. Repeat Steps 3 to 5 for each class type. Thus obtain a new training set $X'$ where $X' = [S_1 \cup S_2 \cup S_3 \cup \cdots \cup S_n]$ for all classes from $C_1$ to $C_n$.
6. Train SVM with the new training set $X'$ and its corresponding class label $Y'$.

In our experiments, genes from some functional classes may not possess similar expression profiles, thus such classes show no cluster structure and they are possibly scattered all over the map grid. One approach to overcome this problem is to recognize only those functional classes that exhibit clear cluster structures inside SOM, use SVM to train the classifiers for those classes, and use the resulting classifiers to discriminate between members and non-members of such functional classes in the test set. The SVM training using loosely coupled classes is expected to result in a relatively low performance accuracy in discriminating between members and non-members of the classes compared to the SVM that has been trained using more tightly coupled classes.

Results and Discussion

Table 2 summarizes the statistics of the data set that we used in our experiments. The total number of samples used is 2,371. We discarded samples that belong to Classes T1 and T2 because there are too few of them. Samples that belong to Classes Ase and TF were also discarded because they are not expected to exhibit uniform within-class expression pattern. Note that all the samples used are labeled with their classes. The genes belong to one of the possible classes: ase-dehydrogenase (A1), ase-DNA polymerase (A2), ase-oxidase (A3), ase-protein phosphatase (A4), ase-synthase (A5), ase-transferase (A6), muscle specific protein (M), ribosomal protein (R), and splicing protein (S).

Due to the biological difficulty of gene labeling, there are only a limited number of gene clones that can be successfully sequenced, among which only a small portion is found to have matches with the public gene sequence database. The majority of elements from the original data set are unlabeled, so it is impossible for us to measure the performance of our approach on the whole original data set consisting of 11,450 samples to verify the improvement of the performance. However, we do have 2,371 labeled samples. To verify if there is any improvement in the performance of the SVM method with data filtering using SOM, we divided these labeled data samples randomly into three sets as summarized in Table 2. One set includes 1,200 elements for training purpose, the other two consist of 600 and 571 samples respectively for testing purpose.

Table 2 Summary of the Training and Test Sets

| Protein class                  | No. of training set | No. of test set 1 | No. of test set 2 |
|-------------------------------|---------------------|-------------------|-------------------|
| Ase-dehydrogenase (A1)        | 5                   | 2                 | 3                 |
| Ase-DNA polymerase (A2)       | 16                  | 9                 | 6                 |
| Ase-oxidase (A3)              | 59                  | 37                | 27                |
| Ase-protein phosphatase (A4)  | 458                 | 236               | 266               |
| Ase-synthase (A5)             | 21                  | 12                | 15                |
| Ase-transferase (A6)          | 129                 | 67                | 62                |
| Muscle specific protein (M)   | 137                 | 50                | 58                |
| Ribosomal protein (R)         | 344                 | 139               | 147               |
| Splicing protein (S)          | 31                  | 19                | 16                |
| Total                         | 1,200               | 571               | 600               |
The experiments were conducted as outlined in Figure 1. Accuracy 1 was obtained from the SVM that has been trained using the original 1,200 samples. Accuracy 2, on the other hand, was obtained from the SVM that has been trained only using the samples filtered by SOM. Sample selection using SOM resulted in a greatly reduced training data set consisting of 397 samples. SOM may help to remove the noise from the data and select only typical vectors before we train the data with SVM. Thus it may increase the performance accuracy of the classifiers.

When the SOM clustering was being performed, the samples in the test set were included without their label information. In Figure 1, this is indicated as the cross-validation data. The selected training data set consists of 397 samples, all of which are part of the 1,200 original training data samples; none is from the cross-validation set. Once SVM has been constructed using these 397 samples, in order to test its generalization ability, the labels of the samples in the cross-validation test were restored.

When all the 2,371 samples were clustered by SOM, the map in Figure 2 was obtained. In this figure, the U-matrix visualizes the distance between model vectors of neighboring map units as gray levels, thus the density of cluster structure of the data can be easily analyzed. The following parameter setting used for SOM training is summarized in Table 3. More detailed information about these parameters can be found in other study (3).

