A DSRPCL-SVM Approach to Informative Gene Analysis

Wei Xiong, Zhibin Cai, and Jinwen Ma*

School of Mathematical Sciences and Laboratory of Mathematics and Applied Mathematics (LMAM), Peking University, Beijing 100871, China.

Microarray data based tumor diagnosis is a very interesting topic in bioinformatics. One of the key problems is the discovery and analysis of informative genes of a tumor. Although there are many elaborate approaches to this problem, it is still difficult to select a reasonable set of informative genes for tumor diagnosis only with microarray data. In this paper, we classify the genes expressed through microarray data into a number of clusters via the distance sensitive rival penalized competitive learning (DSRPCL) algorithm and then detect the informative gene cluster or set with the help of support vector machine (SVM). Moreover, the critical or powerful informative genes can be found through further classifications and detections on the obtained informative gene clusters. It is well demonstrated by experiments on the colon, leukemia, and breast cancer datasets that our proposed DSRPCL-SVM approach leads to a reasonable selection of informative genes for tumor diagnosis.

Key words: microarray data, informative gene selection, clustering analysis, DSRPCL, tumor diagnosis

Introduction

With the rapid development of DNA microarray technology, we can now get the expression levels of thousands of genes via one single experiment with a relative cheap cost (*1*, *2*). These microarray data are essential in analyzing health situation of the human body and recognizing symptoms of human illnesses. Mathematically, they can be expressed as a gene expression matrix $X = (x_{ij})_{n\times m}$, where each row represents a gene, while each column represents a sample or a patient for tumor diagnosis. That is, the numerical value x_{ij} denotes the expression level of a specific gene i at a particular sample j. As a matter of fact, there are many microarray datasets available on the web.

For medical diagnosis and treatment, it is very important to select or discover informative genes of a tumor via the analysis of microarray data, since the informative genes can not only provide valuable information for discovering the crucial reasons of the tumor as well as the treatment methods, but also support to construct an efficient tumor diagnosis system from their expression levels directly without any influence of the other irrelevant genes. Actually, there have already been many elaborate methods for informative gene selection. However, most of them are based on ranking genes according to a kind of criterion, such as t, F, rank sum and χ^2 test statistics (*3*–*7*) and the information criterion (*8*). Recently, the up- and down-regulation probabilities for each gene were defined and then the informative genes can be successfully selected according to the decrease rank of the absolute difference values between the two regulation probabilities (*9*, *10*). Generally, these statistical and information methods just select a number of top genes (the number is fixed or determined by the threshold given to the criterion). In this way, informative genes are selected through individual gene evaluations and thus the relations among the genes are neglected, which may lead to an incomplete selection of informative genes for tumor analysis and diagnosis.

The relations or structures of genes can be discovered through unsupervised classification or clustering. In fact, some typical clustering methods have been already applied to the analysis of informative genes and tumor diagnosis, such as hierarchical clustering (*4*, *11*, *12*), k-means (*13*) and self-organizing map (SOM) (*3*). However, as the number of genes is large, hierarchical clustering is very difficult to be

***Corresponding author.**
E-mail: jwma@math.pku.edu.cn

implemented and cannot determine the informative genes by itself. Moreover, the other classical clustering methods like k-means and SOM require predetermination of the number of gene clusters. However, we usually do not know the number of gene clusters since it depends on the structures of genes that are implied in the microarray data.

