COMPREHENSIVE DATA ANALYSIS FOR BIOMARKER PATTERN DISCOVERY USING DNA/PROTEIN MICROARRAYS

by

YOUNG BUN KIM

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

August 2008

I delicate this dissertation to my family, especially...

to Jung Hun for all his continued love and support;

to Yuna for her love and patience;

to Mom and the families of my two sisters, Myung Hee and Myung Ah for their prayers and endless supports;

to the family of Jung Hun for encouragement all these years.

ACKNOWLEDGEMENTS

I would like to express my warmest gratitude to my supervising professor Dr. Jean Gao, who introduced me to the area of bioinformatics. I have been amazingly fortunate to have an advisor who gave me the freedom to explore on my own and at the same time the guidance to reach my goal. Her patience and support helped me overcome many critical situations and finish this dissertation.

I would like to thank Dr. Pawel Michalak for his constant encouragement throughout my academic years and carefully reading and commenting my manuscripts. I would also like to extend a sincere thanks to Dr. Nikola Stojanovic who has been very generous and encouraging. I would like to thank to Dr. Leonidas Fegaras and Dr. Roger Walker for serving my Ph.D. committees and for contributing their time and efforts.

Most importantly, I wish to extend my warmest thanks to my mother who has encouraged me over this long journey. I deeply appreciate all of her love and support she has given me along the way. Finally, to my husband, Jung Hun and daughter, Yuna, they have been a constant source of love, support, and encouragement all these years. I could not have done it without them. A special thanks to my husband's family and my systers for their endless supports.

June 24, 2008

ABSTRACT

COMPREHENSIVE DATA ANALYSIS FOR BIOMARKER PATTERN DISCOVERY USING DNA/PROTEIN MICROARRAYS

Young Bun Kim, Ph.D.

The University of Texas at Arlington, 2008

Supervising Professor: Jean Gao

During the last decade, the advent of microarray technology has stimulated rapid research advances in bioinformatics. Microarray data pose great challenges for computational data analysis, because of their large dimensionality (up to several tens of thousands of genes) and their small sample sizes. In order to deal with these particular characteristics of microarray data, the need and importance for feature selection techniques were realized. While a lot of research deals with classification methods and their application to microarray data, only a few approaches are explicitly designed to consider interaction among the investigated features. It is well known that the interactions between genes or proteins are important for many biological functions, i.e. signals from the outside of a cell are mediated to the core of the cell by protein-protein interactions of the signaling molecules. Hence, to achieve optimal classification accuracy, these interactions among features need to be taken into account. My research goal is to develop algorithms which not only effectively select the most informative features but also identify the relationship among those features.

For the clustering of the genes, researchers have attempted to apply feature subset selection to select a subset of genes that are common for all possible un-known classes. However, the fact that a certain set of genes may be only related to a subset of experiments due to experiment design and no enough knowledge on gene function is overlooked. In the thesis, a new subspace semi-supervised clustering algorithm called EPSCMIX (Emerging Pattern Subspace Clustering by MIXure models) is designed. This algorithm is used to find gene expression patterns which in turn could be used to predict pathological phenotypes and identify genes that might anticipate the clinical behavior of diseases. Our method is based on feature saliency measure, the probability of feature relevance, which is estimated by an Expectation Maximization (EM) algorithm. This approach employs Emerging Patterns (EPs) to identify effectively relationships among genes. The best number of classes and the relevant set of genes are discovered by EPSCMIX.

To address the problem of identifying informative genes from a large amount of gene expression data when no prior knowledge is available, we develop a hybrid methodology for unsupervised gene (feature) selection and sample clustering. The algorithm, PFSBEM (hybrid PCA based Feature Selection and Boost-Expectation-Maximization clustering), introduces a new PCA (principal component analysis) based feature selection within a wrapper framework. PFSBEM uses a three-step approach to feature selection and data clustering. The first step initially reduces high-dimension feature space by retrieving feature subsets with original physical meaning based on their capacities to reproduce sample projections on PCs (principal components). Each feature subset corresponds to a certain PC. The second step then determines the important PCs that contribute to data clustering. A boost-EM (expectation-maximization) clustering method is developed to achieve stable

data grouping. Finally, from the merged feature subsets of important PCs, the best feature subset that maximizes data clustering is selected.

Feature pattern (combination of features) identification techniques could be used to capture more underlying semantics than single feature. However, it is very hard to find meaningful patterns in large datasets like microarray data because of the huge search space. Furthermore, infrequent patterns are often irrelevant or do not improve the accuracy of the classification. To tackle these problems, we finally design a discriminative feature patterns identification system named DFPIS. Instead of simply identifying genes contributing to the network, this methodology takes into consideration of gene interactions which are represented as Strong Jumping Emerging Patterns (SJEP). Furthermore, infrequent patterns though occurred are considered irrelevant. The whole framework consists of three steps: feature (gene, protein) selection, feature pattern identification, and pattern annotation.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

IDENTIFYING MARKER GENES USING SEMI-SUPERVISED SUBSPACE CLUSTERING

1.1 Introduction

Clustering analysis of microarray gene expression data has been increasingly used in cancer research for discovering and validating various cancer classes. Traditional clustering algorithms tend to consider all the features of an input dataset. However, for high dimensional data such as gene expression microarray data, many of the features are often irrelevant or do not contribute to the clustering. Moreover, for distance metrics based clustering, such analysis becomes meaningless as the number of features dimensionality goes up. Alternatively, researchers have attempted to apply feature subset selection to select a subset of features which are common for all possible un-known classes. While clustering algorithms integrating feature selection search the whole dataset, subspace clustering algorithms localize the search for relevant features [2]. Subspace clustering algorithms are one solution to the analysis of microarray expression data. Another deficiency of the most clustering approaches is that they do not take notice of the correlation of the genes even though it is well-known fact that (co-)expression of genes in a cell is based on regulatory control. Emerging Patterns (EPs) are one method that can be used to find the mutual relationships of genes; see, for example, [3], [4], [5] and [6], among others.

In this chapter, we introduce a new subspace clustering algorithm called EPSCMIX (Emerging Pattern Subspace Clustering by MIXture models), which can be applied to prediction of pathological features of diseases using microarray ex-

pression data. This is based on a feature saliency measure that is obtained by the EPs algorithm and the EM (expectation maximization) algorithm, which is a sound mathematics-based approach that does not involve any explicit search. [7] defined the concept of feature saliency as the probability that the feature is relevant and introduced an EM algorithm to estimate it in the context of mixture-based clustering, which can simultaneously perform feature selection and clustering. Our approach extends this algorithm to support two other things : the use of interrelation of genes in classification and a subspace clustering. For the first consideration, the EPSCMIX approach employs the EPs to estimate feature saliency. So, the EPSCMIX approach not only can model interactions of genes, but also find a meaningful specific set of genes related to each cluster. Another advantage of our solid theoretical EM based EPSCMIX approach is the avoiding of a combinatory feature search like most subspace clustering algorithms do and avoiding of choice of search strategy.

1.2 Methods

1.2.1 Emerging Patterns (EPs)

Emerging Patterns (EPs) were first introduced by [8] as associations of features (conditions involving several genes in microarray data), whose supports increase significantly from one class to another. They have the special advantage of modeling interactions among genes, which builds powerful classifiers. However, it is a hard task to find short and meaningful EPs for large datasets because of the huge search space. There are several approaches to discover EPs applied to microarray data such as: border based mining algorithm [8], condensed representation [9], CART-based approach [3], etc.

Step 1: gene selection and discretization

We discretize all data sets using the entropy based discretization method of [10], not only to efficiently explore the most discriminatory features but also to earlier remove many of the noisy features. Let's suppose that there is a given set S of all samples and a partition boundary T by which S is partitioned into two subsets S_1 and S_2. Let $P(c_i, S_j)$ be the proportions of samples in subset S_j that belong to class c_i. The class entropy for subset S_j is expressed as:

$$Ent(S_j) = -\sum_{i=1}^{m} P(c_i, S_j)\mathbf{log}(P(c_i, S_j)) \tag{1.1}$$

where m is the number of classes. Assuming the subsets S_1 and S_2 are derived from partitioning a feature A at a point T, then the class information entropy of the partition is defined by:

$$E(A, T, S) = \frac{|S_1|}{|S|} Ent(S_1) + \frac{|S_2|}{|S|} Ent(S_2). \tag{1.2}$$

The cut point T_A for which $E(A, T, S)$ is minimal among all the candidate cut points is selected for a binary discretization for feature A. The same process is applied recursively to S_1 amd S_2 until the minimal description length criterion used by [10] is reached.

Note that this method needs class information. So, we applied known class labels for this and this is the reason that our algorithm is called as a semi-supervised clustering algorithm even though clustering algorithm does not need any class information. However, for datasets which have no class labels, we can apply unsupervised feature selection algorithm and unsupervised discretization method for the same purpose.

For example, the discretization method partitions genes each into two disjoint intervals on colon cancer dataset. There are two intervals such as (-∞, 59.8) and

[59.8, $+\infty$) for M26383 gene. For the convenience, we index them as the 1st and 2nd items and so on. So, the emerging pattern {2} represents {$gene_{M26383}$@[59.8,$+\infty$)}.

Step 2: generating JEPs

We employed Jumping EPs (JEPs), which are defined as the patterns that are found only in one class relative to another class and have stronger ability to discriminate different classes than any other types of EPs. For example, let's suppose one of JEPs on cancer class is {2, 3}. It represents {$gene_{M26383}$@[59.8,$+\infty$), $gene_{M63391}$@(-∞, 1700)}. And it can be interpreted as :
the pattern that the expression of M26383 is $\geq$ 59.8 and the expression of M63391 $<$ 1700 was found at least one only in cancer samples.

To reduce the computational cost to get JEPs, we use top 35 ranked genes based on information gain criterion and applied the border based algorithm in [8] and [4] which directly produce EPs without generating all candidate patterns.

Step 3: selecting the most expressive genes

Finally, we select the Most Expressive Genes (ME-genes) which are often participating in JEPs. For example, let's suppose that there are 5 JEPs such as {1, 3, 5}, {2, 6}, {4, 9}, {1, 10} and {2, 4, 10}. The 1st ME-gene is {$gene_{M26383}$@(-∞, 59.8), [59.8,$+\infty$)} because the frequency of $gene_{M26383}$ is 4 (1,2,1,2). Note that the frequency information of ME-genes is used to estimate feature saliency in our algorithm.

1.2.2 Feature Saliency

The concept of feature saliency originates from the definition of feature irrelevancy [11], [12]: the l-th feature is relevant if its distribution is dependent on the

class labels. However, for unsupervised learning, it is hard to tell when a feature is useless without class labels. So, [7] defined the feature saliency as the probability that the features are relevant to a class because it may be better to indicate how salient the feature is instead. They presented the algorithm to simultaneously estimate the feature saliencies and the number of clusters using EM algorithm. But, they also have two limitations related to interactions among genes and the subspace of clusters, the same as I have already mentioned earlier.

In this paper, we present a new method to overcome these problems, which obtains feature saliency by using both EPs algorithm and EM algorithm. This approach finds maximum likelihood estimation of feature saliency as calculating its expectation value by EPs algorithm and re-estimating it by a maximization step of EM algorithm.

1.2.3 EM clustering algorithm for Feature Saliency

The EPSCMIX approach is inspired by the extension of the EM algorithm for feature saliency [7]. To conform to the consistency of original authors, we keep the same notations.

Let $\mathcal{Y}$=$\{y_1, \ldots, y_N\}$ denote a set of D-dimensional N observations. It is assumed that each observation y_i is from an initially specified number K (all classes are assumed to have the same Gaussian mixtures, but they can be generalized to other types of mixtures) in some unknown mixing probabilities $\alpha_1, \ldots, \alpha_K$.

$$p(y) = \sum_{j=1}^{K} \alpha_j p(y|\theta_j) \tag{1.3}$$

Where for $\forall_j$, $\alpha_j \geq 0$, $\sum_j \alpha_j = 1$,θ_j is the set of unknown parameters defining the j-th component. And $\theta \equiv \{\theta_1, \ldots, \theta_K, \alpha_1, \ldots, \alpha_K\}$ will denote the full parameter sets. The goal of mixture estimation is to infer θ from a set of N data points

$\mathcal{Y}$=$\{y_1, \ldots, y_N\}$, assumed to be samples of a distribution with density given by Eq. (1.3). Under the assumption that $y_1, \ldots, y_N$ are independent observations, the maximum likelihood estimate of θ can be iterated by applying the EM algorithm (Dempster et al. 1977).

For applying feature saliency to mixture estimation method, we let $\Phi = \{\{\phi_{11}, \ldots, \phi_{1D}\} \ldots \{\phi_{K1}, \ldots, \phi_{KD}\}\}$ be a set of binary parameters, such that $\phi_{jl} = 1$ if the l-th feature is relevant to j-th cluster and otherwise,$\phi_{jl} = 0$ and consider two probability density functions (pdf) : $p(\cdot|\theta_{jl})$, which is the pdf of the l-th feature in the j-th component and $q(y_l|\lambda_l)$ which is common density (we shall limit it as Gaussian) for the irrelevant l-th feature. Then, the mixture density in Eq. (1.3) is re-written as

$$p\{y|\Phi, \{\alpha_j\}, \{\theta_{jl}\}, \{\lambda_l\}\} = \sum_{j=1}^{K} \alpha_j \prod_{l=1}^{D} [p(y_l|\theta_{jl})]^{\phi_{jl}} [q(y_l|\lambda_l)]^{1-\phi_{jl}} \tag{1.4}$$

where θ_{jl} is the set of parameters of the j-th component and λ_l is the set of parameters of l-th feature. Finally, according to previously introduced feature saliency definition as $\rho_{jl} = P(\phi_{jl} = 1)$, i.e., the probability the l-th feature is relevant to j-th component, and Eq. (1.4) can be further written as

$$\rho_{jl} = p(\phi_{jl} = 1|z_i = j) = \frac{Freq_i(y_{il} \in \{ME - genes\}_j)}{Max_l(Freq_i(y_{il} \in \{ME - genes\}_j))} \tag{1.5}$$

$$p(y|\theta) = \sum_{j=1}^{K} \alpha_j \prod_{l=1}^{D} (\rho_{jl} p(y_l|\theta_{jl}) + (1 - \rho_{jl}) q(y_l|\lambda_l) \tag{1.6}$$

where $\{ME\text{-}genes\}_j$ is the set of ME-genes for j-th component, y_{il} is l-th feature value of i-th sample, $Freq_i(y_{il})$ is the sum of the number of the sample whose interval

value is the one of ME-genes set of j-th component in the l-th feature dataset of j-th component, and $Max_l(Freq_i(y_{il}))$ is the maximum value of $Freq_i(y_{il})$ among all features in the dataset of j-th component.

Typically, EM algorithm consists of two major steps, namely, an expectation step and maximization step. The expectation step is to estimate the unknown underlying variables, using the current estimate of the parameters and conditioned upon the observations. Then, the maximization provides a new estimate of the parameters. These two steps are iterated until convergence. The two steps of the proposed EM algorithm can be derived as follows: by treating Z as hidden class labels and Φ as hidden variables :

[E-step] : Compute the following quantities.

$$a_{ijl} = P(\phi_{jl} = 1, y_{il}|z_i = j) = \rho_{jl} p(y_{il}|\theta_{jl}) \tag{1.7}$$

$$b_{ijl} = P(\phi_{jl} = 0, y_{il}|z_i = j) = (1 - \rho_{jl}) q(y_{il}|\lambda_l) \tag{1.8}$$

$$c_{ijl} = P(y_{il}|z_i = j) = a_{ijl} + b_{ijl} \tag{1.9}$$

$$w_{ij} = P(z_i = j|y_i) = \frac{\alpha_j \prod_l c_{ijl}}{\sum_j \alpha_j \prod_l c_{ijl}} \tag{1.10}$$

$$u_{ijl} = P(\phi_{jl} = 1, z_i = j|y_i) = \frac{a_{ijl}}{c_{ijl}} w_{ij} \tag{1.11}$$

$$v_{ijl} = P(\phi_{jl} = 0, z_i = j|y_i) = w_{ij} - u_{ijl} \tag{1.12}$$

The feature saliency (ρ_{jl}) used in Eqs. (1.7) and (1.8) is computed using Eq. (1.5). In these equations, the variable a_{ijl} and b_{ijl} measure how relevant the l-th gene is to the j-th component, when the k-th gene is used and is not used, respectively, and the variable c_{ijl} measures how irrelevant the l-th gene is to the j-th component. The variable w_{ij} measures how important the i-th sample is to the j-th component. The variables u_{ijl} and v_{ijl} measure how important the i-th sample is to the j-th compo-

nent when the l-th feature is used and is not used, respectively, for j-th component .

[M-step] : Re-estimate the parameters.

$$\hat{\alpha}_j = \frac{max(\sum_i w_{ij} - (RD/2)\eta, 0)}{\sum_j max(\sum_i w_{ij} - (RD/2)\eta, 0)} \tag{1.13}$$

$$Mean(\hat{\theta}_{jl}) = \frac{\sum_i u_{ijl} y_{il}}{\sum_i u_{ijl}} \tag{1.14}$$

$$Var(\hat{\theta}_{jl}) = \frac{\sum_i u_{ijl}(y_{il} - Mean(\hat{\theta}_{jl}))^2}{\sum_i u_{ijl}} \tag{1.15}$$

$$Mean(\hat{\lambda}_l) = \frac{\sum_i (\sum_j v_{ijl}) y_{il}}{\sum_{ij} v_{ijl}} \tag{1.16}$$

$$Var(\hat{\lambda}_l) = \frac{\sum_i (\sum_j v_{ijl})(y_{il} - Mean(\hat{\lambda}_l))^2}{\sum_{ij} v_{ijl}} \tag{1.17}$$

$$\hat{\rho}_{jl} = \frac{max(\sum_i u_{ijl} - R/2, 0)}{max(\sum_i u_{ijl} - R/2, 0) + max(\sum_i v_{ijl} - S/2, 0)} \tag{1.18}$$

where R and S are numbers of parameters in θ_{jl} and λ_l respectively (R=S=2 for univariate Gaussians). And η is a function that converges to 1 as iterations increase. If K and D are too large, it can happen that no component has enough initial support, and all α_j will be undetermined. To avoid this problem, we adopted η function in Eq. (1.13). And, the reason to estimate ρ_{jl} using the proportion of the term $\sum_i u_{ijl}$ is that it can be interpreted as how likely it is that ϕ_{jl} equals one.

