



Title	A cooperative feature gene extraction algorithm that combines classification and clustering
Author(s)	Chow, CK; Zhu, H; Lacy, J; Lingen, MW; Kuo, WP; Chan, K
Citation	The 2009 IEEE International Conference on Bioinformatics and Biomedicine Workshops (BIBMW 2009), Washington, DC., 1-4 November 2009. In Conference Proceedings, 2009, p. 197-202
Issued Date	2009
URL	http://hdl.handle.net/10722/196713
Rights	Creative Commons: Attribution 3.0 Hong Kong License

A Cooperative Feature Gene Extraction Algorithm that Combines Classification and Clustering

Chi Kin Chow, Hailong Zhu, Jessica Lacy, Mark W. Lingen, Winston Patrick Kuo, Keith Chan

Abstract—In feature gene selection, filtering model concerns classification accuracy while ignoring gene redundancy problem. On the other hand, gene clustering finds correlated genes without considering their predictive abilities. It is valuable to enhance their performances by the help of each other. We report a new feature gene extraction algorithm, namely Double-thresholding Extraction of Feature Gene (DEFG), that combines gene filtering and gene clustering. It firstly pre-select feature gene set from the original dataset. A modified gene clustering is then applied to refine this set. In the gene clustering, specific designs are employed to balance the predictive abilities and the redundancies of the extracted feature gene. We have tested DEFG on a microarray dataset and compared its performance with that of two benchmark algorithms. The experimental results show that DEFG is superior to them in terms of internal validation accuracy and external validation accuracy. Also, DEFG can generalize the pattern structure by a small number of training samples.

Index Terms—Feature gene, extraction, classification, clustering

I. INTRODUCTION

GENE expression data commonly involve thousands of genes at, tens or hundreds of samples. The objective of feature gene extraction is to select minimum number of genes for which the performance of the classifier build from these genes is maximized. Filtering model and gene clustering are two main branches of the extraction.

Filtering model concerns minimizing the number of predictive genes at the same time the performance of the corresponding classifier is maximized. In [2], a gene is regarded as feature if its p-value is lower than a pre-determined cutoff value. Guyon et al. in [3] defined the

relevance of gene as its contribution to the cost function in Support Vector Machine (SVM). The corresponding gene ranking scheme names *Recursive Feature Elimination* (RFE). Several modifications on RFE, such as SQRT-RFE and Entropy-based RFE [10], were proposed to speed up the rank list construction process.

It may happen that some of the extracted genes are highly correlated, to which their level changes against pattern class are virtually equivalent throughout dataset. On the other hand, classifier predicts pattern class from gene level change. When two genes are with the same level change, the performance of the classifier has not any change if one of them is missing. Thus, for the case that q feature genes are highly correlated, $q-1$ of them are regarded as redundant, which we refer to as gene redundancy problem in gene filtering. Even if this group of genes has the largest relevancies, it is preferred to consider only one of them as feature gene and preserve the remaining $q-1$ positions for the less relevant genes, in order to enhance the generalization ability of the classifier.

Clustering [4]-[9] is found useful for discovering groups of correlated genes potentially co-regulated or associated to the disease or conditions under investigation. In other words, it removes redundant genes by which redundancy is in terms of gene similarity. Though their convincing performance on redundant genes reduction, very often a significant number of genes in an expression profile do not play any role in the disease or perturbed conditions under investigation. Forcing all these genes into cluster formation causes false positive and distort the structure of identified clusters.

Note that gene filtering extracts predictive genes for classification but does not concern gene redundancy problem. On the hand, gene clustering considers the redundancy but ignores genes' predictive abilities. It is obvious that ones' advantage can mutually overcome another's drawback; and it is worth to enhance feature gene extraction by the cooperation between them. Recently, Cai et al. [15] reported an extraction algorithm that couples existing classifier and clustering method. It uses gene clustering method to prune the dimensionality of the gene expression space. Afterwards, the patterns in the pruned expression space are fed to construct classifier. Mitra and Majumder [11] used linear dependency amongst genes as a feature

Manuscript received August 10, 2009. This work was supported in part by the Hong Kong Polytechnic University under Grant 1-BB56 and A-PDOE.