In 1993, a new kind of unsupervised clustering method, called the rival penalized competitive learning (RPCL) algorithm (*14*, *15*), was proposed to automatically determine the number of clusters during the clustering or competitive learning on the sample data. For each input sample, the basic idea is that not only the weight vector of the winner unit is modified to adapt to the input, but also the weight vector of its rival (the 2nd winner) is de-learned by a smaller learning rate. In this way, as the learning and de-learning rates are properly selected and the number of units or weight vectors is larger than the number of actual clusters in the sample data, the RPCL algorithm can automatically allocate an appropriate number of weight vectors for a sample dataset, with the other extra weight vectors being driven far away from the sample data. Recently, the RPCL algorithm was generalized to the distance sensitive rival penalized competitive learning (DSRPCL) algorithm through a cost function theory (*16*). Actually, the implementation of the DSRPCL algorithm becomes more efficient and easier since the learning rate can be easily set. Thus, we can use the DSRPCL algorithm to analyze gene clusters from microarray data without knowing the actual number of gene clusters. In fact, the RPCL and DSRPCL algorithms have already been used to analyze microarray data (*17*, *18*). In Nair *et al* (*17*), the RPCL algorithm was applied to classify the genes into a number of clusters that could have certain functional meanings. Moreover, the DSRPCL algorithm was utilized to classify the genes into a number of clusters from which a compact set of informative genes could be established via a post-filtering gene selection method (*19*). Similarly to the DSRPCL algorithm, the cooperative competition clustering algorithm (*20*) was established to classify the genes into an appropriate number of clusters and then the informative genes can be selected from each of the obtained clusters.

As the DSRPCL algorithm can automatically divide the genes into a number of functional clusters, we may wonder whether there exists a cluster that can be served as a set of informative genes directly. Actually, we can utilize support vector machine (SVM) (*21*) to check which gene cluster contributes best to the tumor diagnosis and whether it can be used as an informative gene set for a tumor. That is, we can use SVM to train a tumor diagnosis system with the m expression profiles on the genes in each cluster and to see which tumor diagnosis system gets the best prediction accuracy. If a tumor diagnosis system gets the best prediction accuracy that is high enough, we consider the corresponding gene cluster is just the set of informative genes for the tumor. Moreover, since the distributions of gene expression levels on the samples should have the similar structures at the critical or powerful informative genes, that is, the informative genes that can strongly discriminate the tumor from the normal via their expression levels on the samples, we can consider these critical or powerful informative genes belong to a sub-cluster in this informative gene set. Thus, we can find these critical or powerful informative genes of the tumor by further clustering and checking on the sub-clusters of this informative gene cluster or set. As for the critical informative gene selection, Guyon *et al* have already proposed an SVM-based method (*22*). They used a linear SVM to train a tumor diagnosis system and selected the genes with higher weights in the final discriminate function as the critical or powerful genes. Recently, Zhang *et al* improved this SVM-based method by adding the smoothly clipped absolute deviation penalty on the original objective function of the SVM (*23*). However, the major disadvantage of the SVM-based method is that the number of informative genes is still determined by the pre-assumed threshold value to the weights, not by the structure of the genes.

In this paper, in light of the above ideas, we propose a DSRPCL-SVM approach where the DSRPCL algorithm is firstly utilized to classify the genes expressed through the microarray data into a number of clusters and the informative gene cluster or set is then detected with the help of SVM. Moreover, the critical or powerful informative genes is found through further classifications and detections on the informative gene clusters. The performance of this method is demonstrated by experiments on the colon, leukemia, and breast cancer datasets.

Method

The DSRPCL algorithm for gene clustering

Given a microarray dataset $X = (x_{ij})_{n\times m}$ with n genes and m samples, we let $S = \{X^i\}_{i=1}^n$, where

$X^\mu = [x_{\mu 1}, x_{\mu 2}, \cdots, x_{\mu m}]^T$ represents the μ-th gene through its expression levels over all the m samples. Suppose that X^μ is just an input to a simple competitive learning network, that is, a layer of k competitive units. These competitive units are dominated by the corresponding weight vectors $W_i = [w_{i1}, w_{i2}, \cdots, w_{im}]^T$ for $i = 1, 2, \cdots, k$. All the weight vectors can be represented by a big vector $W = \text{vec}[W_1, W_2, \cdots, W_k]$. For each input X^i, the basic idea of the DSRPCL algorithm is that not only the weight vector of the winner unit (the closest weight vector to the sample) is modified to adapt to the input, but also the weight vectors of the rivals or losers (the other weight vectors) are punished to keep away from the input. As a weight vector diverges to infinity, the corresponding cluster becomes empty and can be canceled. Therefore, we can automatically obtain the number of gene clusters as well as the centers of these clusters assuming k is larger than the true number of the actual gene clusters. As a result, the genes are automatically divided into several clusters by classifying each gene into the cluster whose center is closest to it.