1.2.4 Model Selection

Most of algorithms using EM algorithm for fitting mixtures with unknown numbers of components have difficulties in the following problems: EM is highly dependent on initialization, and EM may converge to the boundary of the parameter space [13]. The one solution is to use model selection criteria like MML (Minimum Message Length) : find the "Most probable" overall model, which generates the shortest description of the data, in the whole set of available models rather than

[Input] K : initial number of component,
Kmin : minimum number of component
[Output] Number of component K,
Parameters set $\{\{\theta_{jl}\},\{\alpha_j\},\{\lambda_l\}\{\rho_{jl}\}\}$

Preprocessing:
Perform an Entropy-Based Discretization using known labels
Perform Standardization with raw data of selected features

Initialization
Set feature saliencies after labeling classes by using KMeans algorithm or random assignment method (K=30)
Set the parameters of a large number of mixture components randomly
Set the common distribution to cover all data

Iterations
While K > Kmin do
    Repeat
        Perform E-step
        Perform M-step
        Assign training data class labels that are most likely generated
        Update feature saliencies based on these labels
    until $F_{(t-1)}(K,\Phi)-F_{(t)}(K,\Phi) < \varepsilon |F_{(t-1)}(K,\Phi)|$
    Save the current model parameters and its MML
    Remove the component with the smallest weight
end while.
Return the model parameters that yield the smallest message length.

Figure 1.1. The EPSCMIX algorithm.

selecting one among a set of candidate models. In this paper, we applied the approach in [7] and [13], which is a penalized maximum likelihood function based on the MML criteria (after discarding the order on term) as follows:

$$F_{(t)}(K,\Phi) = -logp(\mathcal{Y}|\theta) + \frac{1}{2}(K+KD)logN$$
$$+\frac{R}{2}\sum_{j=1}^{K}\sum_{l=1}^{D} log(N\alpha_j\rho_{jl}) + \frac{S}{2}\sum_{j=1}^{K}\sum_{l=1}^{D} log(N(1-\rho_{jl})) \qquad (1.19)$$

This criterion has easily understandable interpretations. $-logp(\mathcal{Y}|\theta)$ is the code-length of the data and $\frac{1}{2}(K+KD)logN$ is a parameter code length corresponding to K α_j values and KD ρ_{jl} values. The other terms are the code lengths required for estimating each θ_{jl} and λ_l : $\frac{R}{2}log(N\alpha_j\rho_{jl})$ and $\frac{S}{2}log(N(1-\rho_{jl}))$.

1.2.5 EPSCMIX algorithm

From a Bayesian point of view, the criterion in Eq. (1.19) corresponds with a posteriori density by applying Dirichlet-type (not exactly) priors on the α_j and ρ_{jl} :

$$p(\alpha_1,\ldots,\alpha_K) \propto \prod_{j=1}^{K} \alpha_j^{-RD/2} \tag{1.20}$$

$$p(\rho_{11},\ldots,\rho_{1D},\ldots,\rho_{K1},\ldots,\rho_{KD}) \propto \prod_{j=1}^{K}\prod_{l=1}^{D} \rho_{jl}^{-R/2}(1-\rho_{jl})^{-S/2} \tag{1.21}$$

These priors are reflected in Eqs. (1.13) and (1.18) of M-step because priors are conjugate to complete data likelihood. So, they have the function to reduce components and feature saliencies with "too weak", meaning they force some of the α_j to go to zero and some of rho_{jl} to go to zero or one. This pruning factor is the way to avoid one of the problems of standard EM algorithm. However, if K is too large, it can happen that no component has enough initial support and all α_j will be undetermined. To avoid this problem, a component-wise version of EM can be adopted. This method updates all parameters sequentially rather than simultaneously. This method also helps to avoid slow convergence situations by updating the parameters sequentially in small groups associated to small hidden data spaces rather than one

large complete data space [14], [7], [13]. We did not apply this method in this paper because η function in Eq. (1.13) can handle the former problem, and there is no slow convergence problem if some of high ranked features are considered the same as ours. This algorithm is initialized with large number of K, where K is much larger than the true number of mixture component. By doing so, this overcomes the problem, which converges to bad local minima by initialization. In other words, by starting with many components, this algorithm avoids the difficulty that EM is unable to move components across least-likelihood regions.

Our proposed algorithm is summarized in Fig. (1.1). The algorithm is composed of three phases : preprocessing, initialization, and iteration. We use two kinds of data sets, labeled discretized data and unlabeled raw data. The former is for EPs algorithm and the other is for EM algorithm. First, we make the former set as applying an entropy based discretization method, and for the other, we standardized the original data set with the method that each column is standardized to have mean zero and unit standard deviation, and then each row is standardized to have zero mean and unit standard deviation. In the initialization phase, we initialize all unknown parameters such as mixture parameters $\{\theta_{jl}\}$,$\{\alpha_j\}$, parameters of common distribution $\{\lambda_l\}$ and feature saliencies $\{\rho_{jl}\}$. In the iteration phase, two steps of EM and updating feature saliencies are iteratively performed until Eq. (1.19) is satisfied. The feature saliencies are updated using Eq. (1.3) based on training data class labels that are most likely generated by previous EM-steps. After performing E-step and M-step, if α_j becomes zero, the j-th class will be pruned, and if ρ_{jl} becomes one, $q(y_l|\lambda_l)$ will be pruned; otherwise, if ρ_{jl} becomes zero, $p(y_l|\theta_{jl})$ will be pruned for j-th component.

Table 1.1. Data description

Name	N	D	c
synthetic data	800 (200 for each of them)	10	4
colon cancer	62 (40 tumor, 22 normal)	2000	2
prostate cancer	102 (52 tumor, 50 normal)	12600	2
wine recognition	178 (59 class1, 71 class2, 48 class3)	13	3
Wdbc	569 (357 benign, 212 malignant)	30	2

1.3 Experiments

We implemented the EPSCMIX approach with the Java language and used not only the synthetic data of [7] but also the well-known datasets, such as the colon tumor set of [15] and the prostate tumor set of [16] from Kent Ridge Bio-medical Data Set Repository (http://sdmc.lit.org.sg/GEDatasets/Datasets.html), to demonstrate the robustness of our new approach. Additionally, to compare our performance with the one of Law's method, we considered two real data sets such as the wine recognition data set and the Wdbc(Wisconsin diagnostic breast cancer) data set from UCI machine learning repository (http://www.ics.uci.edu/ mlearn/MLRepository.html). Data sets are summarized in Tab. 1.1. Each data set has N data points with D features from c classes.

1.3.1 Synthetic Data

We used the synthetic data set, which consists of 800 10-dimensional samples from a mixture of four equiprobable Gaussians $\mathcal{N}(m_i, I)$, i=1,2,3,4, where

$$m_1 = \begin{pmatrix} 0 \\ 3 \end{pmatrix}, m_2 = \begin{pmatrix} 1 \\ 9 \end{pmatrix}, m_3 = \begin{pmatrix} 6 \\ 4 \end{pmatrix}, m_4 = \begin{pmatrix} 7 \\ 10 \end{pmatrix}$$

and eight noisy features sampled from a $\mathcal{N}(0, I)$. We repeated these experiments 10 times, each initialized with K=30 and the stopping threshold at 10^{-7}. Eight noisy features are removed early in the preprocessing step. In all ten runs, the four components were exactly identified, and the saliencies of two features for four classes are always one. This result shows that the proposed algorithm finds the true clusters successfully and assigns appropriate salience value to the feature.

1.3.2 Real Data : wine and Wdbc

We used two kinds of real data sets used in [7] for the comparison of the performance because our method is the extension of their algorithm. The first data set is wine recognition data (wine) that contains results of chemical analysis of wines grown in different cultivars. It is composed of 179 samples with 13 features from 3 classes. The other data set is the Wisconsin diagnostic breast cancer data (Wdbc) that was used to obtain a diagnosis (benign or malignant) based on 30 features from cell nuclei presented in an image. It has 576 samples.

In this experiment, the dataset was first randomly divided into two sets: one for training, another for testing. We used 70% of samples as training data and 30% of samples as testing data. The entire procedure is repeated 20 times, each initialized with K=30 and the stopping threshold at 10^{-7}. We evaluate the results by considering components as clusters and comparing them with the ground truth

Table 1.2. The error rate of experiments using wine and wdbc

	EPSCMIX		**Algorithm of Law et al.**		
	error rate	$\hat{c}$	error rate	error rate for post-processing	$\hat{c}$
wine	0.06321 (0.04)	3	0.0661	0.0661	3.1
Wdbc	0.0801 (0.03)	2	0.0955	0.0935	5.65

classes. The test data is assigned to the cluster that most likely generated it, and the sample is classified to the cluster. These conditions are applied for all experiments using real data sets.

In our experiment, one of options that can be adjusted is the number of features. We consider some cases by selecting some high-ranked features among the features that are ranked according to the information gain calculated by entropy based discretization method. When we use all features for wine and 15 features for Wdbc, the results are the best. The result is shown in Tab. (1.2). The "error rate" corresponds to the mean of error rates on the testing set. $\hat{c}$ denotes the number of Gaussian components estimated. The numbers in parenthesis are the standard deviation of the corresponding error rate over 20 random runs. As you see, the error rates are a little reduced in both cases. The most different thing is the number of clusters estimated for Wdbc, and it may be the reason for the performance improvement.

1.3.3 Real Data : colon cancer

The dataset consists of 40 tumor and 22 normal colon tissue samples. These samples are composed of 2,000 gene expression values with highest minimal intensity across the samples. Before we considered the clustering of this set, we have rearranged the data so that the tumors are labeled 1-40 and the normals 41-62 [17]. By

the preprocessing step (the entropy based discretization), which is required for EP algorithm, 136 significant genes among 2000 genes were found. We compared the performance according to the number of genes and ME-genes, and they are shown in Fig. (1.2). As we can see in Fig. (1.2), our approach performs better when 35 top-ranked genes are used. This may be due to the fact that noisy features can hurt clustering results. In our algorithm, if many features make too many emerging patterns, lots of ME-genes from them reduce discrimination power of feature saliency. Finally, we picked, the 35 top-ranked genes which are also consistent with [5].

Table 1.3. Classification accuracy over 20 random runs

	# genes	# ME-features	accuracy (STD)	$\hat{c}$
colon	35	35	**0.88158 (0.068)**	2.15
	35	10	0.87895 (0.059)	2.2
prostate	35	35	**0.90645 (0.05)**	3.45
	35	10	0.903226 (0.04)	3

The averaged results of classification accuracy are shown in Tab. (1.3). The "accuracy" corresponds to the mean of the accurate rates on the testing set when the clustering results are compared with the ground truth labels. STD denotes the standard deviation of the corresponding quantities and $\hat{c}$ means the number of Gaussian components estimated. Our approach performs better than the other clustering approaches or is comparable with them. For example, EPPC [6], which is a new iterative top-down subspace search method applying EPs based on the framework of ORCLUS [18], has 0.799 0.857% accuracy (on the 70% training set over 50 runs). CAST [19] has 88.7% accuracy (in the LOOCV evaluation) and

several methods such as Nearest Neighbor, Boosting, etc. in [20], have much lower accuracy than ours. It is also comparable with the results of classification methods like EP-based classification by [3] or SVM. In [3], the accuracy of four methods (EP-based, LDA, 3-NN, SVM) is between 0.766% and 0.886% (on the testing set over 50 random runs).

We now show the feature saliency for each cluster in Tab. (1.4) (in other words, subspaces of each cluster). Diff denotes the difference of feature saliency value between normal and tumor, B denotes marker genes identified by proposed algorithm, M denotes smooth muscle genes, R denotes the ribosomal protein genes and E denotes whether identified genes by [1] or not. Top 18 genes are characteristic of tumor cluster and the others (bottom) are genes mainly used for normal cluster. As you can see in the result, we identified eleven marker genes, which are composed of five genes being characteristic of a tumor cluster (U21090, T57619, H55758, R36977 and Z50753) and six genes being representitive of a normal cluster (M26383, X12671, T96873, M76378, J02854 and U25138) when threshold is 0.5. Some of them can be interpreted by already known facts, and the others need to be explained in the future. For example, polyserase-2 (U21090) is detected in several cancer cell lines including colon adenocarcinomas in [21], and Ribosomal protein S6 (T57619) can be explained by the fact that the intensity of the ribosomal protein genes is relatively low in the normal colon tissues than in tumor tissues from [15] and [20]. For Alpha enolase (H55758), we can refer the enolase activities explained in [22]. In the case of the MONAP gene (M26383), a very interesting interpretation can be possible. It was believed that over-expression of MONAP (M26383) plays an important role in tumor angiogenesis and tumor aggression [1]. As you can see in Tab. (1.4), it is a marker gene which has high feature saliency value for normal cluster relative to tumor cluster. This means that it would be down-regulated in all normal tissue

Table 1.4. 33 genes with mean of feature saliency over 20 random runs

GeneBank Accesstion No.	Feature Saliency normal	tumor	diff	Name	B	M	R	E
U21090	0.167018	0.962104	0.795086	Human DNA polymerase delta small subunit mRNA	✓			
T57619	0.209899	0.965287	0.755388	40S RIBOSOMAL PROTEIN S6 (Nicotiana tabacum)	✓		✓	
H55758	0.232204	0.945239	0.713035	ALPHA ENOLASE (HUMAN)	✓			✓
R36977	0.235194	0.933298	0.698104	TRANSCRIPTION FACTOR IIIA	✓			
Z50753	0.351667	0.871837	0.52017	H.sapiens mRNA for GCAP-II/uroguanylin precursor	✓			
R84411	0.553432	0.903717	0.350284	SMALL NUCLEAR RIBONUCLEOPROTEIN ASSOCIATED PROTEINS B AND B'				
H23544	0.5591	0.932838	0.373738	GTP-BINDING NUCLEAR PROTEIN RAN (Homo sapiens)				
U09587	0.610685	0.820255	0.20957	Human glycyl-tRNA synthetase mRNA, complete cds				
M16937	0.61237	0.908202	0.295833	Human homeo box c1 protein, mRNA, complete cds.				
U32519	0.61475	0.911533	0.296782	Human GAP SH3 binding protein mRNA, complete cds.				
T47377	0.6995	0.848499	0.148999	S-100P PROTEIN (HUMAN)				
U30825	0.719625	0.915206	0.195581	Human splicing factor SRp30c mRNA, complete cds.				
H43887	0.725775	0.905677	0.179902	183264 COMPLEMENT FACTOR D PRECURSOR (Homo sapiens)				
X53586	0.733254	0.899753	0.166499	Human mRNA for integrin alpha 6.				
H40560	0.733546	0.897696	0.164151	175410 THIOREDOXIN (HUMAN)				
X63629	0.741238	0.861703	0.120466	H.sapiens mRNA for p cadherin.				
H08393	0.75196	0.900941	0.148981	45395 COLLAGEN ALPHA 2(XI) CHAIN (Homo sapiens)				✓
R10066	0.81404	0.870241	0.056202	PROHIBITIN (Homo sapiens)				
H40095	0.896774	0.894528	0.002247	175181 MACROPHAGE MIGRATION INHIBITORY FACTOR (HUMAN);.				
U09564	0.910235	0.763688	0.146546	Human serine kinase mRNA, complete cds.				
M26383	0.913627	0.322907	0.59072	Human monocyte-derived neutrophil-activating protein (MONAP) mRNA, complete cds.	✓			✓
T92451	0.917462	0.767149	0.150313	118219 TROPOMYOSIN, FIBROBLAST AND EPITHELIAL MUSCLE-TYPE (HUMAN)		✓		
T71025	0.924445	0.719374	0.205071	84103 Human (HUMAN)				
R87126	0.934787	0.821411	0.113377	197371 MYOSIN HEAVY CHAIN, NONMUSCLE (Gallus gallus)				✓
X12671	0.936812	0.423104	0.513708	Human gene for heterogeneous nuclear ribonucleoprotein (hnRNP)	✓			
J05032	0.96184	0.712202	0.249638	Human aspartyl-tRNA synthetase alpha-2 subunit mRNA, complete cds.				
T60155	0.970551	0.590048	0.380503	81422 ACTIN, AORTIC SMOOTH MUSCLE (HUMAN)		✓		
T96873	0.978702	0.443865	0.534838	HYPOTHETICAL PROTEIN IN TRPE 3'REGION (Spirochaeta aurantia)	✓			
M63391	0.986911	0.516009	0.470903	Human desmin gene, complete cd		✓		✓
M76378	0.997333	0.454979	0.542354	Human cysteine-rich protein (CRP) gene, exons 5 and 6	✓			
M22382	0.999699	0.728405	0.271294	MITOCHONDRIAL MATRIX PROTEIN P1 PRECURSOR (HUMAN)				
J02854	1	0.374897	0.625103	MYOSIN REGULATORY LIGHT CHAIN 2, SMOOTH MUSCLE ISOFORM (HUMAN)	✓	✓		✓
U25138	1	0.458372	0.541628	Human MaxiK potassium channel beta subunit mRNA	✓	✓		

rather than up-regulated in all tumor tissue. [23] observed that the expression and induction of the CRP gene (M76378), which has been associated with protection against DNA damage, oxidative stress and apoptosis, is down-regulated in tumor tissue. And, we found five smooth muscle genes (J02854, U25138, T60155, M63391 and T92451) with high feature saliency for a normal cluster (smooth muscle-related genes showed high expression levels in the normal tissue samples compared to the tumor samples [15], [20]) and J02854 and U25138 genes among them are selected as marker genes.

Regarding some of genes which have high feature saliency for both clusters reported in Tab. (1.4), it should be emphasized that this was achieved by applying EPs during clustering. This means that these kinds of genes closely interact with marker genes.

1.3.4 Real Data : prostate cancer

This data set consists of 52 tumor and 50 normal prostate tissue samples. They have 12,600 genes with highest minimal intensity across the samples. As colon data sets, we have rearranged the data so that the tumors are labeled 1-52 and the normals 53-102. By the preprocessing step, 1,607 significant genes among 12,600 genes were found, but we considered two cases: 35 or 100 top ranked genes by an entropy method because the performance got worse according as the number of genes increases. The averaged results of classification accuracy are shown in Tab. 1.3. This result shows comparable accuracy with the results (greater than 90% accuracy in LOOCV evaluation) using a k-nearest neighbor (k-NN) supervised algorithm of [16].

In this experiment, we found eight marker genes (33674, 40435/40436$_g$, 31527, 37366, 33614, 33121$_g$, 36589 and 40282$_s$) when the threshold is 0.5. Some of them can be interpreted by previous works, and the others need to be explained in the future.