Chi Kin Chow (richow@inet.polyu.edu.hk), Hailong Zhu (852-34003520; fax: 852-27640011; e-mail: rihlzhu@inet.polyu.edu.hk), and Keith Chan (cskcchan@inet.polyu.edu.hk) are with the Hong Kong Polytechnic University, Hung Hom, Hong Kong

Jessica Lacy (jlacy@fas.harvard.edu) and Winston Patrick Kuo (email: wkuo@rics.bwh.harvard.edu) are with Laboratory for Innovative Translational Technologies, Harvard Medical School, Boston, MA

Mark W. Lingen (Mark.Lingen@uchospitals.edu) is with Departments of Pathology, The University of Chicago, Chicago, Illinois

similarity measure for gene clustering and feature selection. Jörnsten and Yu [12] proposed a Minimum Description Length model selection criterion for simultaneous clustering genes and subset selection of gene clusters for sample classification. Chandra et al. [13] presented a two-way clustering technique to simultaneously cluster genes and samples. The method of Kianmehr et al. [14] selects feature genes by the means of integrating fuzzy class association rules and SVM.

In this paper, we proposed a novel cooperative feature gene extraction algorithm, namely *Double-thresholding Extrcation of Feature Gene* (DEFG). It uses thresholding to select certain amount of mostly relevant genes from the original dataset. The corresponding genes are denoted as pre-selected feature genes. We then use them to pre-train a linear SVM; and the corresponding weights are used to quantify the predictive abilities of the pre-selected feature genes. Afterwards, a weighted K-mean clustering method that treats the predictive ability as data weights is employed. Rather than directly regarding the cluster centers as the refined feature genes, the objective of the clustering method in DEFG is to suggest possible refined feature gene set. Given that the clustering method generates N possible refined feature gene sets, DEFG chooses the one of which the corresponding classifier build from it has maximum error margin.

Since of the mentioned advantages of the cooperative approach, it is expected that DEFG is superior to the filtering models [2][1][3][10] in the sense of that the redundancies of the feature genes extracted by DEFG is lower. DEFG is also in contrast to [11]-[14] by which they represent the performance of a classifier as either training accuracy, leave-one-out cross validation accuracy or K-fold validation. Alternatively, DEFG selects a feature gene set of which the error margin (i.e. a representative measure of the generalization ability) of the corresponding classifier is maximized.

DEFG also differs from [11][15] in the sense of that the gene clustering in DEFG considers the predictive abilities of the genes. On the other hand, the methods of [11][15] clusters gene purely according to their similarities. The corresponding cluster centers may be interfered by the irrelevant genes.

The rest of this paper is organized as follows: section II presents the details of DEFG method. Section III reports the experiment setup. Section IV presents the experimental results and section V gives a conclusion.

II. METHOD

A. Double-thresholding Extraction of Feature Gene

In this section, we report a novel feature gene extraction scheme, called *Double-thresholding Extraction of Feature Gene* (DEFG). DEFG uses thresholding to pre-select feature genes whilst the second threshold controls the amount of pre-

selected feature genes to be filtered out by gene clustering. Thus, DEFG takes the advantage of gene expression classification and clustering to extract and refine feature genes. The resultant feature genes are less correlated and dominate classification accuracy.

In DEFG, a conventional gene filtering spots a pool of potential feature genes $S_I = \{x_j\}$. Afterwards, a feature gene refinement by the cooperation between clustering and classification is performed. During the refinement, gene clustering method generates a collection of possible refined feature gene sets according to the similarities and the predictive abilities of the genes in S_I . The refined gene set with optimal classification quantity is then selected as the refined feature gene set. Note that most of existing feature selection algorithms [2][1][3][10]-[15] presume the classification quantity as training accuracy or leave-one-out cross validation. However, since the number of training patterns related to the pattern dimension is small, training accuracy is thus not a representative performance measure. Alternatively, validation accuracy is more reliable. Though validation accuracy is never known in the training process, it relates to the generalization of a classifier whilst the generalization of a classifier is commonly measured from its error margin. It is reasonable to hypothesize that the validation accuracy is proportional to the width of error margin. Thus, the refined feature gene set selection criterion is error margin based.