Theoretically, the DSRPCL algorithm can be realized by minimizing the cost function in Equation 1, where $c(\mu)$ is the index of the winner unit for the μ-th gene, $W_{c(\mu)}$ is the nearest weight vector for X^μ, and P is a positive constant. Ma and Wang (*16*) obtained the derivatives of $E(W)$ with respect to w_{ij} as in Equation 2, where $\delta_{i,j}$ is the Kronecker function. With these derivatives, the DSRPCL algorithm is designed as a kind of gradient-descent algorithm. Table 1 summarizes the details of the DSRPCL algorithm and its variants, where we denote it as the batch DSRPCL algorithm. The DSRPCL1 algorithm is the adaptive DSRPCL algorithm, and the DSRPCL2 algorithm modifies only the rival weight vector (the second winner) so that $E_2(W)$ is only affected by the largest term with $r(\mu)$, which is consistent with the original RPCL algorithm (*14*, *15*). The other variant

$$E(W) = E_1(W) + E_2(W) = \frac{1}{2}\sum_{\mu} \|X^\mu - W_{c(\mu)}\|^2 + \frac{2}{P}\sum_{\mu, i\neq c(\mu)} \|X^\mu - W_i\|^{-P} \tag{1}$$

$$\frac{\partial E(W)}{\partial w_{ij}} = -\sum_{\mu} \delta_{i,c(\mu)}(x_j^\mu - w_{ij}) + \sum_{\mu,i}(1-\delta_{i,c(\mu)})\|X^\mu - W_i\|^{-P-2}(x_j^\mu - w_{ij}) \tag{2}$$

Table 1 The DSRPCL algorithm and its variants

1 Randomly initialize the vector $W_1^{(0)}, \cdots, W_k^{(0)}$, and let $T = 0$.

2 Update W_i with a learning rate η $(0 \le \eta \le 1)$:

1) Batch DSRPCL:

$$\Delta W_i = -\eta\frac{\partial E(W)}{\partial W_i} = \begin{cases} \eta\sum_\mu (X^\mu - W_i), & \text{if} \quad i = c(\mu), \\ -\eta\sum_\mu \|X^\mu - W_i\|^{-P-2}(X^\mu - W_i), & \text{otherwise.} \end{cases}$$

2) DSRPCL1:

$$\Delta W_i = \begin{cases} \eta(X^\mu - W_i), & \text{if} \quad i = c(\mu), \\ -\eta\|X^\mu - W_i\|^{-P-2}(X^\mu - W_i), & \text{otherwise.} \end{cases}$$

3) DSRPCL2:

$$\Delta W_i = \begin{cases} \eta(X^\mu - W_i), & \text{if} \quad i = c(\mu), \\ -\eta\|X^\mu - W_i\|^{-P-2}(X^\mu - W_i), & \text{if} \quad i = r(\mu), \\ 0, & \text{otherwise.} \end{cases}$$

4) SARPCL:

a) Let $\lambda = e^{(-k_1 T - k_0)}$, $\eta = \eta_0/(c_1\ T + c_0)$ and $t = 0$.

b) Randomly select X^μ from $S = \{X^1, \ldots, X^n\}$, and take $\xi \sim$Uniform$[0, 1]$.

c) $$\Delta W_i = \begin{cases} \eta(X^\mu - W_i), & \text{if} \quad i = c(\mu), \\ -\eta\|X^\mu - W_i\|^{-P-2}(X^\mu - W_i), & \text{otherwise.} \end{cases}$$

If $\xi \le \lambda$, let $\Delta W_i = -\Delta W_i$.

d) If $t < M$, let $t = t + 1$ and return to STEP b).

e) If $\lambda < \varepsilon$, stop.

3 If $|E(W)^{(T+1)} - E(W)^{(T)}| > \varepsilon_1$, let $T = T + 1$, and return to STEP 2; otherwise, stop.

of the DSRPCL algorithm is the simulated annealing rival penalized competitive learning (SARPCL) by applying the simulated annealing mechanism to the DSRPCL1 algorithm. The stopping threshold value ε is a pre-fixed small positive number. Parameters k_0, k_1, c_0 and c_1 are positive constant numbers that can be selected by experience. Since these DSRPCL algorithms have the similar functions in clustering analysis, we typically use the DSRPCL1 algorithm in our experiments.