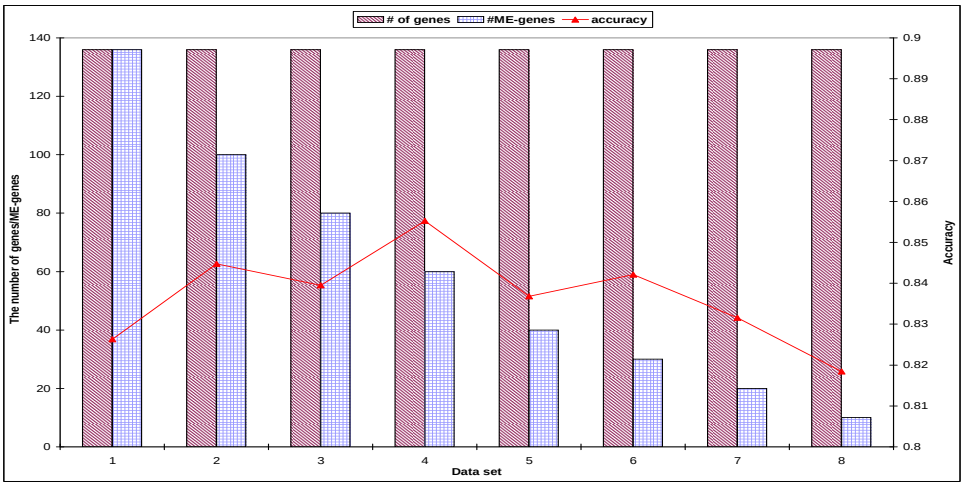
For the adipsin marker gene (40282_s), [24] suggested that prostate cancer cells grow differently in an adipocyterich environment such as bone marrow. DKFZp564A072, cDNA sequence, was specific to prostate epithelial cells, and its expression level was elevated significantly in prostate cancers [25]. The interesting thing is the number of clusters estimated (3.45 in best accuracy), which are two normal clusters and one tumor cluster. This result can be used to decide the rational selection of patients at high risk for relapse for clinical testing adjuvant therapeutics. [16] also showed the feasibility of using gene expression differences to predict the identity of prostate samples in their experiments. For this, the gene expression of 35 top ranked genes is shown in Fig. (1.4).

As we can see in Fig. (1.4), most genes selected by the preprocessing step have obvious differences of gene expression between tumor and normal samples. And, we found several expression patterns (by interaction among genes) from their differences. Such patterns are naturally reflected to our feature saliency by EPs. The summary about feature saliency of top-ranked features will be shown in an appendix. Eleven genes among 35 top-ranked genes are the same as these most commonly used in the 16-gene model of [16].

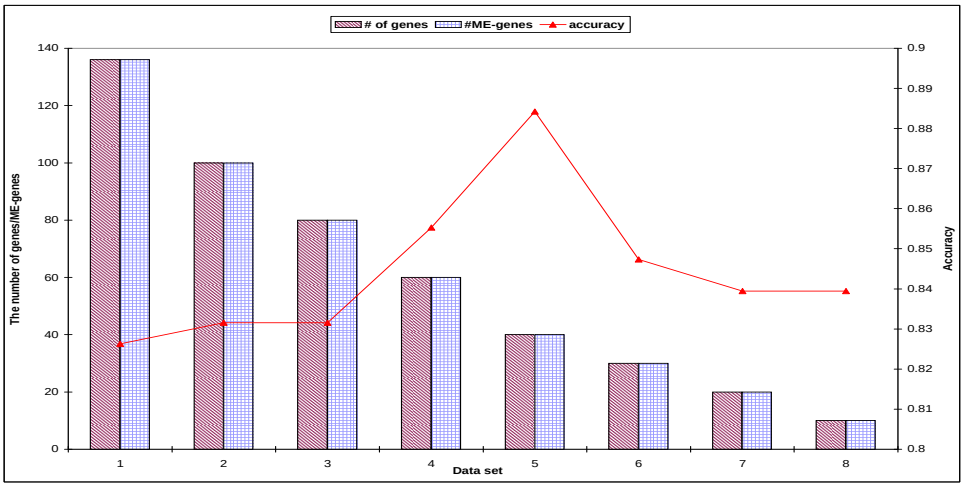
1.4 Conclusion

In this chapter, we have introduced an EPSCMIX algorithm, which employs EM algorithm to estimate the best number of classes for mixture based semi-supervised subspace clustering and salient features selected by EPs algorithm. The major strength of the proposed algorithm is that by using a new approach to get feature saliency, the algorithm finds effectively the subspace (feature subsets) of classes as well as classes at the same time. This approach is also not only an efficient method to estimate the number of classes but also less sensitive to local minima by

initialization than standard EM by applying MML criterion. We applied this algorithm in the cancer problem and achieved better or comparable performance than other supervised/unsupervised algorithms, even though all classes are assumed to be Gaussian mixtures. The robustness of using emerging patterns based on mixture models, as well as using the Gaussian mixture model for subspace clustering was also demonstrated on both synthetic and real data sets. Our results of both cancer data sets were also consistent with current biological knowledge. In future work, we will adapt principles other than MML, such as BIC, to perform model selection. We can replace border based EPs algorithm by other types of EPs algorithm. Finally, we can verify our algorithm on other several types of cancer microarray data sets for corresponding performance analysis, comparison, and meanings exploration.

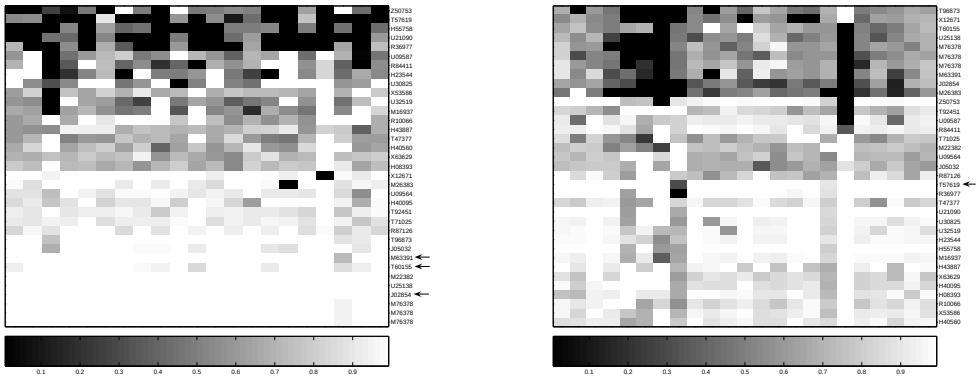


(a)



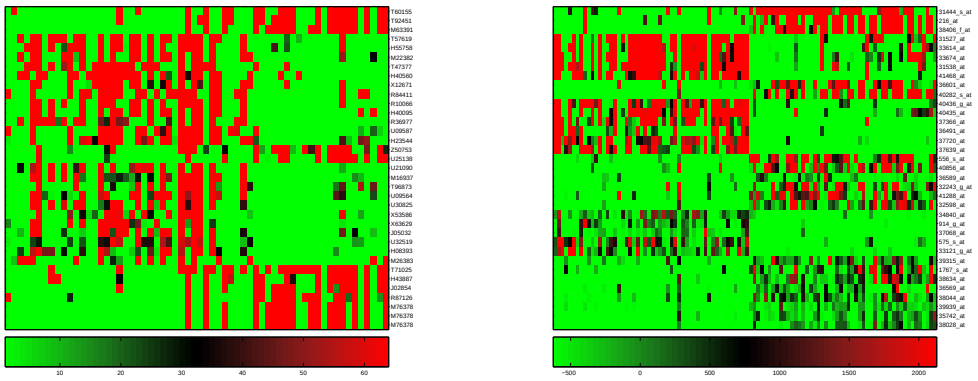
(b)

Figure 1.2. The accuracy comparison according to the number of genes and ME-genes. The red lines represent the mean of accuracy. (a) The effect of the number of genes and ME-genes, (b) The effect of the number of genes.



(a) (b)

Figure 1.3. Clustergram of feature saliency for colon data over 20 random runs. Genes are marked with arraows if they are smooth muscle genes (a) or ribosomal protein genes (b). (a) Normal class (b) Tumor class .



(a) (b)

Figure 1.4. Gene expression correlates of feature saliency. 35 top-ranked genes by entropy method are used. Genes and samples are shown as ordered by Gene Cluster. The expression of each gene in each sample is represented by the number of standard deviations above (red) or below (green) the mean for that gene across all 62 samples for colon cancer or 102 samples for prostate cancer. (a) Colon data (b) prostate data.

CHAPTER 2

A NEW MODEL SELECTION METHOD IN EPSCMIX ALGORITHM

2.1 Introduction

In likelihood analysis of mixture models, maximum-likelihood (ML) estimation is usually done by EM algorithm, which require the number of components (k) to be known. In this case, multiple models have to be estimated and certain model selection criterion has to be used. Several such strategies are available in the literature (refer to a special Journal of Mathematical Psychology issue on model selection). The most representitive criteria are the Akaike information criterion (AIC, [26]), the Bayesian Information Criterion (BIC, [27]), cross-validation (CV, [28]), minimum description length (MDL, [29]) and minimum message length (MML, [30]). In those five methods, ML is always used as a goodness of fit measure, but they differ in how model complexity is conceptualized. There are more criteria such as the information complexity (ICOMP, [31]), the integrated classification likelihood (ICL, [32]) and the Laplace-empirical criterion (LEC, [33]) and etc. However, only employing these criteria did not overcome drawbacks of a standard EM (sensitivity to initialization and possible convergence to the boundary of the parameter space) ([34]). So [13] presented the unsupervised learning algorithm which are seamlessly integrating estimation and model selection and show their approach outperforms the existing criteria. But it also has an local maxima problem in certain situation. Recently, [35],[36] and [34] have been presented Split and Merge EM algorithm which alternately splits and merges components, estimating k and other parameters of components simulta-

neously. Each of them presented their criteria for split and merge operations. [37] proposed the model metaselection approach which can be used to choose a model selection strategy based on a data-driven adaptive procedure. However, although some alternative ways to estimate k have been proposed, finding it effectively considering a trade-off between goodness-of-fit and complexity of the models involved, is still an open problem ([34]).

In this paper, we introduce a new subspace clustering algorithm called EPSCMIX (Emerging Pattern Subspace Clustering by MIXture models), which can be applied to prediction of pathological features of diseases using microarray expression data as well as image pattern recognition. This is based on a feature saliency measure that is obtained by the EPs algorithm, which is data mining technique to be able to effectively find patterns discriminating different classes, and ML likelihood estimation by the EM algorithm. In terms of an estimation of k, we applied the annihilation process seamlessly integrating estimation and model selection like [13]. But to overcome drawback of removing action of component (specially, in the case where we use a small data set, it is hard to perfectly estimate many paramaters in large initial value for k to remove components), we also implemented agglomerative step with merge criterion of [34]. Our simulation show improved convergence capability relative to CEM together with MML like criterion.

2.2 Methods

2.2.1 Model Selection

In this section, three representative criteria are introduced. They are the Akaike Information Criterion (AIC), minimum description length (MDL) and mini-

mum message length (MML). In those methods, ML is always used as a goodness of fit measure, but they differ in how model complexity is conceptualized.

The goal of AIC is to minimize the Kullback- Leibler (K-L) distance of the selected density from the true density. The AIC rule is to select the predictive density that has the lowest estimated K-L discrepancy, which amounts to the minimization of Eq.(2.1). AIC for a given model are defined as follows:

$$AIC = -2logp(\mathcal{Y}|\theta) + 2c \tag{2.1}$$

where θ is a MLE estimate, log is the natural logarithm of base e, c is the number of parameters and N is the sample size. The first term represents a lack of fit measure, the second term represents a complexity measure. In complexity of AIC like BIC, dimension of model complexity, is not considered. The other two selection methods, MDL and MML, described next, are sensitive to dimension of model as well as the number of parameters.

These two criteria share the basic idea that the best model is one that facilitates the shortest encoding of observed data but the codes that are used are quite different. While the MDL codes minimize worst-case relative code-length (regret),the two-part codes used by MML are designed to minimize expected absolute code-length from a subjective prior distribution defined on the collection of models and parameters under consideration. The one version of MDL (is coincident with the well-known MDL criterion) and MML derived from the particular form in Eq.(2.2) of the MML approach by [13] are defined as follows:

$$\begin{aligned} \hat{\theta} &= arg\min_{\theta}\{-logp(\theta) - logp(\mathcal{Y}|\theta) \\ &\quad + \frac{1}{2}log\det I(\theta) + \frac{cK+K}{2}(1 + log\frac{1}{2})\} \end{aligned} \tag{2.2}$$

Where I(θ) is the Fisher information matrix, detI(θ) denotes its determinant, K is the number of components.

$$MDL = -logp(\mathcal{Y}|\theta) + \frac{c}{2}log(N) \tag{2.3}$$

$$MML = \frac{N}{2}\sum_{\alpha>0} log(\frac{n\alpha_m}{12}) + \frac{K}{2}log\frac{N}{12} + \frac{K(c+1)}{2} - logp(\mathcal{Y}|\theta) \tag{2.4}$$

This MDL criterion is obtained by two assumptions which a flat prior $p(\theta)$ and large N. And the MML criterion is derived by using the standard noninformative Jeffreys' prior.

In the referenced papers, they show that AIC criterion has a strong tendency to select models that are too complex ([38]) while MDL to underestimate the number of components. And MML was the best two part message length estimate discussed ([30]). However, in many cases, the methods described here not only perform very well but also have cases where they perform suboptimally compared to other state-of-the-art methods. Often these are the reasons: First, an asymptotic formula was used and the sample size was not large enough to justify this. Second, the normalized maximum likelihood (NML) distribution was undefined for the models under consideration, and this was solved by cutting off the parameter ranges at ad hoc values. Finally, the true distribution is not under consideration. Anyhow, all model selection methods that are used in practice choose a trade-off between goodness-of-fit and complexity of the models involved. MDL provides one particular means of achieving such a trade-off ([39]).

Our proposed algorithm have been designed to be used for the analysis of microarray. This data is one of cases where while model complexity is so high, usually enough data has not been given. Because of this property, we desire to

show the result of simulations using three criteria based on our new approach, which seamlessly integrate model selection and our novel estimation approach.

Next equation show how above MML criterion is adopted in this experiments. We omit equations for AIC and MDL because of their simplicity.

$$F_{(t)}(K,\Phi)_{MML} = -logp(\mathcal{Y}|\theta) + \frac{1}{2}(K+KD)logN$$
$$+\frac{R}{2}\sum_{j=1}^{K}\sum_{l=1}^{D} log(N\alpha_j\rho_{jl}) + \frac{S}{2}\sum_{j=1}^{K}\sum_{l=1}^{D} log(N(1-\rho_{jl})) \tag{2.5}$$

This criterion has easily understandable interpretations. $\frac{1}{2}(K+KD)logN$ is a parameter code length corresponding to K α_j values and KD ρ_{jl} values. The other terms are the code lengths required for estimating each θ_{jl} and λ_l : $\frac{R}{2}log(N\alpha_j\rho_{jl})$ and $\frac{S}{2}log(N(1-\rho_{jl}))$. The order one term was discarded.

2.2.2 EPSCMIX algorithm using a new model selection

2.2.2.1 Merge criterion

This idea as I have mentioned above has been implemented by [40] as agglomerative EM (AEM). They merge two components of the k-component one based on the symmetric KL divergence. And [13] also proposed new component-wise version of EM which I have already mentioned before. In this paper, this merge criterion has been substituted by removal operation of the component with the smallest weight.

In this paper, we applied the correlation coefficient proposed by [34] for the merge criterion. Let $P_j(\hat{\theta}) = (w_{1j}, w_{2j}, ..., w_{nj})^T$ denote the n-dimensional vector which is composed of the posterior probability of y_i belonging to the jth component. The correlation coefficient as the merge criterion is the follows.

$$\mathcal{M}(j,k;\hat{\theta}) = \frac{(P_j(\hat{\theta}) - \bar{P}_j(\hat{\theta}))^T(P_k(\hat{\theta}) - \bar{P}_k(\hat{\theta}))}{\parallel (P_j(\hat{\theta}) - \bar{P}_j(\hat{\theta})) \parallel \parallel (P_k(\hat{\theta}) - \bar{P}_k(\hat{\theta})) \parallel} \tag{2.6}$$

$$\mathcal{M}(j,k) = \max_{jk}\{\mathcal{M}(j,k;\hat{\theta})^2 : j,k = 1,...,K\} > \mathcal{T}_{merge} \tag{2.7}$$

Where $\|\cdot\|$denotes the Euclidean norm, $\bar{P}_j(\hat{\theta})$ is the n-dimensional mean vector with all entries equal to $\frac{1}{n}\sum_{i=1}^{n} w_{ij}$, $\mathcal{T}_{merge}$ is a prescribed threshold. So, the more $\mathcal{M}^2$ is close to 1 ($\mathcal{M}$ is close to +1 or -1), the higher probability which these two components are likely to come from same component is. Therefore, these two components must be merged. When merging components j and k of the mixture, the resulting component must retain the combined probability, mean, and covariance. So merged component mixture is defined as follows.

$$\begin{aligned} \alpha_m &= \hat{\alpha}_j + \hat{\alpha}_k \\ \theta_m &= \frac{\hat{\alpha}_j\hat{\theta}_j + \hat{\alpha}_k\hat{\theta}_k}{\hat{\alpha}_j + \hat{\alpha}_k} \\ \rho_m &= \hat{\rho}_j + \hat{\rho}_k \end{aligned}$$

The algorithm is composed of three phases : preprocessing, initialization, and iteration. We use two kinds of data sets, labeled discretized data and unlabeled raw data. The former is for EPs algorithm and the other is for EM algorithm. First, we make the former set as applying an entropy based discretization method, and for the other, we standardized the original data set with the method that each column is standardized to have mean zero and unit standard deviation, and then each row is standardized to have zero mean and unit standard deviation. In the initialization phase, we initialize all unknown parameters such as mixture parameters $\{\theta_{jl}\}$,$\{\alpha_j\}$, parameters of common distribution $\{\lambda_l\}$ and feature saliencies $\{\rho_{jl}\}$. In the iteration phase, two steps of EM and updating feature saliencies are iteratively performed until Eq. (2.5) is satisfied. The feature saliencies are updated based on training data class labels that are most likely generated by previous EM-steps. After performing E-step and M-step, if α_j becomes zero, the j-th class will be pruned, and if ρ_{jl} becomes

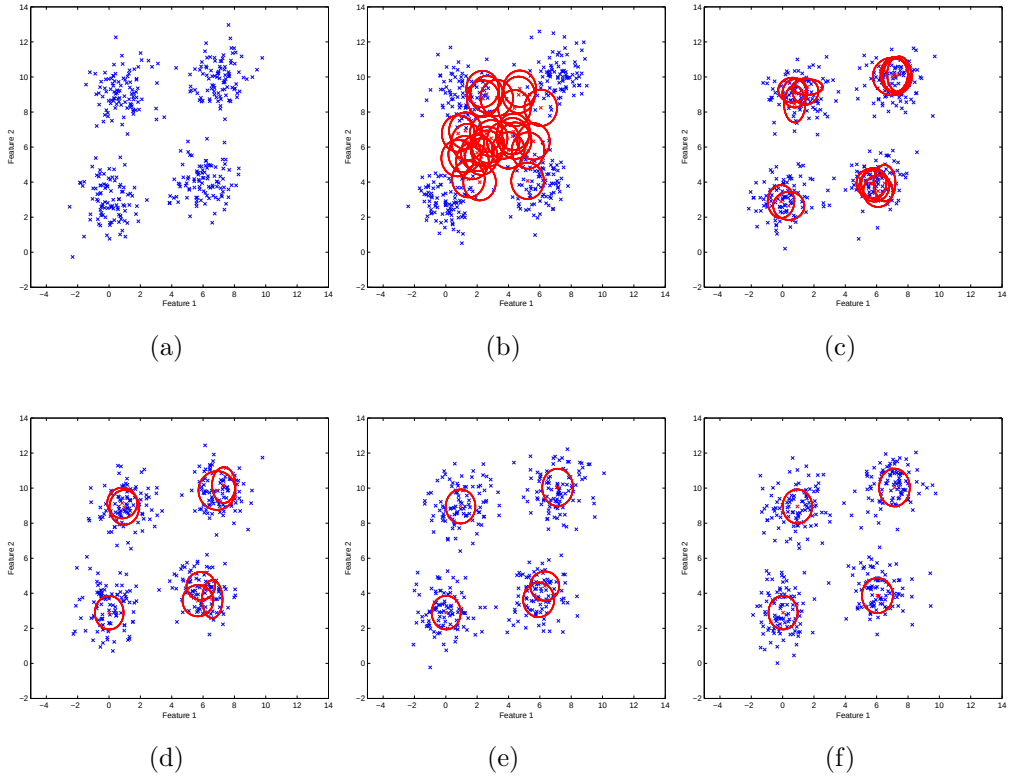


Figure 2.1. Fitting a Gaussian mixtures. This is one of 50 random runs by using EPSCMIX algorithm with merge criterion and MDL together(800 samples). The solid ellipses represent the estimated Gaussian mixture components. (a) The data set (b) Initialization (K=30) (c) K=17 (d) K=8 (e) K=5 (f) K=4.

one, $q(y_l|\lambda_l)$ will be pruned; otherwise, if ρ_{jl} becomes zero, $p(y_l|\theta_{jl})$ will be pruned for j-th component.