Each map unit has been labeled with at most one label indicating the class that this map unit should belong to. We adopted the majority voting here. Each map unit could be the BMU for several sample data points, but only the class label with the highest frequency has been chosen as the label for this map unit.

The map in Figure 2 suggests that the labeling of the SOM units no doubt helps significantly in recognizing the cluster structures that are present in the U-matrix. For example, for the ribosomal proteins, it is observed that the labels clustered at the lower right part in the SOM, while a corresponding bright area was located at the lower right part in the U-matrix. In addition to text labeling, color labeling could also be employed to obtain more information from the trained SOM units.

Color labeling better visualizes the distribution structure of various label types as shown in Figure 3. Basically, each label is represented by a unique color, thus each map unit could be colored correspondingly. By visual inspection of the positions of the classes, seven distinct regions can be identified on the SOM map in this figure. Two regions corresponding to Classes A3 and A6 are located at the upper half of the map. In contrast, another group of Classes A5, R, M, and S lie at the lower half of the map. As neurons with the majority of samples from Class A4 do not form a cluster with a well-defined boundary, we considered the region not covered by the other classes to correspond to neurons from Class A4, as the elements in the U-matrix show mostly blue, the color assigned to this class. Samples that do not belong to the corresponding majority class in the seven regions were discarded. As a result, 397 samples remained.
The cluster map of 2,371 samples by SOM. Left: the U-matrix. Right: the corresponding SOM with text labels. The vertical bar on the right side uses gray level to indicate the distance (with the brightest color representing distance 0.092 and the darkest color representing distance 1.35) between map units. The brighter the shades in the U-matrix, the nearer the map units are to their neighbors. Hence, the light shades are likely to represent clusters, while the dark shades represent the sparser areas or gaps in between clusters. One clear bright shade at the right lower part in the U-matrix is expected to represent one salient class. After plotting class distribution and labeling the SOM units, the bright shade is confirmed to represent the class "ribosomal protein". The triangle area in the right map corresponds to the triangle area in the left map.

The U-matrix with labels A3 (black), A4 (blue), A5 (yellow), A6 (purple), M (green), R (red), and S (sky blue).
Table 3 Parameter Setting for the SOM Training

| Parameter                        | Value                                      |
|----------------------------------|--------------------------------------------|
| Training Mode                    | Long Training                              |
| Map Size (units)                 | $45 \times 23$                             |
| Training Algorithm               | Batch Training                             |
| Initialization                   | Linear                                     |
| Lattice                          | Hexa                                       |
| Rough Training Phase             | 16 Epochs (37 s)                           |
| Fine Training Phase              | 64 Epochs (153 s)                          |
| Learning Rate $\tau$: Rough/Fine Training | 0.5/0.05                                  |
| Radius $\rho$: Initial/Final radius | 6/1                                       |
| Final Quantization Error         | 0.606                                      |
| Final Topographic Error          | 0.026                                      |

and they were used to construct a second SVM to see if the data filtered by SOM can indeed improve the performance of SVM classifiers.

In order to assess the classification performance of SVM, we have fixed the parameters required by the SVM package we used, which is svm 0.54 in Matlab 6.5. We use the radial basis function as the kernel function

$$K(X, Y) = \exp \left( -\frac{|\bar{X} - \bar{Y}|^2}{2\sigma^2} \right),$$

where $\bar{X}$ and $\bar{Y}$ are normalized feature vectors, and $\sigma$ is the standard deviation of the Gaussian distribution.

The experiment was done in Windows 2000 environment with the default parameter setting of $C = 100$ and $\delta = \frac{1}{2\sigma^2} = 0.5$. Since SVM can only separate two classes at one time, and in the experiments we need to classify samples with $N = 7$ classes, we need to divide the classification problem into $N$ smaller classification problems. Each time we separate one class from the other $N-1$ classes, and obtain the performance accuracy of the classifier for this problem. When we obtain $N$ classifiers for $N$ different classes, we can then tabulate and analyze these performance accuracies to gauge the overall performance of SVM on this data set. The results are summarized in Figures 4 and 5 for the two different test sets. The accuracy measures are given in terms of two geometric means ($gm$) that take into account the imbalance in the data distribution.