Given a gene expression pattern set $S = \{\mathbf{x}_i(g)_{g \in [1, n]} \in \mathfrak{R}^n \mid y_i \in \{-1, 1\}\}$ and two thresholds τ_1 and τ_2 , DEFG starts from rank the genes according to their relevancies. In this paper, the relevancy is in terms of p -value in t-test. We pre-select the first τ_1 genes with the smallest p -values to form the pre-selected feature gene set Z . Note that other gene ranking scheme, such as Recursive Feature Elimination (RFE) **Error! Reference source not found.**, is also available. Afterwards, the pattern set S is truncated as the subset $S_c = \{\mathbf{x}_i(g)_{g \in Z} \mid y_i\}$. We pre-train a linear SVM on S_c , for which the weight vector $\mathbf{w} = [w_1, w_2, \dots, w_{\tau_1}]$ of the SVM represent the predictive abilities of the genes in Z , i.e. $|w_i|$ is as the predictive ability of the i^{th} pre-selected feature gene in Z . We then use a modified K-mean clustering method, namely *SVM Weighting K-mean Clustering*, is employed for the gene clustering. The details of the clustering algorithm will be reported in the next section. The clustering method suggests N possible refined feature gene sets $\{K_b\}_{b \in N}$ where $K_b = \{k_{b,j}\}_{j \in [1, \tau_2]} \subset Z$ consists of τ_2 refined feature genes. For each K_b , we define a pattern subset $S_b = \{\mathbf{x}_i(g)_{g \in K_b} \mid y_i\}$ and use it to train a linear SVM. The performance of K_b is represented by the error margin ε_b of the SVM. The feature gene set K_a is said as the refined feature gene set of S if its error margin ε_a is the largest amongst $\{\varepsilon_b\}$.

Algorithm A1 summarizes the procedure of DEFG.

Algorithm A1: DEFG

Input: 1) data set $S = \{[\mathbf{x}_i(g)_{g \in [1,d]} \in \mathfrak{R}^n \mid y_i \in \{-1,1\}]\}$ and 2) threshold τ_1 , 3) threshold τ_2 and 4) the number of clustering trials N .

/* Feature Gene Pre-selection: BEGIN */

1. Compute the relevance r_i of the i^{th} gene:

$$r_i = \Omega(\{x_{k,i}\}_{k \in C_-}, \{x_{k,i}\}_{k \in C_+})$$

where $\Omega(A, B)$ is the p-value of two point sets A and B . C_- contains the indices of all normal-class patterns in S and C_+ contains the indices of all cancer-class patterns in S .

2. Define the gene ranking list $L = \{\phi_j\}_{j=1,2,\dots,d}$ where the relevance of the ϕ_a^{th} gene is larger or equals to that of the ϕ_b^{th} gene, i.e. $r_{\phi_a} \geq r_{\phi_b}$ for all $a < b$.

3. Select the first τ_1 mostly relevant genes as the feature genes. The set of considered genes for classification is reduced as $Z = \{\phi_j\}_{j \in [1,\tau_1]}$.

/* Feature Gene Pre-selection: END */

/* Feature Gene Refinement: BEGIN */

4. Define the pattern subset $S_c = \{[\mathbf{x}_i(g)_{g \in Z} \mid y_i]\} \subset S$.

5. Pre-training a linear SVM on S_c and denote the weight vector of the SVM as $\mathbf{w} = [w_1, w_2, \dots, w_{\tau_1}]$.

6. Repeat the τ_2 -cluster SVM weighting K-mean clustering to collect N possible refined feature gene sets $\{K_b = \{k_{b,j}\}_{j \in [1,\tau_2]} \subset Z\}_{b \in N}$.

7. Compute the error margin ϵ_b of the SVM trained on the pattern subset $\{[\mathbf{x}_i(g)_{g \in K_b} \mid y_i]\}$

/* Feature Gene Refinement: END */

8. Define the refined feature genes as K_a where $a = \arg \max_{b \in [1,N]} \epsilon_b$.

Output: The refined feature gene set K_a of S .