2.3 Experiments

We implemented the EPSCMIX approach with the Java language and used not only the synthetic data but also the well-known datasets, such as the colon tumor set of [15] and the prostate tumor set of [16] from Kent Ridge Bio-medical Data Set Repository (http://sdmc.lit.org.sg/GEDatasets/Datasets.html), to demonstrate the robustness of our new approach. Additionally, we considered two image data sets

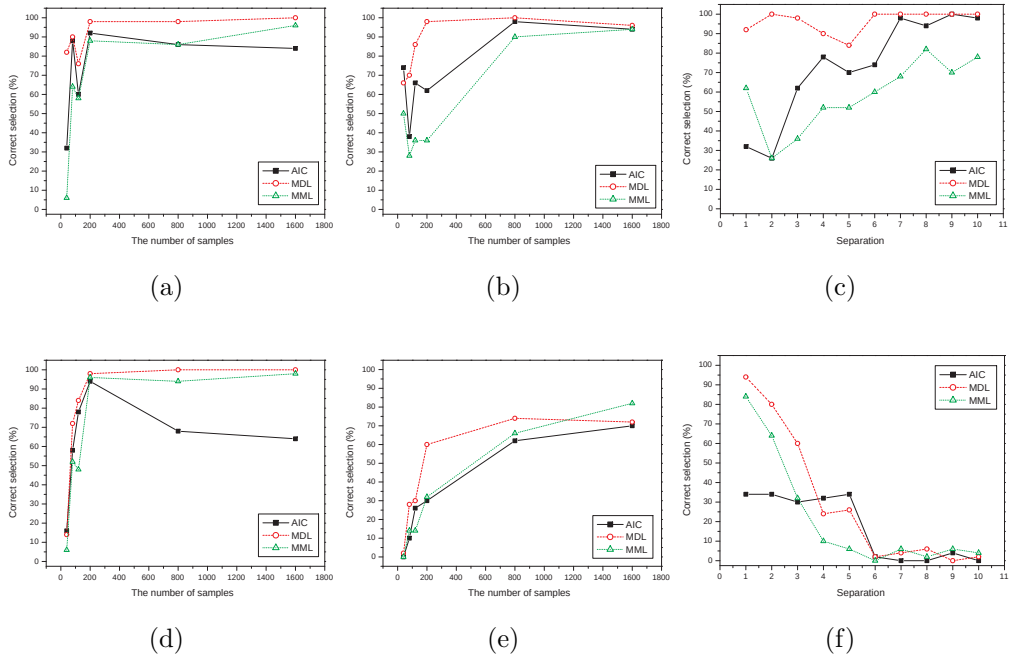


Figure 2.2. The percentage of success selection (k) versus the sample size and separation between components using two annihilation approaches. M and R denote the methods merging same components (M) and removing redundant components (R). (a) simulation 1 (M) (b) simulation 2 (M) (c) simulation 3 (M) (d) simulation 1(R) (e) simulation 2 (R) (f) simulation 3(R) .

such as the wine recognition data set and the Wdbc(Wisconsin diagnostic breast cancer) data set from UCI machine learning repository (http://www.ics.uci.edu/ mlearn /MLRepository.html). Data sets are summarized in Tab. 2.1. Each data set has N data points with D features from c classes.

2.3.1 Synthetic Data

We repeated all experiments 50 times, each initialized with K=30, the stopping threshold at 10^{-7} and the component merge threshold at 0.5. We compared the results of two component annihilation methods, merging same components (M) and removing redundant components (R).

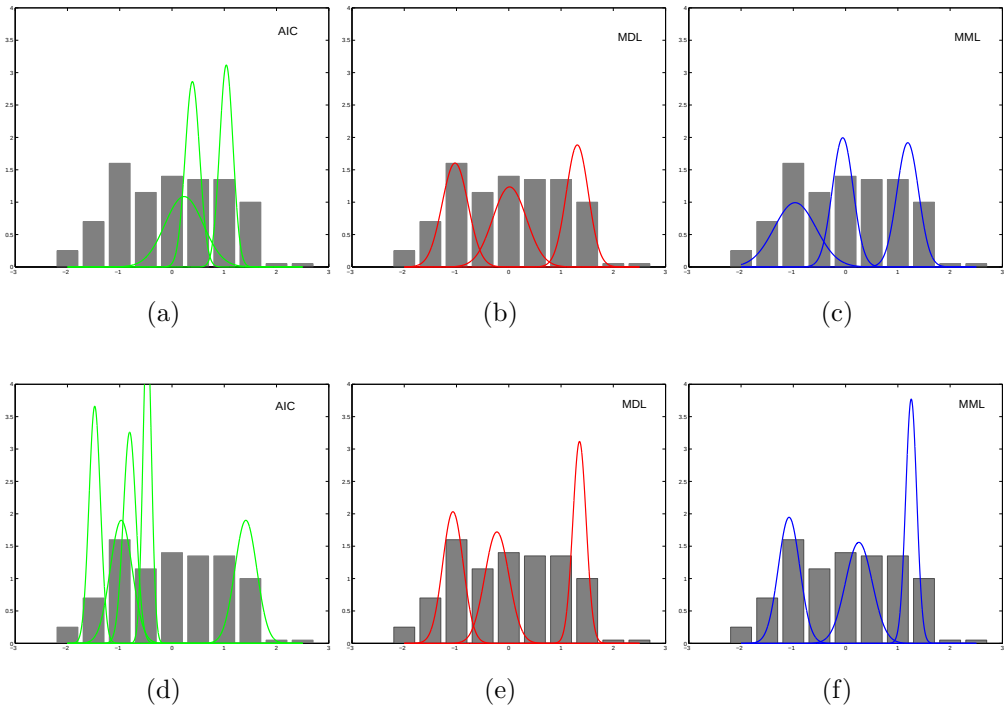


Figure 2.3. A Gaussian mixture estimates obtained by the AIC, MDL and MML criteria together with M (top row) and R (bottom row) approaches for the first feature of the wine data set. The bar graphs are histograms of the first feature of the wine data. (a) AIC with M (b) MDL with M (c) MML with M (d) AIC with R (e) MDL with R (f) MML with R.

Table 2.1. Data description

Name	**N**	**D**	**c**
synthetic data1	Multi sets: 40,80,120,200,800,1600	10	4
synthetic data2	Multi sets: 40,80,120,200,800,1600	5	2
synthetic data3	200	5	2
wine recognition	178 (59 class1, 71 class2, 48 class3)	13	3
Wdbc	569 (357 benign, 212 malignant)	30	2
colon cancer	62 (40 tumor, 22 normal)	2000	2
prostate cancer	102 (52 tumor, 50 normal)	12600	2

First, we use 10-dimensional samples (being considered the various number of samples such as 40, 80, 120, 200, 800 and 1600) from a mixture of four equiprobable Gaussians $\mathcal{N}(m_i, I)$, i=1,2,3,4, where

$$m_1 = \begin{pmatrix} 0 \\ 3 \end{pmatrix}, m_2 = \begin{pmatrix} 1 \\ 9 \end{pmatrix}, m_3 = \begin{pmatrix} 6 \\ 4 \end{pmatrix}, m_4 = \begin{pmatrix} 7 \\ 10 \end{pmatrix}$$
$$\alpha_1 = \alpha_2 = \alpha_3 = \alpha_4 = 0.25$$

and eight noisy features sampled from a $\mathcal{N}(0, I)$ in Fig. (2.1 (a)). One of the 50 runs by using M approach with MDL (800 samples) is shown in Fig. (2.1). And as you can see in Fig. (2.2:(a),(d)), when we have enough samples (in this case, more than 800 samples), both annihilation methods almost exactly identified the four components and the saliencies of two features to these components. And it was also showed that while in the case of being used M approach, the MDL criterion consistently show good performance without regard to sample size, R approach was a little more sensitive in the sample size.

Next, we consider a situation where both two components are composed of five dimensions. This is why because eight noise features are early removed in preprocessing step, only two features are considered for the estimation in the first example. The Gaussians $N(m_i, I)$, i=1,2, where
$\alpha_1 = \alpha_2 = 0.5$ and $m_1 = [0, ..., 0]^T$, $m_2 = [3, ..., 3]^T$. Fig. (2.2:(b),(e)) show that M method is more confident relative to R method in all situations.

Finally, to study how these methods perform when the distance between the two components varies, we used a 5-dimensional Gaussian mixture $N(m_i, I)$, i=1,2 with $m_1 = [0, ..., 0]^T$, $m_2 = [\delta, ..., \delta]^T$. These results reveal an excellent performance of our M approach. Notice that for $\delta > 3$, the results of R approach are now disastrous.

2.3.2 Real Data : wine and Wdbc

We used two kinds of real data sets used in [7] for the comparison of the performance because our method is the extension of their algorithm. The first data set is wine recognition data (wine) that contains results of chemical analysis of wines grown in different cultivars. It is composed of 179 samples with 13 features from 3 classes. The other data set is the Wisconsin diagnostic breast cancer data (Wdbc) that was used to obtain a diagnosis (benign or malignant) based on 30 features from cell nuclei presented in an image. It has 576 samples.

In this experiment, the dataset was first randomly divided into two sets: one for training, another for testing. We used 50% of samples as training data and 50% of samples as testing data. The entire procedure is repeated 50 times, each initialized with K=30, the stopping threshold at 10^{-7} and the component merge threshold at 0.5. We evaluate the results by considering components as clusters and comparing them with the ground truth classes. The test data is assigned to the cluster that most likely generated it, and the sample is classified to the cluster. These conditions are applied for all experiments using real data sets.

The summary of this simulation is shown in Tab. (2.2). The "accuracy" corresponds to the mean of the accurate rates on the testing set when the clustering results are compared with the ground truth labels. The "error rate" corresponds to the mean of error rates on the testing set. The "stdev" are the standard deviation of the corresponding error rate over 50 random runs. $\hat{c}$ denotes the number of Gaussian components estimated. Interestingly, the selected numbers of mixture components coincide with k=3 and k=2, for the wine and wdbc data sets, respectively. Notice the big difference of k estimated in wdbc data set. Fig. (2.3) shows histograms of the first feature of wine data sets with the mixture densities obtained by our algorithm. In any case, our method outperforms the other.

Table 2.2. Classification accuracy over 50 random runs on wine and wdbc data sets

[WINE]

	Our M		**Our R**		**Law et al.' R**
	MDL	**MML**	**MDL**	**MML**	**MML**
accuracy	0.951948	0.95437	0.94635	0.94616	0.9339
error rate	0.048052	0.04563	0.05365	0.05383	0.0661
stdev	0.028666	0.02192	0.02600	0.02800	0.0391
$\hat{c}$	3(#47)	3(#49)	3(#40)	3(#48)	3.1(N/A)

[WDBC]

	Our M		**Our R**		**Law et al.' R**
	MDL	**MML**	**MDL**	**MML**	**MML**
accuracy	0.91464	0.91767	0.91606	0.90556	0.9045
error rate	0.08535	0.08232	0.08394	0.09443	0.0955
stdev	0.04530	0.04669	0.04207	0.04214	0.0199
$\hat{c}$	2(#50)	2(#50)	2(#50)	2(#50)	5.65(N/A)

2.3.3 Real Data : colon and prostate cancer

The colon data set consists of 40 tumor and 22 normal colon tissue samples. These samples are composed of 2,000 gene expression values with highest minimal intensity across the samples. Before we considered the clustering of this set, we have rearranged the data so that the tumors are labeled 1-40 and the normals 41-62 ([17]). By the preprocessing step (the entropy based discretization), which is required for EP algorithm, 132 significant genes among 2000 genes were found. In our algorithm, if many features make too many emerging patterns, lots of ME-features from them reduce discrimination power of feature saliency. Finally, we picked, the 35 top-ranked genes which are also consistent with [5]. In this experiment, the dataset was first randomly divided into two sets: one for training, another for testing. We used 70% of

Table 2.3. Classification accuracy over 50 random runs on colon and prostate data sets

[COLON]

	Our M		**Our R**	
	MDL	**MML**	**MDL**	**MML**
accuracy	0.87970	0.86370	0.86098	0.86706
error rate	0.12030	0.13630	0.13902	0.13294
stdev	0.07735	0.06907	0.05721	0.04822
$\hat{c}$	2(#42)	2(#39)	2(#42)	2(#28)

[PROSTATE]

	Our M		**Our R**	
	MDL	**MML**	**MDL**	**MML**
accuracy	0.91774	0.90015	0.89780	0.90681
error rate	0.08226	0.09985	0.10203	0.09319
stdev	0.06153	0.05464	0.05928	0.0372
$\hat{c}$	2.59	2.19	2.558	2.326

samples as training data and 30% of samples as testing data. The other parameters are the same as the ones have been used before.

The averaged results of classification accuracy are shown in Tab. (2.2). Our approach performs better than the other clustering approaches or is comparable with them. The most beauty of proposed algorithm is that we can identify marker feature of estimated component using feature saliency value. In this dataset, we identified eleven marker genes, which are composed of five genes being characteristic of a tumor cluster (U21090, T57619, H55758, R36977 and Z50753) and six genes being representitive of a normal cluster (M26383, X12671, T96873, M76378, J02854 and U25138) when threshold is 0.5. Some of them can be interpreted by already known facts, and the others need to be explained in the future. For example, polyserase-2 (U21090) is detected in several cancer cell lines including colon adenocarcinomas

in [21], and Ribosomal protein S6 (T57619) can be explained by the fact that the intensity of the ribosomal protein genes is relatively low in the normal colon tissues than in tumor tissues from [15] and [20].

The prostate cancer data set consists of 52 tumor and 50 normal prostate tissue samples. They have 12,600 genes with highest minimal intensity across the samples. As colon data sets, we have rearranged the data so that the tumors are labeled 1-52 and the normals 53-102. By the preprocessing step, 1,607 significant genes among 12,600 genes were found, but we considered two cases: 35 top ranked genes by an entropy method because the performance got worse according as the number of genes increases. The averaged results of classification accuracy are shown in Tab. (2.3). This result shows comparable accuracy with the results. In this experiment, we found eight marker genes (33674, $40435/40436_g$, 31527, 37366, 33614, 33121_g, 36589 and 40282_s) when the threshold is 0.5. However, as you can see in Tab.(2.3), all criteria used in this example have a problem to identify expected two components for prostate data set in 50 simulations. Two reasons can be considered. One is that the number of samples is not large enough for this mixtures and the other is the possibility of unknown component (for example, separation of patients at high risk for relapse from normal patients). In this case that while the maximum number of parameters (c) is K+R*K*D(when K is 30 and D is 35, c is 2130), the number of samples is 102, formal one makes sense.

2.4 Discussion

The proposed algorithm can determine both the number of components and feature subsets. However, because of the amount of computation and effectiveness for EPs algorithm, using the small number of feature subsets is more efficient. In this case, the goals of our algorithm will be the estimation of the exact number of

mixtures and the identification of marker features for each of them. For feature selection, there are many kinds of methods in the world and any of them can be applied.

The another strength of the proposed algorithm is that by applying merge criterion to reduce mixtures, the algorithm can fit more accurate mixtures. For this criterion, the merge threshold is required but it is not difficult to handle. Although because small thresholds encourage merge operation, it should be prompted as small positive number, practically, a little bit bigger threshold would be better for the real data including noise. As Wang's simulation (2004), our experiments is not very sensitive to this threshold.

2.5 Conclusion

In this paper, we have introduced a new EPSCMIX algorithm that applies new agglomerative step to reduce the components, which is different from the first version of EPSCMIX algorithm. And we make comparative experiments of using this new approach together with three model selection criterion such as AIC, MDL, and MML. The robustness of using emerging patterns based on mixture models, as well as using the Gaussian mixture model for subspace clustering was demonstrated on both synthetic and real data sets. And experiments show that a new proposed algorithm outperforms the previous one and MDL criterion works well consistently. For the analysis of microarray cancer data sets, we comment that if there is large enough data relative to the number of parameters, our proposed algorithm can be used to find both sub cluster and bio-marker and as we have already shown the possibility in this paper, it will work very well.

CHAPTER 3

UNSUPERVISED GENE SELECTION VIA A NOVEL HYBRID APPROACH

3.1 Introduction

One of the major challenges in gene microarray analysis is to identify a small set of informative genes (a.k.a. features) from a large amount of gene expression data. Early research efforts mostly focus on gene selection based on known phenotype information (supervised feature selection). However, a certain set of genes might correspond to unknown phenotypes, and thus it is important to develop unsupervised gene selection methods with clustering to lead to re-definition of phenotypes.