According to the properties of the DSRPCL algorithm shown in Ma and Wang (*16*), when k is selected to be large enough, the DSRPCL algorithm can detect the number of gene clusters during the clustering. From the obtained gene clusters, we can check them with SVM and find the informative gene cluster or set.

The DSRPCL-SVM approach to informative gene analysis

We further consider the informative gene analysis through the DSRPCL algorithm and SVM. In order to do so, we can implement the DSRPCL algorithm directly on the sample data $S = \{X^\mu\}_{\mu=1}^n$ from the microarray data with respect to a tumor. Usually, we can overestimate the number of the clusters in S and set it to be k. As a result of the implementation, the DSRPCL algorithm divides the n genes (represented by X^μ) into a number of functional gene clusters among which there is an informative gene cluster contributing the best to the recognition or diagnosis of the tumor. If we think this informative gene cluster is too large, that is, it contains too many genes for tumor analysis, or we just want to get some more critical or powerful informative genes for the tumor, we can implement the DSRPCL algorithm on the previously obtained informative gene cluster or set (the obtained subset of S). Again, the DSRPCL algorithm divides the selected informative genes into a number of gene clusters, among which we can find the most powerful sub-cluster and genes. In such a way, we can finally find a small number of critical or powerful genes for the tumor.

On the other hand, the above informative gene search can be only done with the help of a supervised learning system for checking which gene cluster or sub-cluster contributes the best to the recognition or diagnosis of the tumor. Actually, suppose $G_s = \{g_{i_1}, \cdots, g_{i_p}\}$ is a divided gene cluster or sub-cluster obtained by the DSRPCL algorithm, if the genes in this cluster (or sub-cluster) are informative (or powerful informative) to the tumor, then a supervised learning system on the sample expression profiles of these genes, $\hat{X}^j = [x_{i_1,j}, \cdots, x_{i_p,j}]^T (j = 1, \cdots, m)$, and their corresponding diagnosis results (if the sample is tumorous, the diagnosis result is 1; otherwise the diagnosis result is 0) will lead to the highest prediction accuracy or the lowest average error. In this way, we can find the informative or powerful informative genes through a supervised learning system. SVM (*21*) is a powerful learning machine for the supervised learning or classification. In fact, many researchers have used it to analyze microarray data and demonstrated its advantages (*24–26*). Therefore, we also utilize SVM to check the informative or powerful informative gene cluster. For comparison, we use the MATLAB toolbox OSUSVM 3.0 to set up three kinds of SVMs, including linear SVM (no kernel), 3-poly SVM (cubic polynomial kernel), and radial basis function SVM (RBF kernel).

Evaluation

To test the effectiveness of our proposed DSRPCL-SVM approach for informative gene analysis, we conducted experiments on three real-world microarray datasets:

Colon cancer dataset. It contains expression profiles of 2,000 genes in 22 normal tissues and 40 colon tumor tissues, which can be retrieved from the web at http://microarray.princeton.edu/oncology/affydata/index.html. In our experiment, we used the training set (22 normal and 22 tumorous tissues) and the test set (18 tumorous tissues) provided at the website.

Leukemia dataset. It contains expression profiles of 7,129 genes in 47 acute lymphoblastic leukemia (ALL) and 25 acute myeloid leukemia (AML) samples, which can be retrieved from the web at http://www.genome.wi.mit.edu/MPR/data_set_ALL_AML.htm. In our experiments, we used the training set (27 ALL and 11 AML) and the test set (20 ALL and 14 AML) provided at the website.

Breast cancer dataset. It contains expression profiles of 5,776 genes in 14 normal tissues and 13 breast tumorous tissues, which can be retrieved from the web at http://genome-www.stanford.edu/breast_cancer/sbcmp/data.shtml. In our experiment, we used the 14 normal and 13 tumorous samples provided at the website as both the training set and the

test set since the number of the samples is very limited.