B. SVM Weighting K-mean Clustering

Conventional K-mean clustering **Error! Reference source not found.** assumes equal importance of data point, to which cluster centroid is updated as the average of all data points in the same cluster. On the other hand, the elements of the weight vector \mathbf{w} in SVM represent the predictive abilities of the gene, which are expected to be distinct. Thus, the assumption of equal data importance in the conventional K-mean clustering method is not practical in feature gene extraction. In this section, we propose a predictive ability based K-mean clustering method, namely, *SVM Weighting K-mean Clustering*.

Given that $Y = \{[\mathbf{x}_i(g) \in \mathfrak{R}^n]\}$ is a pattern set, $\mathbf{w} = [w_1, w_2, \dots, w_n]$ is the weight vector of the SVM trained on Y and K is the number of clusters, the SVM weighting K-mean clustering starts from initializing the cluster centroid set $C = \{\mathbf{k}_j\}$ for $j \in [1, \tau_2]$. During the initialization, each cluster centroid \mathbf{k}_j is randomly assigned to be one of data points in Y subjects to that every centroid is distinct, i.e. $\mathbf{k}_a \neq \mathbf{k}_b$ for all $a \neq b$. Afterwards, the data points in Y are clustering according

to the sum of absolute difference from the cluster centroids. A data point $\mathbf{x} \in Y$ belongs to the cluster \mathbf{k}_j if the sum of absolute difference between them is minimal amongst all cluster centroids, i.e. $j = \arg \min_{a \in [1,K]} \left(\sum_{b=1}^n |x_b - k_{a,b}| \right)$. The cluster

centroid \mathbf{k}_j is then recomputed as the weighted sum of its data point set $Y_j = \{\mathbf{x}_b\} \subset Y$:

$$\left(\sum_b w_b \mathbf{x}_b \right) \left(\sum_b w_b \right)^{-1} \quad (1)$$

The procedure is repeated from re-clustering all data points in Y until there is no further change in the assignment of the data points to new cluster centroids. Note that SVM weighting K-mean clustering searches for the K mostly distinct feature genes from Y . On the other hand, the eq. (1) cannot guarantee that the centroid is an element in Y . Thus, all resultant centroids $\{\mathbf{k}_j\}_{j \in [1,K]}$ have to be enforced as:

$$\mathbf{k}_j \leftarrow \mathbf{x}_u = \arg \min_{a \in Y_j} \left(\sum_{b=1}^n |a_b - k_{j,b}| \right)$$

These enforced centroids are regarded as the refined feature genes by the SVM weighting K-mean clustering are $\{\mathbf{k}_j\}_{j \in [1,K]}$. *Algorithm A2* summarizes the procedure of SVM weighting K-mean clustering.

Algorithm A2: SVM Weighting K-mean Clustering

Input: 1) Gene pattern set $Y = \{[\mathbf{x}_i \in \mathfrak{R}^n]\}$, weight vector $\mathbf{w} = [w_1, w_2, \dots, w_n]$ of the SVM and 2) the number of clusters τ_2

1. Initialize the cluster centroid set $C = \{\mathbf{k}_j\}$ for $j \in [1, \tau_2]$ where \mathbf{k}_j is randomly selected from S subject to $\mathbf{k}_i \neq \mathbf{k}_j$ for all $i \neq j$.

2. Assign each data point to nearest cluster center. The data point $\mathbf{x} \in Y$ belongs to the j^{th} cluster if $j = \arg \min_{a \in [1,K]} \left(\sum_{b=1}^n |x_b - k_{a,b}| \right)$

3. Re-compute the cluster centroids: suppose all data points in the subset $Y_j = \{\mathbf{x}_b\} \subset Y$ belongs to the j^{th} cluster, \mathbf{k}_j is recomputed as $\left(\sum_b w_b \mathbf{x}_b \right) \left(\sum_b w_b \right)^{-1}$.