A sample is considered as a positive sample if it belongs to the class of interest. The performance of each classifier is measured by examining how well the classifier identifies the positive and negative samples for each class in the test set. Here we consider all the samples that do not belong to the specified class as

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{gm1_test_600.png}
\caption{Comparison of the $gm_1$ and $gm_2$ results obtained on the data set test_600 when SVM was constructed using the original training set (1,200 samples) and the training data set filtered by SOM (397 samples), respectively.}
\end{figure}

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{gm2_test_571.png}
\caption{Comparison of the $gm_1$ and $gm_2$ results obtained on the data set test_571 when SVM was constructed using the original training set (1,200 samples) and the training data set filtered by SOM (397 samples), respectively.}
\end{figure}
negative samples. Most of the classification methods return a rank ordering of the test set. Accordingly, each gene in the test set can be labeled in one of the four ways: false positive (FP), false negative (FN), true positive (TP), and true negative (TN), where FP is the incorrect prediction that a sample is positive, FN is the incorrect prediction that a sample is negative, TP is the correct prediction that a sample is positive, and TN is the correct prediction that a sample is negative.

The accuracy (AC) is the proportion of the total number of predictions that are correct. It is determined using the equation \( AC = (a + d)/(a + b + c + d) \), where \( a, b, c, d \) is the number of TN, FP, FN, and TP, respectively. The recall or true positive rate (TPR) is the proportion of positive cases that are correctly identified, computed as \( TP = d/(c + d) \), while the false positive rate (FPR) is the proportion of negative cases that are incorrectly classified as positive, computed as \( FP = b/(a + b) \). The true negative rate (TNR) is defined as the proportion of negative cases that are classified correctly, computed as \( TN = a/(a + b) \), and the false negative rate (FNR) is the proportion of positive cases that are incorrectly classified as negative, computed as \( FN = c/(c + d) \). Finally, precision (P) is the proportion of the predicted positive cases that are correct, computed as \( P = d/(b + d) \).

Note that the classes within the given data set are highly imbalanced in general, since the number of negative samples is quite large compared to that of positive samples for each class. Instead of reporting the performance accuracy in terms of \( AC, TP, FP, TN, FN \), or \( P \), we computed our results as:

\[
\begin{align*}
gm_1 & = \sqrt{TP \times P} \\
gm_2 & = \sqrt{FP \times TN}
\end{align*}
\]

For Test set 1, SVM accuracies for Classes A3 and A5 increased sharply from 0 to 0.72 and 0.68 in their \( gm_2 \) values, respectively. A similar improvement was observed for Test set 2, where the accuracies increased from 0.20 to 0.52 and from 0 to 0.29, respectively. Note that Classes A3 and A5 are the two smallest sets among the functional classes, yet they show salient cluster structure on the SOM map (Figure 3). These encouraging results suggest that we have successfully removed the noises in the training data for the two classes and captured the most important representative genes by means of SOM selection. Since the two classes have much fewer samples, noises of such classes influence the performance of SVM much more than those of the classes with larger number of samples. For the rest of the classes, although the size of the new training set is only about one third of the original training set, the SVM performance did not really degrade much. The slight difference between accuracies of the two training sets is expected, because the new training set is a much smaller subset of the original training set. Since Classes A4 and A6 exhibit fairly distinct cluster structure on the SOM map (Figure 3) and the new training set is only one third of the size of the original training set, a small decrease in accuracy is expected. At the same time, for large classes that show tightly coupled structures such as Classes M, R, and S, the SVM performance was slightly better. The similar accuracies obtained imply that the impact of noises for these classes is small, yet the selected genes are representative enough for the original gene sets.

Although Test sets 1 and 2 are two distinct data sets, the test performance on the two sets was consistent. This convinces us on the reliability of the experiment results. The approach above is quite important for gene selection. If SVM can generally achieve similar or better accuracy when trained using the filtered training set than the accuracy obtained from the original set, we are confident that this SOM-based gene selection method is indeed effective. If the accuracy of the SVM is improved, we can then be more confident of the classification given by this SVM on unlabeled samples.