For the three kinds of SVMs, there are two parameters in the last two SVMs, γ and C, and their selection affects the performance of SVM. In our experiments, by experience we set $\gamma = 0.02, C = 0.05$ on colon cancer data and $\gamma = 0.002, C = 10$ on leukemia data and breast cancer data.

To improve the efficiency of the DSRPCL1 algorithm, we set the de-learning rate in the update rule, that is, the corresponding learning rate for the opposite or minus direction learning, to attenuate to zero with the number of iterations. In this way, the algorithm could make the convergent weight vectors converge to the centers of the actual clusters in the sample data without any deviation, keeping the extra weight vectors being driven far away from the sample data.

For the convenience of implementation of the two algorithms, we filtered out 10% obviously irrelevant genes using the cosine method. That is, we computed the cosine value of the vector of the expressions of each gene at all the samples with the reference vector in which each element is 1, and then ranked these genes with the decrease of the cosine value and finally filtered out 10% genes from the last. Moreover, we normalized the expression vectors of the remaining genes for our analysis.

First, we ran the DSRPCL1 algorithm on the three microarray datasets and obtained five, four, and nine gene clusters (except the empty clusters represented by the weight vectors of the DSRPCL1 algorithm), respectively, where the number of the weight vectors or clusters was always set initially by 10 (the number of clusters in each microarray data was assumed to be no more than 10). According to these gene clusters obtained by the DSRPCL1 algorithm, we trained the three kinds of SVMs with the training data on each gene cluster of the three datasets. Then we obtained the prediction or classification accuracy of each trained SVM with the test data. The experiment results on the three datasets are summarized in Table 2.

From Table 2, we can find that the DSRPCL1 algorithm not only discovers a good information gene set to a tumor, but also improves the accuracy of tumor diagnosis or classification. Actually, in some cases, the SVM on the selected informative gene set (cluster) can even reach 100% prediction accuracy. This means that the optimal gene cluster obtained from a DSRPCL-SVM procedure on the microarray data is just the informative gene set to the tumor.

Next, we repeated the above DSRPCL-SVM procedure (DSRPCL clustering and SVM checking) on the optimal or informative gene cluster. That is, we just considered the genes in the optimal cluster and

Table 2 Experimental results of the first DSRPCL-SVM procedure on three datasets

Dataset	Gene cluster	No. of genes	Classfication accuracy		
			Linear SVM	Poly SVM	RBF SVM
Colon cancer	1	381	0.8889	0.9444	0.3889
	2	182	0.7778	0.5556	0.3889
	3	385	0.8889	0.8889	0.5000
	4 (optimal)	435	0.9444	0.8889	1
	5	418	0.8333	0.9444	0.8889
Leukemia	1	2,769	0.9545	0.9545	0.8182
	2	1,939	0.8182	0.7727	0.7273
	3 (optimal)	1,708	0.9545	0.9545	0.9545
	4	1	0.1818	0	0
Breast cancer	1	3	0.6667	0.5556	0.5185
	2	2	0.4444	0.5556	0.5185
	3 (optimal)	1,580	1	1	1
	4	926	1	1	1
	5	771	1	1	1
	6	486	1	1	1
	7	175	1	1	1
	8	672	1	1	1
	9	584	1	1	1

neglected all the other genes. In this way, we clearly obtained a smaller optimal gene cluster or sub-cluster. If the SVM on this smaller optimal gene cluster still has a high prediction accuracy, we could consider the genes in this cluster are more powerful. In other words, they are powerful informative genes to the tumor. From the second DSRPCL-SVM procedure, we also obtained a number of sub-clusters of genes and their prediction accuracies on the three datasets (Table 3).

From Table 3, we can find that, after the second DSRPCL-SVM procedure on each microarray dataset, a smaller informative gene cluster was obtained, while the best prediction accuracy of SVM on this gene cluster was still rather high. In this way, we could repeat the procedures and ultimately find a set of most powerful genes to the tumor. Table 4 gives the size (the number of genes) of the optimal gene cluster and the corresponding highest prediction accuracy of SVM after each round of the DSRPCL-SVM procedure.