4. Repeat from step 1 until there is no further change in the assignment of the data points to new cluster centroids.

5. Enforce cluster centroids to the nearest data points: suppose all data points in the subset $Y_j = \{\mathbf{x}_b\} \subset Y$ belongs to the j^{th} cluster, \mathbf{k}_j is enforced as $\arg \min_{a \in Y_j} \left(\sum_{b=1}^n |a_b - k_{j,b}| \right)$.

Output: a possible refined feature gene set C .

III. EXPERIMENT SETUP

A. Dataset

Oral cancer multiple datasets: We have available four microarray datasets; the first was measured with HG-U133 Plus2 and it has 11 normal and 50 cancerous samples, the second is from a HG-U133A and it has 22 normal and 22 cancerous samples, the third set comes from a HG-Focus and has only 22 cancerous samples and the fourth has 12 normal and 26 cancerous samples and measured also with HG-U133 Plus2. All the chips are manufactured by Affymetrix (Santa Clara, CA).

B. Algorithms for Comparison

To evaluate the impact of DEFG, we compare its performance with two algorithms. The designs and settings of DEFG and the algorithms for comparison are summarized below.

Test algorithm 1 – SVM with t-test based feature gene extraction (SVM-ttt): In SVM-ttt **Error! Reference source not found.**, the relevance of a gene is measured on its p -value in t -test. After computing the p -values of the genes, the f_0 mostly relevant genes are selected as feature genes, and f_0 is pre-determined by user.

Test algorithm 2 - SVM with Recursive Feature Elimination (SVM-rfe): The gene relevance list is computed according to *recursive feature elimination* (RFE) **Error! Reference source not found.** At each iteration, RFE figures out and removes the least contributed gene from a set of considered genes. The iteration is repeated until all genes are removed from the set. The relevance of a gene is represented as the iteration index which it is removed. The curve representing the cross-validation error versus the number of mostly relevant features f_0 is fitted by an exponential function $g(f_0)$. The optimal number of feature genes is obtained as the value to which the change of $g(f_0)$ is just smaller than threshold.

Test algorithm 3 – Double-thresholding Extraction of Feature Gene with p-value gene ranking scheme (DEFG): DEFG is a feature gene extraction algorithm that combines gene filtering and gene clustering. It pre-selects feature genes according to their p -value in t -test. A predictive ability based K-mean clustering method is applied to refine these genes.

For DEFG, the threshold τ_1 in feature gene pre-selection is chosen to be 60. The value of τ_2 in DEFG varies according to the experiment setting. The values of f_0 in SVM-ttt and SVM-rfe are chosen to the same of DEFG.

C. Experiment Settings

The first three datasets form a superset T . Suppose n_+ is the number of normal-class patterns and n_- is number of cancer-class patterns in T , and r is sampling rate, we randomly pick m_+ normal-class patterns and m_- cancer-class patterns from T

to form the training set. In this paper, test algorithms are examined on two groups of validation sets:

Internal validation set – It is defined as all patterns in T except for those appeared in the training set. The corresponding accuracy, namely *internal validation accuracy* (IVA) represents the generalization ability of a test algorithm on the training set.

External validation set – The fourth dataset is defined as the external validation set. The corresponding accuracy, namely *external validation accuracy* (EVA) represents the generalization ability of a test algorithm on the oral cancer classification problem.

The values of τ_2 and f_0 vary from 3 to 30. The sampling rate is chosen to be 0.5.

To provide a fair and repeatable comparison amongst the test algorithms, the performance of each test algorithm on a particular simulation is evaluated based on statistics obtained from 100 independent runs. The patterns in the training set are independently and randomly re-picked for each run. All test algorithms are implemented in MATLAB language.

IV. EXPERIMENTAL RESULTS

Figure 1 shows the IVA of the test algorithms against τ_2/f_0 . Figure 2 shows the EVA of the test algorithms against τ_2/f_0 . In each of the figures, the results of DEFG, SVM-ttt and SVM-rfe are with the indicators 'O', ' Δ ' and '*' respectively.