Unsupervised gene selection can be stated as an optimization problem in terms of search strategy and evaluation criterion. When using a search strategy, a variety of ways have been explored to investigate the solution space and then to generate candidate features [41, 42, 43]. The evaluation criterion is then used to evaluate the quality of candidate genes. Greedy hill-climbing methods such as SFS (Sequential Forward Selection) and SBS (Sequential Backward Selection) are commonly used as a search strategy. Evaluation criteria for unsupervised gene selection methods can be grouped as filters, wrappers, or hybrids. Filter approaches use the general property of the data to select a subset of genes without involving any clustering algorithm [44, 45]. Wrapper approaches apply criterion functions that utilize the clustering result to obtain gene subsets [46, 47, 7]. Generally, wrapper approaches give higher prediction accuracy but tend to be more computationally expensive than

filter approaches. Hybrid approaches try to utilize different evaluation criteria from both filters and wrappers in different search stages [48].

In recent years, the importance of unsupervised feature selection methods has been increasingly realized. Dy and Brodley (2000) introduced a wrapper framework through FSSEM (Feature Subset Selection using Expectation-Maximization clustering) with order identification which identifies the number of clusters during clustering [46]. Vaithyanathan and Dom (2000) proposed an approach to choose the feature subset and to create hierarchical clusters using a Bayesian framework [47]. Figueiredo and Jain (2004) presented an algorithm to simultaneously estimate the feature saliencies and the number of clusters using an EM algorithm [7]. Kim et al. (2004) introduced a new subspace clustering algorithm using feature saliency measure based on EPs (Emerging Patterns) to find informative features and clusters at the same time [49]. The gene-shaving method of Hastie et al. (2000) identifies subsets of features with large variations measured by its first principal component [50]. The two-way ordering approach from Ding (2003) moves irrelevant genes toward bordering area for unsupervised feature selection [51].

Due to the nature of gene expression microarray data, initial feature reduction based on filter methods can help to reduce computational expense. Quite often PCA (Principal Components Analysis), which is a classical data reduction technique [52], is applied to unsupervised feature selection as filter methods [53, 54, 55, 56, 57]. This method works by constructing linear combinations of the original variables using principal components (PCs). However, one disadvantage of PCA is that the interpretation of the reduced features in the new transformed space in relation to the original variables may not be straightforward. To deal with this problem, Mao recently presented an algorithm to link the PCs back to a subset of original features

by applying the LSE (Least-Square Estimation)-based evaluation [57]. The number of PCs and the threshold of LSE have to be determined to stop a searching process.

In this paper, we present a hybrid approach for unsupervised feature selection and sample clustering. The PCA based feature selection within a wrapper framework is called PFSBEM (hybrid PCA based Feature Selection and Boost-Expectation-Maximization clustering). PFSBEM is composed of three steps. The first step retrieves feature subsets with original physical meaning based on their capacities to reproduce sample projections on PCs. Different from Mao's work where a common feature subset was selected for all selected PCs, here we retrieve separate feature subsets with respect to individual PCs. The second step then decides the most important PCs in terms of maximizing clustering performance. To improve the quality of partitioning, we develop a new ensemble boost-EM-clustering approach based on boosting [58] and cluster validity index [59]. Finally, to further remove redundant features from the merged feature subsets of selected important PCs, a final feature subset is found utilizing boost-EM-clustering and cluster performance. Our experiment results show that redundant features that may be detected by PCs are effectively spotted by running a wrapper boost-EM clustering with SFS search strategy.

The remainder of the paper is organized as follows: First, the PCA based feature selection method using the LSE-based forward selection is briefly introduced. Subsequently, we present the new boost-EM clustering algorithm. Then, we introduce the PFSBEM hybrid approach for unsupervised gene selection and clustering. The new feature selection strategy will be presented. The comparison experimental results on widely-used microarray data sets of colon cancer and leukemia as well as synthetic data and machine learning benchmark data sets will be presented.

3.2 Methods

3.2.1 Feature selection based on principal components analysis

Principal components analysis (PCA) [41] is used to find a subspace whose basis vectors correspond to the maximum-variance directions in the original measurement space. Given a random variable $X \in \mathcal{R}^n$ in the original n-dimensional space, the new random random varialbe $Y \in \mathcal{R}^d$ in the subspace is calculated as,

$$Y = Q_d^T X, \tag{3.1}$$

where $X = [x_1, x_2, ..., x_n]^T$, $Y = [y_1, y_2, ..., y_d]^T$, and $d < n$. The linear transformation matrix Q_d is of size $n \times d$ whose columns are the d orthonormal eigenvectors corresponding to the d largest eigenvalues of the covariance matrix $\sum_X$ for variable X.

The well-recognized limitation of PCA approach in variable dimension reduction is the loss of physical meaning in the transformed subspace. Mao (2005) presented an algorithm to link the principal components back to a subset of original features by applying the least-square-estimation (LSE) evaluation criterion[57]. A *common* feature subset was found for the d eigenvectors. In this paper, using the similar concept from Mao's work, we aim to find *distinctive* feature subsets with respect to the d eigenvectors. The main idea of our approach comes from the fact the same set of features may not contribute equally to the principal components transformation. In the following paragraphs we will introduce PCA based feature selection from Mao's work. Our new PCA based feature selection will be presented in Section II C.

Instead of constructing the linear transformation matrix Q_d using all the feature dimensions, a linear model using a subset of the features is defined as

$$\hat{Y} = P_d^T Z, \tag{3.2}$$

where P_d is a linear transformation based on a subset of features Z for random vector X, $Z = [z_1, z_2, ..., z_m]^T$, $z_j \in \{x_1, x_2, ..., x_n\}(j = 1, ..., m)$, $\hat{Y} = [\hat{y_1}, \hat{y_2}, ..., \hat{y_d}]^T$, $m < n$ and d is the number of principal components (PCs) to be considered. P_d is an $m \times d$ matrix which comes out of $n \times d$ matrix Q_d, which means the parameters of certain PC in P_d are the same as those of Q_d for selected features. To evaluate the importance of important variables in original matrix Q_d, the cost function for the i-th principal component is defined as the total square error between the estimated $\hat{y_i}$ and the actual value of y_i using all the features:

$$J_i = \frac{1}{N} \sum_{k=1}^{N} [y_i(k) - \hat{y_i}(k)]^2, \quad i = 1, 2, ..., d, \tag{3.3}$$

where $k = 1, 2, ..., N$, and N is the total number of samples. The final subset of features is discovered by minimizing the total cost for the d principal components. To select a subset of features, a sequential forward-selection searching strategy can be used starting from an empty set. For multiple PCs, the feature evaluation and selection are based on minimizing the total summed error, $J = \sum_1^d J_i$, for all PCs based on Eq. 3.3. A certain feature generating the minimum J will be sequentially added to the selected feature set. The significance of a feature then can be measured based on the error reduction after adding the feature to current subset. Obviously, the error decreases with the increase of the number of features. A pre-specified threshold of summed J_i can be used to stop the searching process.

Several critical issues exist in Mao's approach. The first one is how to select d, the number of PCs, to maintain the cluster separability in the unsupervised microarray data samples. Another concern is for multiple PCs, during the feature selection process, the same feature subset was used in contributing the different projected $y_i, i = 1, \cdots, d$. To attack these concerns, we present a wrapper based approach uti-

lizing different evaluation criteria for feature subset selection, and a unique feature subset contributing to certain principal component (PC) will be used.

3.2.2 A boost-EM clustering algorithm

3.2.2.1 EM clustering

The expectation-maximization (EM) algorithm [60] is an iterative optimization method for computing the maximum likelihood (ML) estimate of missing data problems. It breaks the missing data problem into two parts, Expectation (E) step and Maximization (M) step. The expectation step evaluates the posterior probability of the unknown variables using the current parameter values in the model and conditioned upon the observations. It then assigns the observed data fractionally to each cluster according to this probability distribution. The maximization step is to re-estimate the parameters given the new fractional assignment. The two steps are iterated until convergence. The EM algorithm finds a local maximum contingent on parameter initializations. When applied to data clustering, the EM algorithm tries to estimate the distribution of each cluster using a mixture of component density functions. A more comprehensive review of the EM algorithm can be found in McLachlan and Krishnan's book (1997) [61].

3.2.2.2 The boost-EM clustering

Different clustering algorithms or replications of the same algorithm with random initializations may generate different partition results for the same data set. To deal with this problem, Frossyniotis et al. originally presented a multi-clustering fusion method to specify a common partition through combining the results from sev-

eral runs of a clustering algorithm [62]. Later on, they introduced another method which is an ensemble clustering approach based on boosting [58]. Boosting is a general method to improve the performance of any learning algorithm. The original boosting classification algorithm works by continuously designing weak learners and the final classification decision of a test sample is based on the weighted sums of the outputs given by the component classifiers (weak learners) [63, 64]. Based on the boosting concept, Frossyniotis et al. showed that boosting a simple clustering algorithm can improve the quality of the partitioning by generating multi-clustering solutions [58]. As an extension of the boosting-clustering framework, we present how this can be applied to EM clustering to boost the performance.

As well-recognized, the EM algorithm is sensitive to parameter initialization and tends to trap to local minima. Boosting approach can be one of the solutions to improve the stability and reproducibility of the EM algorithm. We thus propose a new boost-EM clustering based on the general boost-clustering framework by Frossyniotis *et al.*. The algorithm is summarized in Fig. 3.1. We keep the same notations to conform to the consistency.

The Boost-EM algorithm iteratively uses various bootstrapped replicates of original data to generate multiple EM clusterings. The resampling process is randomly done with replacement based on sample weights. Clustering using finite Gaussian mixture models by EM algorithm is applied to partition the new training data set in every iteration. The posterior probability $h_{i,j}^t$ that sample $X(i)$ belongs to the jth component of the mixture model in iteration t is thus estimated, which is used to set the cluster hypothesis H_i^t. Cluster index from EM algorithm is adjusted based on the highest similarity to current aggregated clusters. The assembled clustering result H_{ag}^t up to iteration t for sample $X(i)$ is determined by the maximum aggregation

from weighted voting of hypotheses $h_{i,j}^t$ for all possible j-th cluster as presented in Equation (iii) in Fig. 3.1.

To evaluate the clustering quality (CQ) of sample $X(i)$ for partition H^t, we used an entropy-based measure index CQ_i^t based on the posterior probabilities for sample $X(i)$. The index takes a high value when $h_{i,j}^t$ is comparable for all clusters j. The larger the value of CQ_i^t, the worse the clustering quality. Overall clustering quality for all samples from EM algorithm is calculated as a pseudoloss ϵ_t representing weighted sample clustering quality (Equation (i) in Fig. 3.1).

Analogous to boosting classification algorithm, a set of sample weights is estimated over the clustering process. The weight of training sample $X(i)$ at iteration t is denoted w_i^t. Initially, all weights are set equally. During the iterative process, samples with poor clustering, i.e., high value of CQ_i^t, will gain more weight (Equation (ii) in Fig. 3.1). The consequent bootstrap data resampling enables the EM clustering to focus on those difficult samples. The final aggregate cluster hypothesis is denoted as H_{ag}^T.

3.2.2.3 Finding the number of clusters

As any clustering mechanisms, boost-EM clustering does not have knowledge of the number of clusters for the analyzed data. One standard approach would experiment a range of values for cluster number C, the final value is then determined as the most stable solution [7, 65]. In this paper, we use the cluster validity index $\mathcal{T}$ to determine the number of clusters for all data sets, irrespective of the underlying clustering technique [59]. The details of the index $\mathcal{T}$ will be explained in the next section.

3.2.3 The PFSBEM Algorithm

Our goal is to efficiently explore the most discriminatory genes (features) and to remove the redundant ones while simultaneously discovering the categorization of the unknown data. We refer to our approach as a hybrid filter-wrapper unsupervised feature selection that is based on a PCA feature selection within a wrapper framework. This new method, PFSBEM (hybrid PCA based Feature Selection and Boost-Expectation-Maximization clustering), is summarized in Fig. 3.2.

PFSBEM is composed of three major parts. The first part (steps 1-2 in Fig. 3.2) retrieves the best feature subsets for individual PCs (principal components). Due to the high dimensionality of original gene features, PCA (principal component analysis) is used here to capture general characteristics of input data without involving any clustering algorithm. Borrowing the concept of maintaining original physical meaning from reduced feature size [57], we apply feature selection using LSE (least square error) criterion as shown in Eq. 3.3. The distinction between our work and Mao's work is that instead of finding the same feature subset for multiple PCs (principal components), a unique feature subset with original physical meaning is found for corresponding PC based on its ability to reproduce sample projection. Therefore, we will obtain different feature subsets for different PCs. And the resulting PCs vectors will have different size during the transformation process to get components of projection $\hat{Y}$. Fig. 3.3 shows the illustrative concept of our methodology. To increase efficiency of searching, we use only PCs (number d in Eq. refEq.2) whose values are larger than the average of all the eigenvalues after PCA.

To determine the importance of individual PCs in sample clustering, the feature subsets (in the original space) which correspond to PCs are evaluated to find the best PCs which maximize the quality of clustering. This is the second part of PFSBEM carried out as step 3 in Fig. 3.2. Therefore introduced Boost-EM-Clustering is used

as the clustering paradigm. The feature subset from each PC is treated as a single feature group during the search process. The optimal number of PCs is evaluated based on the maximum value of index $\mathcal{T}$. Note that instead of working in the transformed PC space, the original feature subsets are used since clustering with the PCs does not necessarily improve, and often degrades cluster quality [66].

Since the significance of individual features for PCs is evaluated as a group in each PC's feature subset, there might be redundancy in the features merged from selected feature subsets of important PCs. The third part (steps 4 Fig. 3.2) of PFSBEM thus removes redundant, irrelevant or insignificant features. For the merged features from selected PCs, using certain search strategy, boost-EM-clustering, and cluster validity index $\mathcal{T}$, the best feature subset that maximize clustering performance will be found at the end. The best number of clusters and sample clustering will also be outputs of PFSBEM algorithm. For certain pre-labeled data, using the final selected feature subset, Step 5 in Fig. 3.2 exams the clustering accuracy of test data using cross-validation. Next we will briefly introduce the feature search strategy and evaluation criterion used in PFSBEM.

3.2.3.1 Search Technique

There are a variety of ways to explore the solution space to select features [41], [42], [43]. An exhaustive search strategy is unaffordable in most cases (2^n feature subsets, where n is the number of original features). Some simple methods are as SFS (Sequential Forward Selection) and SBS (Sequential Backward Selection). SFS adds one feature at a time which in association with the selected features maximizes the learning performance. In SFS, once a feature is retained, it cannot be discarded. Though SFS does not guarantee an optimal solution, it is computationally attractive

with $\mathcal{O}(n^2)$ complexity. In this paper, without loss of generality, we use SFS [41] as our feature search technique .

3.2.3.2 Feature Evaluation Criterion

The quality of the selected PCs and the final feature subset is evaluated based on their ability to achieve natural grouping of input data. Since there is no knowledge of the actual clusters in given data, certain clustering validity criterion is estimated using the clustering output. We explored several cluster validity indices such as $trace(S_w^{-1}S_b)$, Partition Index (SC), Separation Index (S), Xie and Beni's Index, Silhouette and index $\mathcal{T}$ (XB) [46], [67], [68], [59], and finally applied a recently developed index $\mathcal{T}$ due to its performance [59]. The index $\mathcal{T}$ is defined as follows:

$$\mathcal{T}(C) = \left(\frac{1}{C} \times \frac{E_1}{E_C} \times D_C\right)^p, \tag{3.4}$$

where

$$E_C = \sum_{k=1}^{C}\sum_{j=1}^{N} u_{kj} \parallel x_j - z_k \parallel, \tag{3.5}$$

$$D_C = \max_{i,j=1}^{C} \parallel z_i - z_j \parallel, \tag{3.6}$$

C is the estimated number of clusters, N is the number of samples, $U = [u_{kj}]_{C\times N}$ is a data partition matrix, i.e. $u_{kj} = 1$ when sample j is in class k, $U_{kj} = 0$ when data j is not in class k, and z_k is the center of the kth cluster. The cluster number C is selected when maximum value of index $\mathcal{T}$ is achieved.

Index $\mathcal{T}$ is a regulated output balanced by the number of clusters, within cluster scatter, and between cluster separability. As a multiplication of three factors, the first factor of the index $\mathcal{T}$ is $\frac{1}{C}$ which reduces the index $\mathcal{T}$ as C tends to increase for better data separation. The second ratio factor consists of E_1 that is constant

Table 3.1. Description of the experimental data sets

Datasets	# of samples	# of features	# of classes
Synthetic data1	500 (250 class1, 250 class2)	5	2
Synthetic data2	500 (125 class1, 125 class2, 125 class3, 125 class4)	5	4
Wine	178 (59 class1, 71 class2, 48 class3)	13	3
IRIS	150 (50 Iris-setosa , 50 Iris-versicolor, 50 Iris-virginica)	4	3
Colon	62 (40 tumor, 22 normal)	2000	2
Leukemia	72 (25 AML, 47 ALL)	7129	2

for a given data set, and E_C that decreases with increased C. This factor shows that formation of more numbers of clusters, which will be compact in nature, would be encouraged. The second factor compensates the leverage of increased C. The third factor, D_C, which measures the maximum separation between two clusters over all possible pairs of clusters, increases with the value of C. The power p to control the contrast between the different cluster configurations is set as 2.

3.3 Experiments

To test the performance of the proposed PFSBEM algorithm for unsupervised feature selection and sample clustering, we used two synthetic data, two machine learning benchmark data, and two microarry gene expression data sets. Table 3.1 gives a description of the data sets. In the following subsections, we will discuss them in detail.

For all the data sets, since the class label is known, we estimated the clustering error which is defined as the number of misclassified samples divided by the total number of tested samples. Here we used the error from multiple runs of ten-fold cross-

Table 3.2. Error rate and average number of features on synthetic data sets

Method \ Data	Synthetic datasets					
	2-Class			4-Class		
	% CV Error	# of features	# of classes	% CV Error	# of features	# of classes
EM	22.3 ± 15.3	5 ± 0.0	2 ± 0.0	22.2 ± 9.8	5 ± 0.0	4 ± 0.0
PFSEM	**10.3 ± 2.0**	**2.0 ± 0.30**	**2 ± 0.0**	**2.5 ± 3.1**	**2 ± 0.0**	**4 ± 0.0**
PFSBEM	**9.85 ± 2.9**	**2.0 ± 0.35**	**2 ± 0.0**	**2.4 ± 1.25**	**2 ± 0.0**	**4 ± 0.0**

validation (CV). Ten-fold CV randomly partitioned the data set into ten mutually exclusive folds (subsets). Each single fold was considered as the test set while the other nine folds were treated as the training set. Using feature subset estimated from the training set, clustering was carried out on the test data. Each test data sample was assigned to the cluster that most likely generated it and then compared with its true class label. An error rate is thus calculated for all the test data within one iteration of ten-fold CV. The overall error rate will be shown as average and standard deviation values from 30 runs of ten-fold cross-validations.