From Table 4, we can find that the division of the optimal gene cluster would stop after several DSRPCL-SVM procedures. As we successively divided the optimal gene cluster, the prediction accuracy of SVM would drop slightly. This means that some informative genes were left, while the remaining genes became more powerful. As the division of the DSRPCL-SVM procedure on the optimal gene cluster finally stopped, we obtained the most powerful informative genes to the tumor in this indivisible cluster. Obviously, these genes are keys to the diagnosis and treatment of the tumor. Table 5 is a list of the identity numbers of the powerful genes obtained from the experiments on the three datasets, respectively. Although we do not know the biological meanings of these genes, we are sure that they are critical to the corresponding tumors for the medical diagnosis and treatment.

From the further experiments, we can find that the DSRPCL-SVM procedure can always find a set of several powerful genes to a tumor or some biological phenotypes. Clearly, this result is very significant for the medical analysis and treatment. However, the experiment result is not very stable. That is, the powerful genes may be changed greatly with the different initial values of the parameters in the DSRPCL algorithm. The experiment results given above are some typical examples. We think that the reasons of the instability or sensitivity to the initial parameter values may be two-fold. First, the DSRPCL algorithm may be sensitive with the initial values of the parameters when there are just a small number of samples in the dataset. Second, the real powerful genes for a tumor my be dependent and our DSRPCL-SVM method can only find a set of powerful genes on which an SVM diagnosis system can still be made efficiently.

Table 3 Experimental results of the second DSRPCL-SVM procedure on three datasets

Dataset	Gene cluster	No. of genes	Classfication accuracy		
			Linear SVM	Poly SVM	RBF SVM
Colon cancer	1	157	0.5556	0.0556	0.8333
	2 (optimal)	145	0.8889	0.7222	0.8889
	3	37	0.6111	0	0.8333
	4	96	0.9444	0.2222	0.8333
Leukemia	1 (optimal)	959	0.9545	0.9545	0.9545
	2	747	0.9091	0.8636	0.9091
	3	1	0.3636	0.3182	0.3182
	4	1	0.8636	0.7727	0.7727
Breast cancer	1	70	1	0.9259	0.5185
	2	152	1	1	0.6296
	3	4	0.5926	0.5185	0.5185
	4	93	1	1	0.6296
	5	187	1	1	0.5926
	6	82	1	1	0.5185
	7 (optimal)	632	1	1	0.8148
	8	62	1	1	0.5185
	9	298	1	1	0.6296

Table 4 Experimental results of the successive DSRPCL-SVM procedures on three datasets

Dataset	Subdivision	Highest accuracy	Size of optimal gene cluster
Colon cancer	1	1	435
	2	0.8889	145
	3	0.7778	61
	4	0.8333	24
	5	0.7222	6
	6	0.7222	6
Leukemia	1	0.9545	1,708
	2	0.9545	959
	3	0.9545	479
	4	0.9545	479
	5	0.9545	271
	6	0.9091	104
	7	0.9091	31
	8	0.9091	31
	9	0.8636	5
	10	0.8636	5
Breast cancer	1	1	1,580
	2	1	632
	3	1	94
	4	0.8519	25
	5	0.7037	11
	6	0.7037	11
	7	0.6296	4
	8	0.6296	4

Table 5 Identity numbers of the powerful genes for the three datasets

Dataset	Powerful gene ID No.
Colon cancer	211, 1215, 1394, 1621, 1858, 1865
Leukemia	331, 569, 787, 2281, 4586
Breast cancer	383, 385, 5294, 5797

Conclusion

We investigated the problem of informative gene discovery and analysis from the perspective of the newly established unsupervised clustering method—DSRPCL algorithm. Since the DSRPCL algorithm can detect the number of clusters automatically in a dataset, we apply it to dividing the genes expressed through microarray data into a number of functional gene clusters and use SVM to check which cluster is the set of informative genes to the tumor. Moreover, this DSRPCL-SVM procedure can be further implemented on the informative gene cluster successively and find out the critical or powerful informative genes of the tumor. Our experiments on the colon, leukemia, and breast cancer datasets demonstrated that this DSRPCL-SVM method is really efficient for discovering the informative gene set as well as the powerful informative genes for a tumor through the microarray data.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (Grant No. 60471054) and the President Foundation of Peking University.