Seen from the figures, DEFG performs better than both SVM-ttt and SVM-rfe for all trialed τ_2 except for $\tau_2 = 3$. The IVA differences between DEFG and SVM-ttt range from 1.14% to 5.05%, whilst the EVA differences between DEFG and SVM-rfe ranges from 0.14% to 1.28%. While considering EVA, the improvement made by DEFG is much significant. The ranges of the improvement by DEFG are [2.79%, 10.55%] and [2.47%, 6.50%] by comparing with SVM-ttt and SVM-rfe respectively.

Since the training set cannot fully represent the problem structure (the general mapping from gene expression to its class), it is expected that the internal validation accuracies are always higher than the external validation accuracies at all trialed τ_2/f_0 for all test algorithms.

The performance differences between DEFG and SVM-ttt in terms of both internal and external validation accuracies are significant. Moreover, the significance of the EVA differences between DEFG and SVM-ttt is higher than that of the IVA differences between DEFG and SVM-ttt. This infers the higher generalization ability of the feature gene set obtained by DEFG by comparing with SVM-ttt.

The maximal IVA variations of DEFG, SVM-ttt and SVM-rfe are 2.8, 6.4 and 3.04 respectively, whilst the maximal EVA of them are 2.51, 8.23 and 4.05 respectively. As DEFG has the smallest variation in both IVA and EVA, the performance of DEFG is more stable than those of SVM-ttt and SVM-rfe.

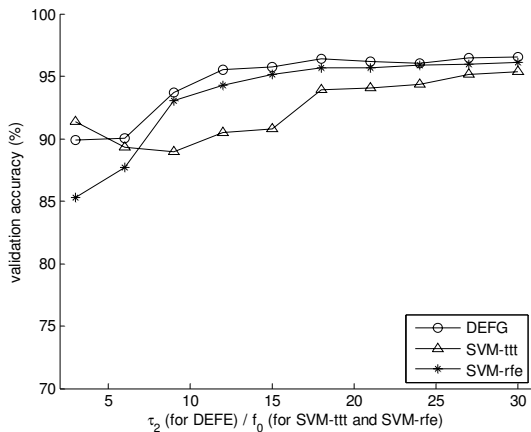


Fig. 1. Internal validation accuracy (%) against τ_2/f_0 .

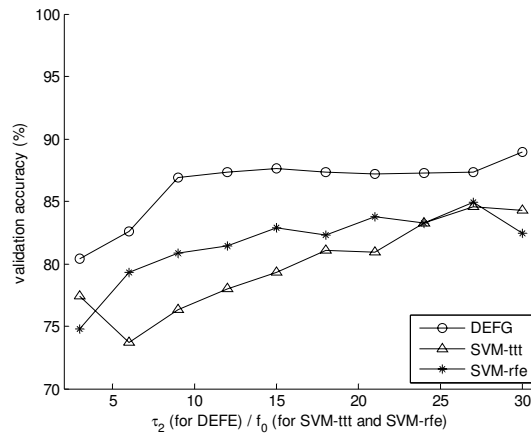


Fig. 2. External validation accuracy (%) against τ_2/f_0 .

V. CONCLUSIONS

Gene filtering extracts representative genes for classification but their redundancies are not taken into account. On the hand, gene clustering considers redundancy but does not concern their predictive abilities. Clearly, it is worth to extract feature genes by the cooperation between them, to which the genes have high predictive abilities and low redundancies.

This paper proposes a feature extraction algorithm that combines two branches of pattern analysis: classification and clustering. It uses gene ranking list to pre-select feature genes. Afterwards, a modified gene clustering method is performed to generate a collection of possible refined feature gene sets. An error margin based criterion is employed to choose the resultant feature gene set. Two thresholds are required for the feature gene pre-selection and gene clustering, in which the proposed algorithm names *Double-thresholding Extraction of Feature Genes* (DEFG).

Comparing to existing feature extraction algorithms **Error! Reference source not found.**, **Error! Reference source not found.**, **Error! Reference source not found.**, **Error! Reference source not found.**, DEFG has three distinct characteristics:

DEFG balances predictive abilities and redundancies of feature genes by the proposed gene clustering method.

The components of the clustering method are classification oriented: the similarity between genes is represented by their correlation; and the update scheme of cluster center is based on the predictive abilities of the genes in a cluster.