We will first present the results on the synthetic data followed by two benchmark machine learning data sets. Then we will discuss experiments on microarray data sets.

3.3.1 Synthetic data

For the two synthetic Gaussian mixture data sets, the first one consists of two clusters described by five features out of which three are noise features. The two clusters have covariance matrix, $\sum_1 = \sum_2 = I$ and mean vectors $\mu_1 = (0, 0)^T$ and $\mu_2 = (0, 3)^T$. There is considerable overlap between the two clusters, and the

Table 3.3. Error rate and average number of features on wine and iris data sets

Method \ Data	Benchmark machine learning datasets					
	WINE			IRIS		
	% CV Error	# of Features	# of classes	% CV Error	# of Features	# of classes
FSSEM-k-TR	12.4 ± 13.0	3.8 ± 1.8	3.6 ± 0.8	4.7 ± 5.2	2.7 ± 0.5	3.1 ± 0.3
EM	10.1 ± 16.3	13 ± 0.0	3.0 ± 0.0	3.4 ± 5.1	4 ± 0.0	3.0 ± 0.0
PFSEM	**6.40 ± 6.0**	**7.1 ± 0.74**	**2.4 ± 0.52**	**2.67 ± 3.5**	**2.9 ± 0.31**	**3.0 ± 0.71**
PFSBEM	**6.30 ± 1.9**	**8.0 ± 0.53**	**2.7 ± 0.5**	**2.65 ± 0.10**	**2.8 ± 0.32**	**3.2 ± 0.4**

three additional noise features increase the difficulty of the problem. The second synthetic data set has four clusters with means at $(0,0)^T, (1,4)^T, (5,5)^T$ and $(5,0)^T$ and identity covariance matrices. We also added three Gaussian normal random noise features to original two dimension feature space. For the two synthetic data sets, we generated $N = 500$ samples with equal proportion to each cluster.

The performance of EM clustering (using all 5 features) was calculated to see whether or not feature selection help in finding unknown partitions. Additionally, to evaluate the performance of the boost-EM clustering, the results of PFSEM (PCA based feature selection and expectation maximization) as well as PFSBEM (PCA based feature selection and boost expectation maximization) were also presented. The difference between PFSEM and PFSBEM is the clustering algorithm algorithm used. For PFSBEM, the proposed boost-EM-clustering in Fig. 3.1 was applied. For EM, PFSEM, and PFSBEM, the best number of clusters, C, is estimated by the maximum value of the validity index $\mathcal{T}$. As a preprocessing step, we standardized input data sets by dividing each element by the corresponding feature standard deviation.

Table 3.4. Error rate and average number of features on colon data set

[Colon] Method	**# of Features**	**% CV Error**		
		EM	**BEM**	**Kmeans**
Baseline	2000	41.2 ± 23.8	35 ± 18.3	51.5 ± 17.2
PCA	61 PCs (from 2000)	47.2 ± 19.2	33.3 ± 13.6	48.3 ± 20
Two way ordering	200	30.3 ± 15.1	36.6 ± 23.3	48.1 ± 17.2
	100	34.1 ± 28.3	30 ± 18.9	53.3 ± 15.8
	20	23 ± 23.1	28.3 ± 13.7	49.5 ± 25
PFSBEM	**21.1 ± 3**	**15.9 ± 9.8**	**14.4 ± 10.1**	**28.6 ± 16.3**

Tab. 3.2 shows the cross-validation (CV) error rate (%), number of features, and number of clusters, all represented as mean± standard deviation for the two synthetic data sets. Both PFSEM and PFSBEM successfully detected noise features. We can see the dramatic error rate change using traditional EM with no feature selection. The average estimated number of sample clusters is also presented. Figs. 3.4(a) and (b) show the scatter plots and clusters projected on the two dimensional features continuously detected by PFSBEM in one of the 30 ten-fold runs.

3.3.2 Benchmark machine learning data

To compare our algorithm with other unsupervised feature selection method, such as FSSEM-TR (Feature Subset Selection using Expectation-Maximization clustering TRace) by Dy and Brodley (2004) [69], we tested the two benchmark machine learning data sets, Iris data and wine data, as being used in the original paper. The Iris data contains three classes, four features, 50 samples for each class (total 150 samples), where each class refers to a type of iris plant. The second wine data set

Table 3.5. Error rate and average number of features on leukemia data set

[Leukemia] Method	**# of Features**	**% CV Error**		
		EM	**BEM**	**Kmeans**
Baseline	7129	25 ± 22.57	20 ± 17.1	50.15 ± 19.5
PCA	71 PCs (from 7129)	23.33 ± 22.29	20 ± 19.9	53.1 ± 19.32
Two way ordering	1000	38.33 ± 13.72	35 ± 16.0	50.66 ± 20.86
	200	33.3 ± 20.79	34 ± 19.1	51.25 ± 22.99
	100	28.3 ± 15.81	30 ± 10.1	48.11 ± 19.2
PFSBEM	**300 ± 50**	**18.0 ± 10.6**	**16.5 ± 9.8**	**30.1 ± 20**

contains results of chemical analysis of wines grown in different cultivars. It has three classes, 13 features and 178 samples (59 belong to class one, 71 are class two and the remaining 48 go to class three.)

The experimental results in Tab. 3.3 report the cross-validation (CV) error rate (%), number of features, and number of clusters all represented as mean±standard deviation. Clustering and feature selection by FSSEM-TR, PFSEM (principal based feature selection with EM), classical EM, and PFSBEM clustering were compared. Figs. 3.4(c) and (d) show the scatter plots and clusters projected on the two features chosen by PFSBEM (in one of the 30 ten-fold runs). Since the wine data has more than eight features selected, Fig. 3.4(d) shows the clustering visualization when Features 1 and 3 are used. Generally as shown in Tab. 3.3, clustering with our proposed principal based feature selection gave better results than without it. The performance of our method was better or comparable to FSSEM by Dy and Brodley (2004). Our algorithm tended to find more features than FSSEM. Both PFSEM and PFSBEM performed equally well in terms of average error rate and the number of

selected features. However, PFSBEM gave more stable results than PFSEM which is indicated by smaller values of standard deviations. The estimation of number of clusters reflects the prior knowledge of given data.

3.3.3 Microarray gene expression data: colon cancer

The first microarray data is on colon cancer [15] consisting of 40 tumor and 22 normal colon tissue samples. Expression values of 2000 genes were studied. As a preprocessing step, we standardized input data sets by dividing each sample by the corresponding feature standard deviation. No additional preprocessing was applied for outlier removal.

As an example of the PFSBEM algorithm insight, for one iteration of the cross-validation, 11 PCs were selected based on the average eigenvalues using the training data. The feature subsets size for the selected 11 PCs is $\{200, 51, 42, 1, 200, 67, 82, 24, 7, 15, 4\}$. After sequential feature selection (SFS), three out of 11 important PCs were selected based on clustering validity index, which is PC1 with 200 features, PC7 with 82 features, and PC8 with 24 features. From the merged 304 features (two are common), 25 features were selected with index $\mathcal{T}$ value of 2.0109e-005, zero error rate, and number of clusters as two.

As a comparison, we carried out experiments using both different feature selection methods and clustering schemes. Certain feature selection method is followed by different clustering algorithms. The comparison was done without feature selection (baseline), PCA, two-way ordering and PFSBEM. For clustering, traditional EM was tested as well as boost-EM clustering (BEM) and k-means. The combinations of different feature selection and clustering schemes are shown in Tab. 3.4.

The unsupervised feature selection via a two-way ordering of gene expression data [51] forces irrelevant genes towards the middle in the ordering and thus can be

discarded. This algorithm has three steps: (i) identify and discard irrelevant genes by two-way ordering; (ii) perform an initial clustering using remaining genes; (iii) select final set of genes using supervised method based on the cluster structure obtained in (ii). Since the result of Step (i) mainly affects the performance of clustering [51], we considered only this step (two-way ordering) to identify irrelevant genes. Clustering using the top 20, 100, and 200 was carried out. The baseline clustering is the situation when all genes (feature) were used in the clustering process.

In Tab. 3.4, similar as before, the cross-validation (CV) error rate (%) is the average error rate over 30 ten-fold CV runs. We used the generalization error from multiple runs of ten-fold cross-validation. During each ten-fold CV, feature selection (except in baseline situation) and clustering were performed on the training set, and an error rate was calculated on the test set.

For the colon data set, as seen from Tab. 3.4, the average error rates using boost-EM algorithm generally outperformed than the other two classical algorithms. In two-way ordering case, overall best accuracy was obtained using the top 20 features. In the baseline clustering, though boost-EM (BEM) was applied, the error rate is still 35 ± 18.3, which is much higher than 14.4 ± 10.1 of PFSBEM. When looking at the results from PCA feature reduction, we can see though a much reduced size of 61 PCs was used, the clustering performance is not that much of difference with baseline method. This obeys others' conclusion that typical PCA's advantage resides in feature reduction instead of sample separation [41]. Though much improved than baseline algorithm, two-way ordering still gives lower performance comparing to the proposed PFSBEM.

3.3.4 Microarray gene expression data: Leukemia disease

The second gene expression microarray data on leukemia study [70] was labeled as two classes, acute lymphoblastic leukemia (ALL) and acute myeloid leukemia (AML), which are subtypes of leukemia. The data set contains expression of 7,129 genes from 72 samples (47 ALL and 25 AML). Similar standardization as colon data was carried out.

As an example of the PFSBEM algorithm execution during one iteration of cross-validation on training data, 16 PCs were selected from 7129 based on the average eigenvalues. The number of feature subsets size for the 16 PCs is $\{200, 200, 40, 24, 14, 200, 57, 4, 183, 9, 26, 85, 124, 27, 17, 30\}$. Out of the 16 PCs, nine important ones were determined using SFS with $\mathcal{T} = 0.0123$ and the number of clusters as four. The number of merged features from the selected nine PCs is 825. Finally 240 features (genes) out of 825 were selected with number of clusters as two and 14.2875% error rate.

The results of final clustering using the proposed PFSBEM and other feature selection and clustering methods are listed in Tab. 3.5. Same as before, the cross-validation (CV) error rate (%) is the average error rate over 30 ten-CV runs (300 iterations).

For the leukemia data set, the clustering result by boosting EM has error rate of 20.0 ± 17.1 when all 7,129 genes were used. When using around 300 genes selected by PFSBEM, clustering error rates reduces to 17.0 ± 10.0. The other two mechanisms for discarding irrelevant genes such as PCA and unsupervised two way ordering gave lower performance, though the same boosting EM clustering was applied. As can be seen from Tab. 3.5, the average error rates of boost-EM algorithm always outperformed than classical EM and K-means algorithms.

While observing the cross-validation errors from Tables 3.4 and 3.5, we can see the large error deviations. This is related to the cross validation method applied, which may not be reliable due to possible "unfortunate" partition of the training and test data when the number of samples for our gene expression data is small. For example, for colon data set, we only have 22 normal samples versus 2,000 gene feature dimension. It is possible that majority of the 22 normal samples are allocated in the test set while as the training folds are dominated by the cancer samples.

An another important output of our algorithm is the estimated number of clusters for the data sets. For both colon and leukemia data sets, the prior knowledge on the number of clusters for both data sets was two categories. However, from our clustering results, the number of categories is 2.8 ± 0.3 and 2.4 ± 0.28 for leukemia and colon data, respectively. This indicates the possible re-categorization of disease taxonomy, which is worth future exploration using large data sets.

To see whether or not the selected genes have high correlation with t-value, we calculated the individual t-values. Fig. 3.5(a) shows the t-values of the 40 merged genes chosen from 30 ten-fold CV runs of PFSBEM algorithm. Fig. 3.5(c) displays t-values of the 350 chosen genes for leukemia data. As a comparison, Figs. 3.5(b) and 3.5(d) show the t-values of the top 40 and the top 350 genes with respect to colon and leukemia data obtained from two-way ordering. We can observe genes selected by PFSBEM have larger chance of high correlation with t-value, while genes obtained by unsupervised two way ordering have relatively low correlation with t-value.

3.4 Conclusion and Discussion

In this paper, we presented an unsupervised gene selection and clustering algorithm named PFSBEM (hybrid PCA based Feature Selection and Boost-Expectation-Maximization). The algorithm first retrieves feature subsets based on their capacities

to reproduce sample projections on principal components. It then searches for the best PCs that maximize clustering performance. From the merged subsets of features which mostly contribute to deciding data projections on selected principal axes, PFSBEM finally finds the best features by using a validity function that utilizes the clustering results.

For high-dimension data set, due to the simplicity and ease to implement, PCA has been widely used as a preprocessing technique for feature reduction in unsupervised learning. To maintain the original physical meaning of selected features, based on the original concept of LSE (least-square estimation) from Mao [57], we developed a new PCA hybrid feature selection. Our approach comes from the idea that during the principal component (PC) transformation, genes which gives a great impact on major PCs, are mostly informative, and different sets of genes may contribute unequally to different PCs. Thus different gene sets corresponding to individual PCs were selected. Furthermore, the important PCs contributing most to data clustering are selected during the clustering process using clustering validity index $\mathcal{T}$. The original feature subsets selection for PCs falls into the diagram of filter approach, while as the afterward discriminative PC selection and final feature subset culling are wrapper approach. Therefore, we call our methodology a hybrid approach.

This paper also explored improving the performance of generic EM clustering algorithm using boosting learning. Our experimental results from examining boost-EM algorithm showed that boost-EM algorithm gave more stable results than standard EM algorithm. From the comparison study with other feature selections, the results revealed that incorporating PCA into the wrapper based feature selection led to better peformance to improve the class prediction. In analysis of gene expression data, we used the known class information to assess the performance of the proposed PFSBEM algorithm. The clusters obtained using unsupervised feature

selection differed somewhat from the human assigned labels but were reasonably well separated in both colon and leukemia datasets.

[Input] n-dimensional N samples X(1), …, X(N), X(i) ∈ R^n, i=1,...,N
The EM algorithm,
The number of clusters C,
The maximum number of iterations T.

Initialization

Set $w_i^1 = 1/N$ for i=1,...,N

Set t=1, ic = 0 and $\varepsilon_{\max} = 0$

Iterations

While $t \le T$ do

•Produce a bootstrap replicate of the original dataset according to the probability w_i^t for every sample i by resampling with replacement from the original dataset.

•Perform the EM algorithm, where $h_{i,j}$ is the posterior probability that X(i) belongs to the jth component of the mixture.

•Set the cluster hypothesis $H_i^t=(h^t_{i,1}, h^t_{i,2},..., h^t_{i,C})$ for all i, i=1,...,N.

•If $t > 1$, renumber the cluster indexes of H^t.

•Calculate the pseudoloss $\varepsilon_t = \frac{1}{2}\sum_{i=1}^{N} w_i^t CQ_i^t$, where $CQ_i^t = -\sum_{j=1}^{C} h_{i,j}^t \log(h_{i,j}^t)$. (i)

•Set $\beta_t = \frac{1-\varepsilon_t}{\varepsilon_t}$

•Check the stopping criteria:

•If ($\varepsilon_t > 0.5$) {$T = t$-1; break;}

•If ($\varepsilon_t < \varepsilon_{\max}$) {$ic = ic$ +1; if (ic==3){T=t; break;}

else {ic=0; $\varepsilon_{\max} = \varepsilon_t$ }

•Update distribution W: $w_i^{t+1} = w_i^t \beta_t^{CQ_i^t}$ and normalize W. (ii)

•Compute the aggregate cluster hypothesis $H^T_{ag.}$

$$H_{ag}^{t} = \arg\max_{k=1,...,C} \sum_{\tau=1}^{t} \left[\frac{\log(\beta_\tau)}{\sum_{j=1}^{t} \log(\beta_j)} h_{i,k}^{\tau} \right] \qquad \text{(iii)}$$

•$t = t$+1

End while.

Return the final cluster hypothesis H^T_{ag}.

Figure 3.1. The new boost-EM clustering algorithm.

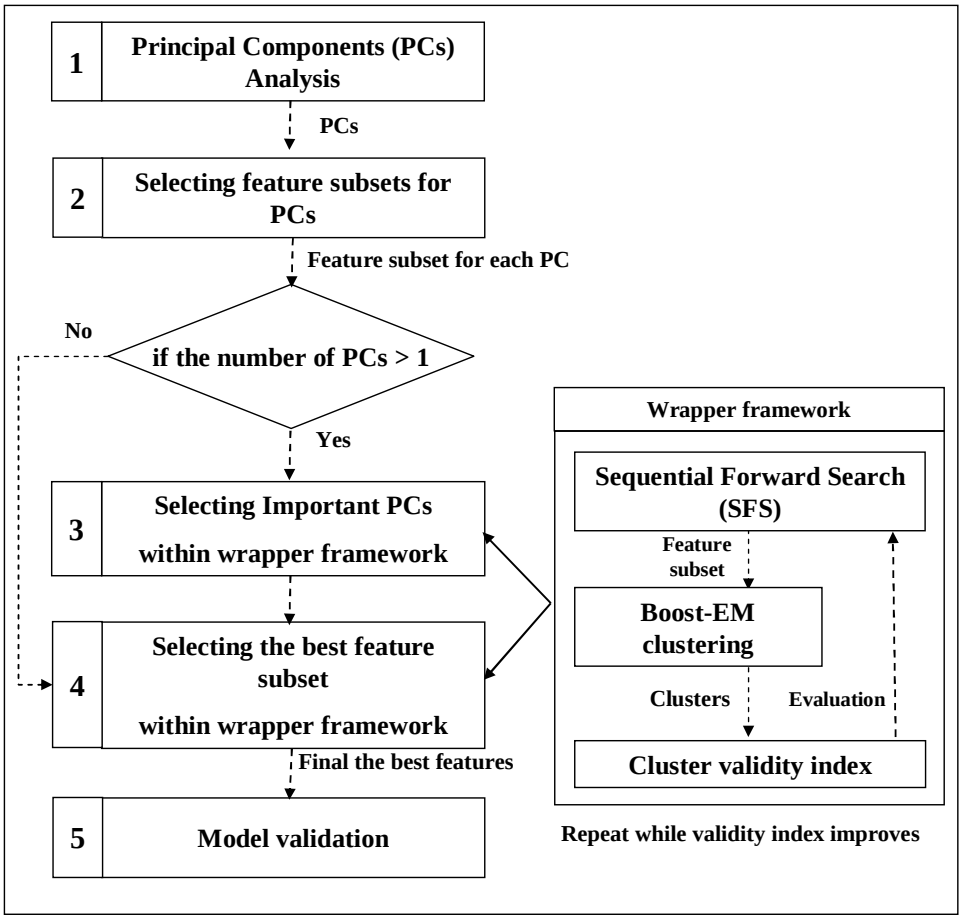


Figure 3.2. Outline of the PFSBEM algorithm.