A possible reason for the significant improvement in the performance accuracy for classes with small number of samples is as follows. Each sample that belongs to these classes will play a more important role, or is more likely to become a support vector. Also the samples in the test set are generally located differently in the feature space from those of the training set, so they may have totally different set of support vectors, if we train an SVM on both training set and test set. It means that the classifier we get after training on the training samples alone cannot represent the class characteristics of all samples. Thus the classifier that we get after training on the original training set will have poor performance accuracy when tested on new samples. However, after the SOM training, critical data points whose BMUs are the critical neurons on the map grid are selected. The critical neuron for one class is determined by the number of the data points of this class that voted this neuron as BMU. Once the number of the supporting data points is above the threshold, the corresponding neuron is considered as the critical neuron of this particular class. For example, if a neuron N has 10 supporting data points of Class A, it means that these 10 data points voted the neuron N as their BMU. This neuron N has many
data points that votes $N$ as their BMU, and not all of these data points are of Class A. We include the data points that belong to Class A as critical points for Class A only if the number exceeds the threshold. The threshold setting is determined by experimental statistics. In our experiments, for most classes, we set the threshold $t$ equal to 2, because the data set is fairly small. The selection of such critical data points is based on the pattern of the whole data set, that is, for the classes with the labeled training set, as well as the unlabeled cross-validation set. However, for the classes with a large number of samples, each sample will play a relatively less important role, or have less possibility to become a support vector. Thus the classifier we get after training on the training set is supposed to have a better representing ability for the whole class of members. Thus there will not be significant change of the performance accuracy for such classifiers after the training samples have been filtered by SOM.

In summary, from our experiments, the major finding is that our approach that combines SOM and SVM is useful for building better SVM classifiers, particularly for small salient functional classes.

Conclusion

We have presented an integrated approach to improve the accuracy of classification of genes based on their expression levels. The two components of the proposed approach are the unsupervised clustering method SOM and the supervised classification method SVM. SOM is used to cluster the training samples and to select the more representative data samples for building the SVM classifier. For the gene expression data of zebrafish, our experimental results show that there is a significant improvement in the accuracy of the SVM classifier built using only the selected training samples, especially for classes with few samples. Our finding is important, because a more accurate classifier can be used to classify unlabeled samples with greater confidence. Often, biologists are more interested in finding those genes that belong to the minority class.

References

1. Lo, J., et al. 2003. 15,000 unique zebrafish EST clusters and their future use in microarray for profiling gene expression patterns during embryogenesis. *Genome Res.* 13: 455-466.
2. Kohonen, T. 1990. The self-organizing map. *Proc. IEEE* 78: 1464-1480.
3. Kohonen, T. 1997. *Self-Organizing Maps* (second edition). Springer, Berlin, Germany.
4. Datta, S. and Datta, S. 2003. Comparisons and validation of statistical clustering techniques for microarray gene expression data. *Bioinformatics* 19: 459-466.
5. Kaski, S., et al. 2001. Analysis and visualization of gene expression data using self-organizing maps. In *Proceedings of the 2001 IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing*, Baltimore, USA.
6. Tamayo, P., et al. 1999. Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation. *Proc. Natl. Acad. Sci. USA* 96: 2907-2912.
7. Vapnik, V. 1995. *The Nature of Statistical Learning Theory*. Springer, New York, USA.
8. Vapnik, V. 1998. *Statistical Learning Theory*. John Wiley & Sons, New York, USA.
9. Furey, T.S., et al. 2000. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics* 16: 906-914.
10. Cai, J., et al. 2000. Classification of cancer tissue types by support vector machines using microarray gene expression data. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology*, La Jolla, USA.
11. Brown, M.P., et al. 2000. Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proc. Natl. Acad. Sci. USA* 97: 262-267.
12. Kubat, M., et al. 1997. Learning when negative examples abound. In *Proceedings of the Ninth European Conference on Machine Learning*, pp. 146-153, Prague, Czech Republic.

The extended abstract of the paper appeared in *the Proceeding of the 17th International Conference on Pattern Recognition (ICPR 2004)*, pp. 323-326, Cambridge, UK.