Instead of training accuracy or leave-one-out cross validation, the performance of a feature gene set is measured on the error margin of the corresponding classifier.

These characteristics of DEFG contribute feature gene extraction in the following ways: Since the gene predictive ability is taken into account, the cluster centers are insensitive to the distribution of the irrelevant genes. Benefitting from the classification-oriented components of the gene clustering method, the feature genes extracted by DEFG have high predictive abilities. It leads to a classifier with higher generalization ability as the feature gene set is targeted on maximizing the error margin.

The proposed algorithm is compared with SVM-ttt and SVM-rfe. They are examined on a real microarray dataset. The experimental results show that:

DEFG is superior to SVM-ttt whilst DEFG-rfe is superior to SVM-rfe in terms of internal validation accuracy and external validation accuracy. This illustrates the contribution of DEFG on refining feature gene set. DEFG can generalize the pattern structure from a smaller number of training samples by comparing with SVM-ttt and SVM-rfe.

REFERENCES

- [1] E. B. Huerta, B. Duval and J. K. Hao, "A Hybrid GA/SVM Approach for Gene Selection and Classification of Microarray Data", *EvoWorkshops 2006*, LNCS 3907, pp. 34 – 44.
- [2] T. K. Man *et al.*, "Expression Profiles of Osteosarcoma That Can Predict Response to Chemotherapy", *Cancer Research*, vol. 65, no. 18, 2005, pp. 8142–8150.
- [3] I. Guyon, J. Weston, S. Barnhill, V. Vapnik, "Gene selection for cancer classification using support vector machines", *Machine Learning*, vol. 46, 2002, pp. 389 – 422.
- [4] M. B. Eisen *et al.*, "Cluster analysis and display of genome-wide expression patterns", In *Proc. Natl Acad. Sci. USA*, 1998, pp. 14863 – 14868.
- [5] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations", In *Proc. of the 5th Berkeley Symp. Math. Stat. Prob.*, vol. 1, 1967, pp. 281–297.
- [6] L. Kaufman, P. Rousseeuw, "*Finding Groups in Data: An Introduction to Cluster Analysis*", Wiley, New York, 1990.
- [7] T. Kohonen, "The self-organizing map", In *Proc. of IEEE*, vol. 78, 1990, pp.1464 – 1480.
- [8] K. Y. Yeung *et al.*, "Model-based clustering and data transformations for gene expression data", *Bioinformatics*, vol. 17, 2001, pp. 977 – 987.
- [9] G. C. Tseng, W. H. Wong, "Tight clustering: a resampling-based approach for identifying stable and tight patterns in data", *Biometrics*, vol. 16, 2005, pp. 10 – 16.
- [10] C. Furlanello, M. Serafini, S. Merler, G. Jurman, "Entropy-based gene ranking without selection bias for the predictive classification of microarray data", *BMC Bioinformatics*, vol. 5(54), 2003.

- [11] P. Mitra, D. D. Majumder, "Feature Selection and Gene Clustering from Gene Expression Data", In *Proc. of the 17th International Conference on Pattern Recognition 2004*, vol. 2, pp. 343 - 346.
- [12] R. Jörnsten, B. Yu, "Simultaneous gene clustering and subset selection for sample classification via MDL", *Bioinformatics*, vol. 19, no. 9, 2003, pp. 1100 - 1109.
- [13] B. Chandra, S. Shanker, S. Mishra, "A new approach: Interrelated two-way clustering of gene expression data", *Statistical Methodology*, vol. 3, 2005, pp. 93 - 102.
- [14] K. Kianmehr, M. Alshalalfa, R. Alhajj, "Fuzzy clustering-based discretization for gene expression classification", *The international Journal of Knowledge and Information System 2009*, (in press).
- [15] Z. Cai, L. Xu, Y. Shi, M. R. Salavatipor, R. Goebel, G. Lin, "Using Gene Clustering to Identify Discriminatory Genes with Higher Classification Accuracy", In *Proc. of the 6th IEEE Symposium on Bioinformatics and BioEngineering*, 2006, pp. 235 - 242