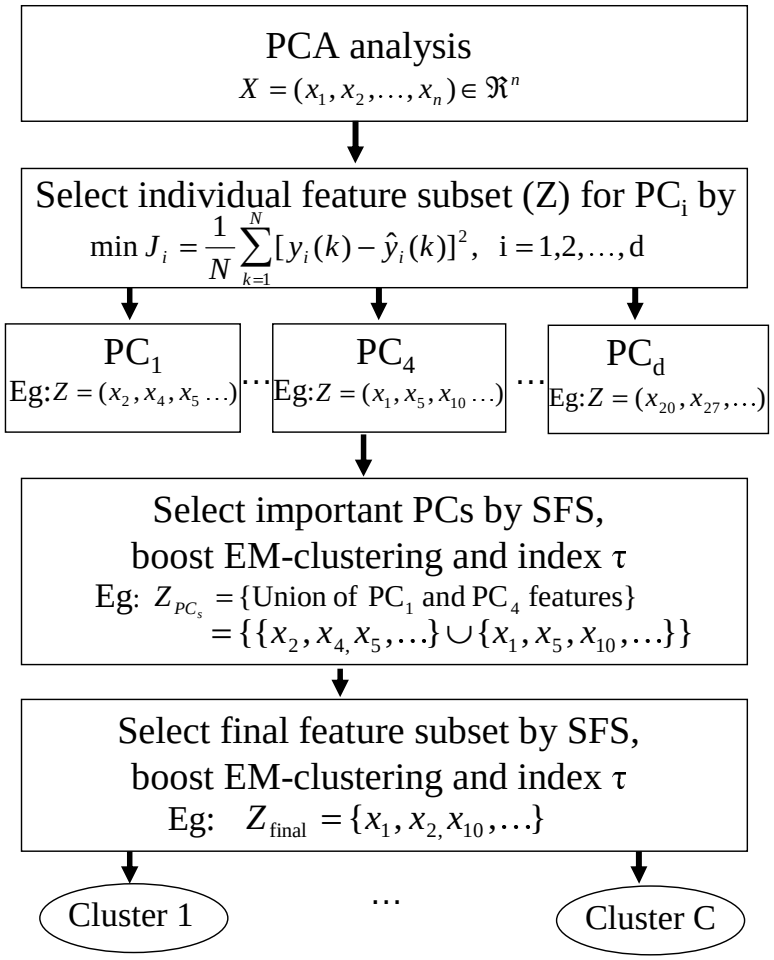


Figure 3.3. Illustrative flowchart for PFSBEM algorithm.

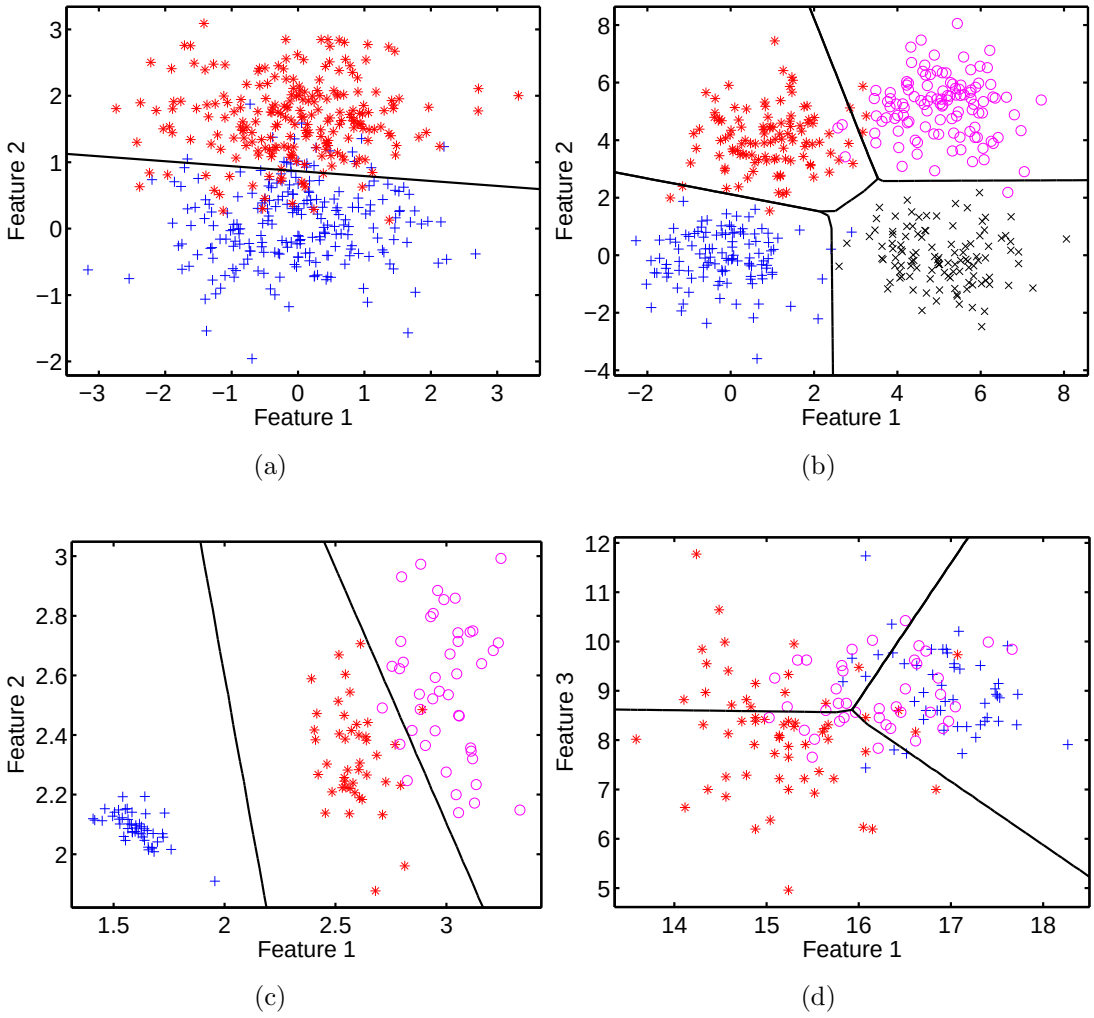


Figure 3.4. The scatter plots on (a) 2-class synthetic data, (b) 4-class synthetic data, (c) Iris data, and (d) wine data using two features chosen continuously by PFSBEM. $\circ$,+, $\times$ and $\star$ represent the different cluster assignments. Areas correspond to the clusters discovered by PFSBEM.

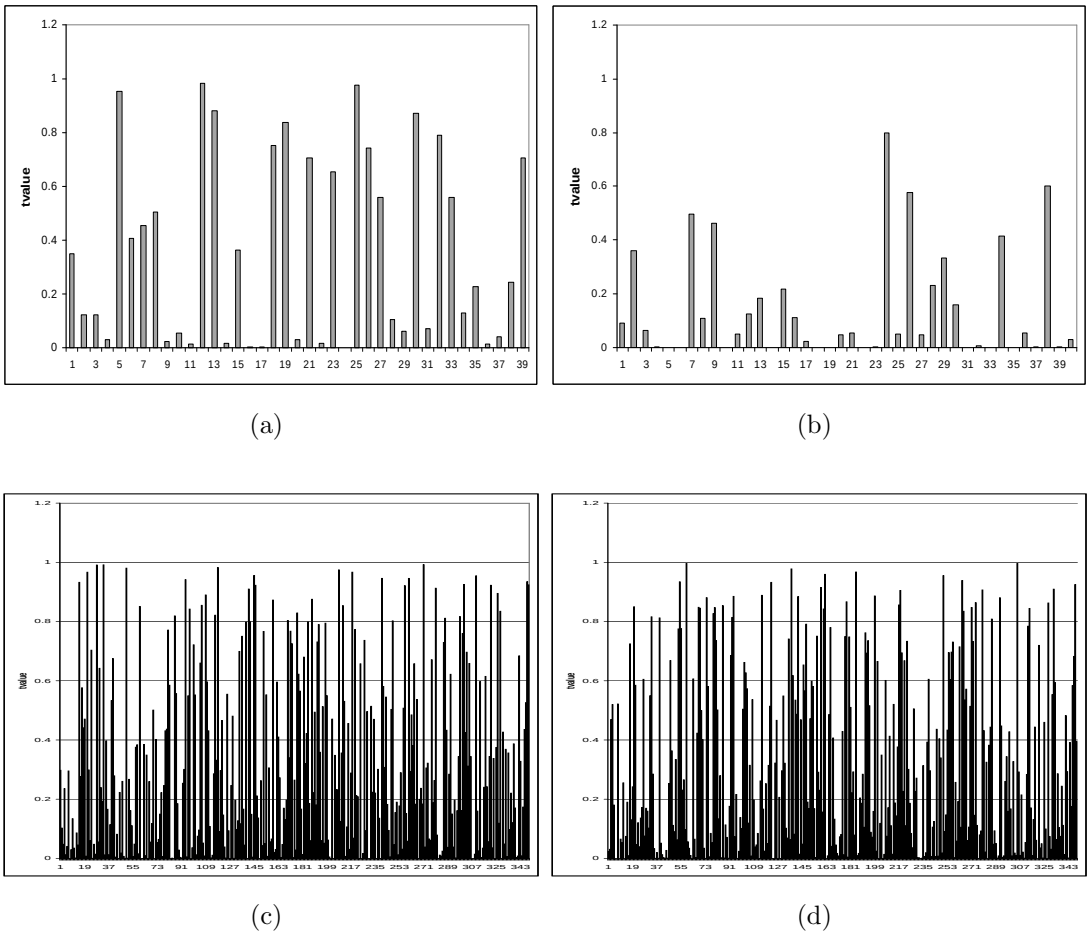


Figure 3.5. t-values (with two-tailed distribution) of selected genes (a) by PFSBEM for colon data, (b) by unsupervised feature selection via two way order ordering for colon data, (c) by PFSBEM for leukemia data, and (b) by unsupervised feature selection via two way order ordering for leukemia data. For (a) and (c), genes selected from 30 ten-fold runs were merged into one final feature set (40 for colon and 350 for leukemia). For the two-way ordering, genes near the two ends were selected (40 for (b) and 350 for (d)).

CHAPTER 4

A NEW MAX-RELEVANCE CRITERION FOR SIGNIFICANT GENE SELECTION

4.1 Introduction

In microarray gene expression analysis, identifying the most representative genes (or features) from tens of thousands of genes in experiments, is critical to improve the prediction performance. Feature selection is one of the important and frequently used techniques in data preprocessing for microarray analysis.

There are three general approaches for feature selection algorithms: filters, wrappers [71] and hybrids [72]. Filter approaches use general characteristics of data to select a subset of features without involving any induction algorithm. Wrapper approaches use estimated accuracy of learning method to obtain feature subsets. Generally, wrapper approaches give higher prediction accuracy but they tend to be more computationally expensive than filter approaches. Hybrid approaches try to utilize different evaluation criteria of two approaches in different search stages. In this paper, we focus on the discussion of filter type feature selections which have better generalization property and can be computed easily and efficiently.

The goodness of feature subset is always determined by a certain criterion. Both Max-Relevance and Min-Redundancy have been instinctively used for this criterion. Max-Relevance is to search features which together have the largest correlation to the target class. Some methods based on statistical tests or information gain have been shown in literature [73], [74]. However this criterion could allow rich redundant genes, which jointly do not contribute to the performance of the prediction

because they are highly correlated. Min-Redundancy is a criterion to select mutually exclusive features. An effort to reduce "redundancy" among genes has been recently made in gene selection. Some recent methods propose a criterion by combining the above two constraints effectively [75], [76], [77].

Our work in this paper focuses on maximizing Max-Relevance by considering the joint effect of features (or subspace) on the target class. Methods which have been used for Max-Relevance do not consider the dependency between interactions among features and the target class. However, this may be critical in many circumstances. Based on this fact, we propose a new Max-Relevance criterion, combining the emerging pattern (EPs), one of the recent data mining techniques used to identify interactions among features, [8], [4], [5], and the currently used techniques.

The main contribution of this paper is to show the usefulness of employing interactions among features to explicitly maximize relevancy in feature selection via filter approach. Our comparative experiments demonstrate that the proposed method not only gives higher accuracy than other criteria but also provides comprehensive explanation about relevancy and redundancy of features.

4.2 Methods

4.2.1 Maximum Relevance

The aim of Max-Relevance is to find features which mutually have the largest correlation to the target class. In developing an approximation method for Max-Relevance, our goal is to effectively catch the joint effect of features on the target class. For this, we employ emerging patterns which have the strong power of modeling interactions among features. The most used notions are defined as follows:

Definition 1. The jumping emerging patterns (JEPs) in dataset of Class$_+(C_+)$, denoted $JEPs(C_+)$, are patterns (P) whose supports in C_- are zero but non-zero in C_+. + and - stand for two class labels.

Definition 2. The most expressive $JEPs(C_i)$, denoted $ME-JEPs(C_i)$, are subsets which have the largest supports of all $JEPs(C_i)$.

Definition 3. The most expressive features of Class$_i$, denoted $ME-features(C_i)$, are features within $ME-JEPs(C_i)$.

Definition 4. The collective impact of $ME-features(C_i)$, denoted $D_{C_i}(F)$, is defined as

$$D_{C_i}(F) = \sum_{P} Supp_{c_i}(F), \tag{4.1}$$
$$F \in \text{ME-features}(C_i), P \in \text{ME-JEPs}(C_i),$$

where $Supp_{c_i}(F)$ is the frequency of occurrence of features(F).

In our approach, we adopt both parametric and nonparametric approach to select discriminative features: t-test (or f-test for multiple classes) and symmetrical uncertainty (SU). The t-test is a statistic criterion based on the assumption that data comes from some kind of distribution, while SU based on the information-theoretical concept of entropy is a measure of the uncertainty of a random variable. SU is more used than information gain because it can make good for information gain's bias toward features with more values. In some papers, as the combination of several criteria often outperforms an individual criterion, we attempted to take advantages of

both criteria. (e.g., the information gain can compensate for the statistical instability of t-test).

The t-test gives the discriminative power of the ith feature as

$$T(F_i) = \frac{|\mu_i^+ - \mu_i^-|}{\sqrt{\frac{(\sigma_i^+)^2}{n^+} + \frac{(\sigma_i^-)^2}{n^-}}}, \tag{4.2}$$

where μ_i^+ and μ_i^- are the means of C_+ and C_- for F_i feature, respectively; σ_i^+ and σ_i^- are the corresponding standard deviations; n^+ and n^- indicate the number of samples contained in each class.

The SU is defined as

$$SU(F_i, C) = 2\Big[\frac{IG(F_i|C)}{H(F_i) + H(C)}\Big], \tag{4.3}$$

where,

$$H(F) = -\sum_i P(f_i)\mathbf{log_2}(P(f_i)), \tag{4.4}$$

$$H(F|C) = -\sum_j P(c_j)\sum_i P(f_i|c_j)\mathbf{log_2}P(f_i|c_j), \tag{4.5}$$

$$IG(F|C) = H(F) - H(F|C), \tag{4.6}$$

P(f_i) is the prior probabilities for all values of F_i, and P($f_i|c_j$) is the posterior probabilities of F_i given the values of C.

In both criteria, the more F_i and C is correlated, the larger the result value is (e.g. in SU, if F_i and C is completely correlated, SU(F_i, C) is 1). We use average ranks between above two ranks as follows:

$$Rank_M(F) = \text{AVG}\left(Rank_{t-test}(F), Rank_{SU}(F)\right), \tag{4.7}$$

where the lower the number of Rank$_M$ is, the stronger the discrimination power is (e.g. the most discriminative feature is the feature whose Rank$_M$ is 1).

Finally, by combining the collective impact of features in $ME-JEPs$ and the merged rank of well-known criteria, our new MAX-Relevance criterion is defined as

$$\begin{aligned} \max D(F) &= argmax D(F) \\ &= \frac{1}{2}\left[\frac{1}{N_k}\sum_{C_i}^{K}\frac{D_{C_i}(F)}{\max(D_{C_i}(F))}\right], \\ &+ \frac{1}{2}\left[\frac{N_f - Rank_M(F)+1}{N_f}\right], \end{aligned} \tag{4.8}$$

where N_k is the number of classes and N_f is the number of features.

4.2.2 Minimum Redundancy

The minimum redundancy condition may be defined in several ways [76], [77], [78], [79]. We use Pearson correlation coefficient which is the most well-known measure of similarity between two random variables. The condition is defined as

$$\min R = argmin R = \frac{1}{N_f^2}\sum_{i,j}|c(i,j)|, \tag{4.9}$$

$$where, \qquad c(i,j) = \frac{cov(i,j)}{\sqrt{var(i)var(j)}}, \tag{4.10}$$

$var(\cdot)$ denotes the variance of a variable and $cov(\cdot)$ represents the covariance between two variables. N_f is the number of features. And we have assumed that both high positive and high negative correlation mean redundancy, and thus take the absolute value of correlations.