Authors' contributions

WX and ZC participated in the method design, performed the experimental work and drafted the manuscript. JM conceived the idea of using this approach, guided the research and completed the manuscript. All authors read and approved the final manuscript.

Competing interests

The authors have declared that no competing interests exist.

References

1. Thieffry, D. 1999. From global expression data to gene networks. *Bioessays* 21: 895-899.
2. Holloway, A.J., *et al.* 2002. Options available—from start to finish—for obtaining data from DNA micoarrays II. *Nat. Genet.* 32: 481-499.
3. Golub, T.R., *et al.* 1999. Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science* 286: 531-537.
4. Ding, C. 2002. Analysis of gene expression profiles: class discovery and leaf ordering. In *Proceedings of the 6th Annual International Conference on Computational Molecular Biology*, pp.601-680. Washington, D.C., USA.
5. Deng, L., *et al.* 2005. Rank sum method for related gene selection and its application to tumor diagnosis. *Chin. Sci. Bull.* 49: 1652-1657.
6. Luo, J. and Ma, J. 2005. A multi-population χ^2 test approach to informative gene selection. *Lect. Notes Comput. Sci.* 3578: 406-413.
7. Dudoit, S., *et al.* 2002. Comparison of discrimination methods for the classification of tumors using gene expression data. *J. Am. Stat. Assoc.* 97: 77-87.
8. Ge, F. and Ma, J. 2005. An information criterion for informative gene selection. *Lect. Notes Comput. Sci.* 3498: 703-708.
9. Wang, H.Q. and Huang, D.S. 2005. A gene selection algorithm based on the gene regulation probability using maximal likelihood estimation. *Biotechnol. Lett.* 27: 597-603.
10. Wang, H.Q. and Huang, D.S. 2006. Regulation probability method for gene selection. *Pattern Recognit. Lett.* 27: 116-122.
11. Alon, U., *et al.* 1999. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc. Natl. Acad. Sci. USA* 96: 6745-6750.
12. Eisen, M.B., *et al.* 1998. Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA* 95: 14863-14868.
13. Tavazoie, S., *et al.* 1999. Systematic determination of genetic network architecture. *Nat. Genet.* 22: 281-285.
14. Xu, L., *et al.* 1992. Unsupervised and supervised classification by rival penalized competitive learning. In *Proceedings of the 11th International Conference on Pattern Recognition*, Vol.1, pp. 672-675. Hague, the Netherlands.
15. Xu, L., *et al.* 1992. Rival penalized competitive learning for clustering analysis, RBF net, and curve detection. *IEEE Trans. Neural Netw.* 4: 636-649.
16. Ma, J. and Wang, T. 2006. A cost-function approach to rival penalized competitive learning (RPCL). *IEEE Trans. Syst. Man Cybern. B Cybern.* 36: 722-737.
17. Nair, T.M., *et al.* 2003. Rival penalized competitive learning (RPCL): a topology-determining algorithm for analyzing gene expression data. *Comput. Biol. Chem.* 27: 565-574.
18. Wang, L. and Ma, J. 2007. Informative gene set selection via distance sensitive rival penalized competitive learning and redundancy analysis. *Lect. Notes Comput. Sci.* 4491: 1227-1236.
19. Wang, L. and Ma, J. 2005. A post-filtering gene selection algorithm based on redundancy and multi-gene analysis. *Int. J. Inf. Technol.* 11: 36-44.
20. Pei, S. and Huang, D.S. 2006. Cooperative competition clustering for gene selection. *J. Cluster Sci.* 17: 637-651.
21. Vapnik, V.N. 1998. *Statistical Learning Theory.* Wiley, New York, USA.
22. Guyon, I., *et al.* 2002. Gene selection for cancer classification using support vector machines. *Mach. Learn.* 46: 389-422.
23. Zhang, H.H. *et al.* 2006. Gene selection using support vector machines with non-convex penalty. *Bioinformatics* 22: 88-95.
24. 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.
25. Mukherjee, S., *et al.* 1999. Support vector machine classification of microarray data. Technical Report AI Memo 1677, MIT. Cambridge, USA.
26. Furey, T.S., *et al.* 2000. Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics* 16: 909-914.