4.2.3 The Minimum Redundancy Maximum Relevance

Ding and Peng proposed the minimum-redundancy-maximum-relevance (mRMR) criterion to minimize redundancy [75]. The idea is to select the genes such that they are mutually maximally dissimilar. The mRMR criterion ($\Phi(D,R)$) has the follow-

Table 4.1. Different conditions to search for the next feature

Acronym	Full Name	Formula
t-test	t-test	$\max_{i\in F}[T(i)]$, (T(i) in Eq.(2))
TCD	t-test correlation difference	$\max_{i\in F}\left[T(i)-\frac{1}{N_f-1}\sum_j \lvert c(i,j)\rvert\right]$
EPMRCD	EPs and merged rank correlation difference	$\max_{i\in F}\left[D(i)-\frac{1}{N_f-1}\sum_j \lvert c(i,j)\rvert\right]$, (D(i) in Eq. (8))
MRCD	merged rank correlation difference	$\max_{i\in F}\left[\frac{N_f-Rank_M(i)+1}{N_f}-\frac{1}{N_f-1}\sum_j \lvert c(i,j)\rvert\right]$, (Rank$_M$(i) in Eq. (7))

ing simplest form to optimize D (relevance condition) and R (redundancy condition) simultaneously.

$$\max \Phi(D,R), \Phi = D - R. \tag{4.11}$$

In this paper, we employ this framework because this is a very simple but efficient method [77]. So, based on Eq. (4.4) and Eq. (4.5), our minimum-redundancy-maximum-relevance optimization condition is defined as

$$\max_{F_i\in F} D(F_i) - \frac{1}{N_f-1}\sum_j |c(F_i,j)|. \tag{4.12}$$

4.3 Experiments

We used the well-known datasets, the colon tumor set of [15] and the Leukemia set of [70] to demonstrate the robustness of our new approach. The colon data set contains 40 tumor and 22 normal colon tissue samples of 2,000 genes with highest minimal intensity across the samples. In the leukemia dataset, the target classes are AML and ALL which are subtypes of leukemia and there are 72 samples of 7,129 genes. For the leukemia data, we merged training and test samples together for the purpose of leave-one-out cross validation. In our experiments, we used two different formats as input. We first discretized the data using the entropy based discretization

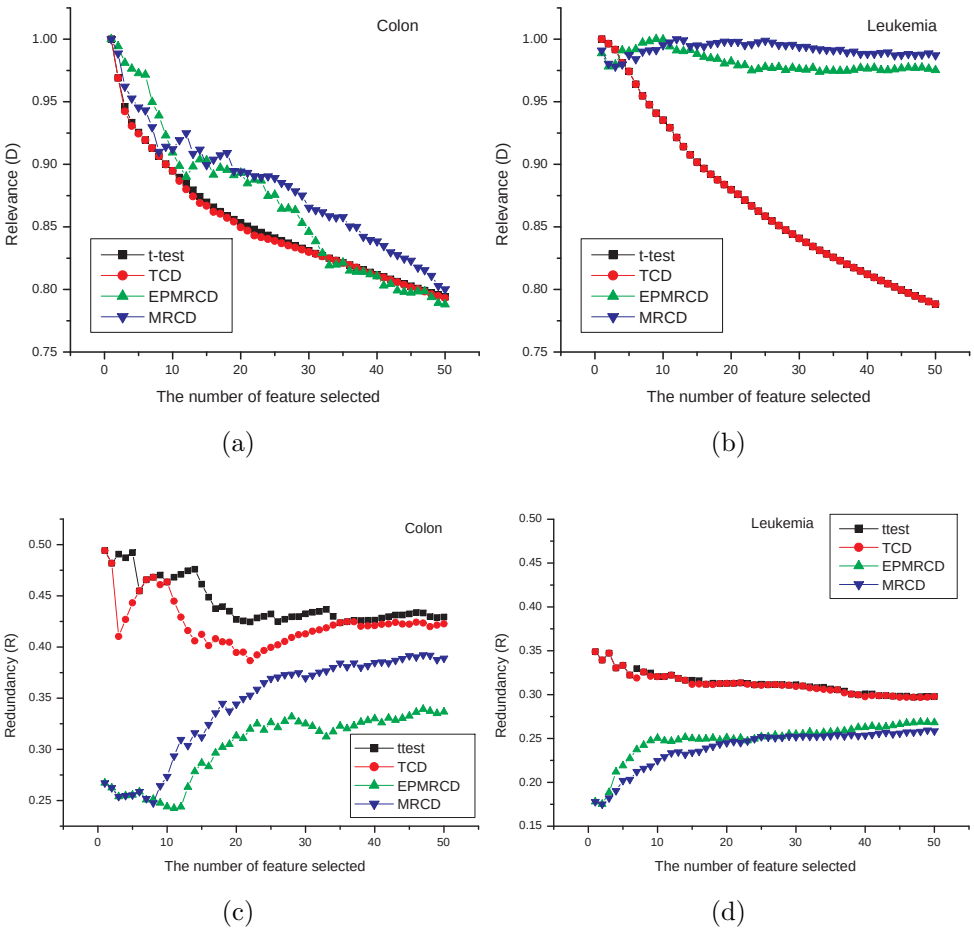
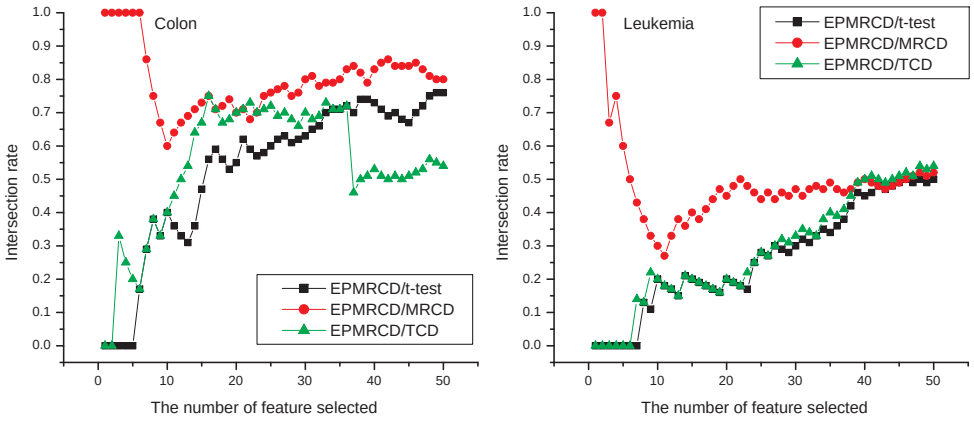


Figure 4.1. (a) Relevance on Colon dataset, and (b) Relevance on Leukemia dataset, and (c) Redundancy on Colon dataset, and (d) Redundancy on Leukemia dataset.

method [10]. This preprocessing step efficiently explores the most discriminatory features with EPs algorithm as well as removes many of the noisy features. Then we normalized the original data so that each gene has zero mean value and unit variance and classified them using SVM. 132 genes in colon dataset and 1026 genes in leukemia dataset were used after discretization.

We measured the classification error rate using Leave-One-Out Cross Validation (LOOCV) to compare the results with Ding and Peng's [75]. Given n samples, LOOCV method constructs n classifiers, where each one is trained with $n-1$ sam-



(a) (b)

Figure 4.2. Intersection of features selected using different conditions. (a) Colon dataset (b) Leukemia dataset.

Table 4.2. LOOCV errors of colon and leukemia datasets

		The number of features (top m features)													
Data	Method	1	2	3	4	5	6	7	8	10	15	20	30	40	50
Colon	t-test	14	10	9	11	10	9	9	9	10	10	13	10	9	8
	TCD	14	10	8	7	7	7	6	7	8	8	8	8	13	14
	EPMRCD	9	10	9	9	9	10	7	7	**6**	8	8	7	7	7
	MRCD	9	10	9	9	9	10	10	10	8	7	8	7	7	7
Leukemia	t-test	9	3	2	2	2	3	3	4	2	3	3	3	4	1
	TCD	9	3	2	3	3	3	2	4	2	3	5	1	1	1
	EPMRCD	12	6	7	5	3	5	3	2	2	2	1	**0**	**0**	**0**
	MRCD	12	6	6	7	2	5	4	4	3	5	2	1	1	2

ples, and is tested with the remaining one sample. The final classification accuracy is the average of each classifier.

In our experiments, we compared feature subsets by using three different mRMR optimization conditions in Tab. 4.1 against the feature sets obtained using t-statistic ranking to pick the top m features. We referred the results of t-test and TCD in [75] to demonstrate the robustness of our proposed criterion. The rea-

son to select TCD (t-test correlation difference) instead of TCQ (t-test correlation quotient) is that TCD is the same scheme as ours.

The results of the LOOCV error are shown in Tab. 4.2. Generally, EPMRCD (EPs and merged rank correlation difference) features outperformed other features. For instance, for colon, EPMRCD leads to 6 errors while t-test leads to 8 errors and MRCD (merged rank correlation difference) leads to 7 errors. And for leukemia, EPMRCD leads to 0 errors while t-test leads to 2 errors and TCD leads to 1 error. However, LOOCV classification error does not provide enough evidence for our efficient criterion.

To demonstrate the effectiveness of our proposed approach, we showed the average relevance (D) and the average redundancy (R) of feature sets in Fig. 4.1 (refer to Eqs. (4.7) and (4.8)) . For colon, although the relevance for EPMRCD reduced as compared to the others, the redundancy also reduced dramatically. Note that both t-test and TCD feature sets show relatively high redundancy. This is more clearly observed in leukemia. For leukemia, the relevance of EPMRCD least reduced relative to others, while the redundancy reduced considerably. In the case of t-test feature set, even relevance reduced impressively according to increase the number of features and within top 10 features, it also shows relatively high redundancy. This results show that the EPMRCD feature set is the most effective one satisfying the Max-Relevance-Min-Redundancy condition.

In order to show how different the EPMRCD feature set is from other features, we also present the rates of intersecting features for the top m ($1 \leq m \leq 50$) features selected as shown in Fig. 4.2. Features selected using EPMRCD have less overlap with those selected by using t-test or TCD when $m \leq 20$, while they are frequently found in the features selected via MRCD when $m \leq 5$.

Above experiment results demonstrated that even though LOOCV classification error rates were comparable, our criterion found great features, which are dissimilar to those selected by other criteria and are sufficiently satisfying the Max-Relevance-Min-Redundancy condition.

4.4 Conclusion

In this paper, we presented a new Max-Relevance criterion applied to the minimum redundancy-maximum relevance (mRMR) framework. This criterion is independent of class prediction methods, and thus does not guarantee the best results for any prediction method. The main benefit of proposed criterion is to capture the class characteristics in a broader scope by identifying the joint effect of features and reducing mutual redundancy within the feature set at the same time. Our experiment results showed that proposed criterion generated features which have better generalization property and improve prediction. For example, we achieved 100%, 90.32% LOOCV accuracy in leukemia and colon, respectively, even though we just used the top m features without considering any kind of selection mechanisms. These features also were sufficiently satisfying the Max-Relevance-Min-Redundancy condition relative to other criteria on the same mRMR framework. In the future work, we will apply the EPMRCD feature selection method on multiclass datasets using several prediction methods and verify that it can outperform consistently regardless of the class prediction methods and the number of classes.

CHAPTER 5

FUNCTIONAL PROTEOMIC PATTERN IDENTIFICATION UNDER LOW DOSE IONIZING RADIATION

5.1 Introduction

The exposure to low dose (10 CGy or lower) ionizing radiation (IR) occurred to nuclear plant works, astronauts, and X-ray operators affects several signaling pathways including DNA damage, DNA repair, cell cycle checkpoints, and cell apoptosis. To understand the possible molecular signaling pathways thus affected, we study the dynamic responses of the networks under different patterns considering both time and dosage changes. An emerging protein microarry called revers-phase protein microarray (RPPM), in conjunction with the quantum dots (Qdot) nano-technology, is used as the detection system. This technology (RPPM-Qdot) offers us the ability to monitor the time series and dosage responses of cells exposed to low dose radiation.

Different from the matured gene microarray technology, protein microarray is a new technology. RPPM is a quantitative assay much like a miniature "ELISA-on-a-chip" platform. In contrast to other protein arrays that immobilize probes, RPPM immobilizes the whole repertoire of sample proteins. It allows numerous samples to be analyzed in parallel using only minute (nanoliter) amounts of sample for making quantitative measurements to profile changes in activity of different candidate signaling molecules in cell lines [80]. The RPPM technology was especially designed by our lab for profiling changes in protein activity (e.g. phosphorylation, cleavage activation, etc.) rather than just protein expression levels. The marriage of RPPM with quantum dots (Qdot) nano-technology due to its high yield of bright fluorescence and

resistance to bleaching offers us an innovative detection technique. Therefore with RPPM-Qdot, we are able to elucidate ongoing kinase activities and post translational modifications to generate a dynamic view for the functional proteomic analysis.

Isogenic human Ataxia Telangietctasia (A-T) cells are employed to study the central role of ATM (ataxia-telangiectasia mutated) in the cellular response to ionizing radiation. Cellular phenotype of A-T cells showed defects in ATM signal transduction and hypersensitivity to IR [81],[82]. ATM is a DNA double strand break (DSB) sensor and can be activated by change of chromatin structure. It plays a pivotal role in both cell cycle arrest and DNA repair. A-T cells therefore provide a great model for the studies of DNA damage responses induced by low dose IR.

For the data output from the Qdot-RPPM technology under different dosages and at different time points, to quantitatively determine the responsive protein/kinases and to discover the pathway motifs formed by them, visual inspections are not always obvious or accurate. Sophisticated computational algorithms have to be explored to robustly discover and identify these complicately expressed molecular patterns and their interactions. While a lot of research deals with classification methods in applications to gene microarray data, only a few approaches are explicitly designed to consider the dependence relationships among the investigated features (proteins). Hence, to capture the global picture of the signaling pathway, the dependence among proteins/kinases needs to be taken into account [83]. Feature pattern (combination of features) identification techniques should be used to provide more underlying semantics than single features. Nevertheless, it is very difficult to find meaningful patterns in large datasets like microarray data because of the huge search space. The difficulty also comes from the existence of infrequent network patterns that exist but are often irrelevant or do not improve the accuracy of network finding [84, 85].

To identify the different proteins/kinases involved in the signaling pathways for low vs. high dose ionizing radiation for ATM cells, we designed a Discriminative Network Pattern Identification System (DiNPIS). Instead of simply identifying proteins contributing to possible pathways, this methodology takes into consideration of protein interaction and dependence that are represented as Strong Jumping Emerging Patterns (SJEP) and infrequent patterns though occurred will be considered irrelevant. The whole framework consists of three steps: feature (proteins, kinases) [1] selection, network pattern identification, and network pattern annotation. For feature selection, the responsive proteins/kinases contributing most to distinguishing dosage and temporal difference will be identified. The network motifs of those selected proteins are discovered by employing SJEP pattern mining using a contrast pattern-tree. The last step of network pattern annotation provides a complete protein pattern characterization such as individual protein significance, protein dependence measurement, and network motif significance under IR. In the following sections, we will describe the system in detail.

5.2 Methods

5.2.1 Support Vector Machines (SVMs)

We begin with a quick summary of the SVM classifier that is used for discriminative protein selection later on. SVMs are kernel based learning algorithms which have been successfully applied to numerous classifications and pattern recognition problems such as text categorization, image recognition and bioinformatics [86]. Suppose that there are n training samples, $(\mathbf{x}_1, y_1), \cdots, (\mathbf{x}_n, y_n)$, where $\mathbf{x}_i \in R^d$ is a d-dimensional feature vector representing the i^{th} training sample with class label

[1] Nomenclatures "feature" and "proteins/probes/kinases" are used interchangeably.

$y_i \in \{+1, -1\}$ for $i = 1, \cdots, n$. SVMs search for an optimal hyperplane which maximizes the margin between two classes. The hyperplane classifying an input pattern $\mathbf{x}$ can be described as the following function :

$$f(x) = \langle \mathbf{w}, \mathbf{\Phi}(\mathbf{x}) \rangle + b, \tag{5.1}$$

where $\mathbf{w}$ is a weight vector, b is a scalar, and $\mathbf{\Phi}(\mathbf{x})$ is a mapping function. We can compute the weight vector by solving a quadratic programming problem formulated to find the optimal hyperplane.

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i y_i \mathbf{x}_i, \tag{5.2}$$

where $\alpha_i \in [0, l], i = 1, \cdots, l$ are Lagrange multipliers and l is the number of support vectors.

5.2.2 Discriminative Network Pattern Identification

The finding of responsive proteins under ionizing radiation utilizes the concept of Emerging Pattern (EP) that reflects the support change of certain proteins from one data set to another one [8]. For each numerical attribute from RPPM, its value range is discretized into two or more intervals. Each "(attribute, continuous-interval)" pair is called an item. $(probe_cAbl, [0.9776, +\infty))$ is an example of items. Let I be the set of all items. Then a set X of items is called an itemset which is defined as a subset of I. $X(f_i)$ is defined as an itemset of the feature f_i which contains all continuous-interval items of the attribute f_i. For example, the discretization method partitions the probe into two disjoint intervals. $X(probe_cAbl) = \{(probe_cAbl, (-\infty, 0.9776)), (probe_cAbl, [0.9776, +\infty))\}$. $sp_D(X)$ is defined as the support of an itemset X in a data set D calculated by $count_D(X)/|D|$,

where $count_D(X)/|D|$ is the number of samples in D containing X. Suppose D contains two different classes: D_1 and D_2. For an item $i \in I$, there is a single itemset $\{i\} \subset I$. We define the importance of $\{i\}$ as Pattern Significance described below:

Definition 1. *(Pattern Significance) Given $\xi > 0$ as a minimum support threshold, the significance of an item $\{i\}$, denoted as $S(\{i\})$, is defined as*

$$S(\{i\}) = \begin{cases} 0 & \text{if } sp_{D_1}(\{i\}) < \xi \wedge \\ & \quad sp_{D_2}(\{i\}) < \xi, \\ sp_{D_2}(\{i\}) & \text{if } sp_{D_1}(\{i\}) = 0 \wedge \\ & \quad sp_{D_2}(\{i\}) \geq \xi, \\ sp_{D_1}(\{i\}) & \text{if } sp_{D_1}(\{i\}) \geq \xi \wedge \\ & \quad sp_{D_2}(\{i\}) = 0, \\ |sp_{D_1}(\{i\}) - sp_{D_2}(\{i\})| & \text{otherwise.} \end{cases}$$

The larger the significance of an item, the sharper the discriminating power associated with the item. If $S(\{i\}) = sp_{D_1}(\{i\})$ or $S(\{i\}) = sp_{D_2}(\{i\})$, we call an item $\{i\}$ as a SJEP (Strong Jumping Emerging Pattern) which is the shortest JEPs satisfying the support constraint. In fact, an item $\{i\}$ is the shortest SJEP. Let $J = \{j_1, j_2, ..., j_p\}$ be the set of all items appearing in $X(f_i)$, we have the following definition for feature significance as the combined significance of items for itemset $X(f)$:

Definition 2. *(Feature Significance) A significance measure S is a function mapping a feature $f \in F$ to a real value such that $S(f)$ is the degree of interestingness of the feature f. $S(f)$ is defined as $S(f) = \sum_{i=1}^{p} S(J(i))/|J|_{S(J)\neq 0}$.*

Given the significance measures of individual proteins, we can define the relative significance between two proteins, f_i and f_j. Let $J = \{j_1, j_2, ..., j_p\}$ be the set of all items appearing in $X(f_i)$ and $K = \{k_1, k_2, ..., k_q\}$ be the set of all items appearing in $X(f_j)$, the dependence relationship of these two proteins is defined as follows:

Definition 3. *(Relative Feature Significance) Given the significance measure S of two features f_i and f_j, the relative significance is defined as*

$$\begin{aligned} S(f_j|f_i) = & \left[\sum_{i=1}^{p}\sum_{j=1}^{q} S(K(j)|J(i))\right] \Big/ (|K| + |J|), \\ = & \left[\sum_{i=1}^{p}\sum_{j=1}^{q} S(K(j)) - R(J(i), K(j))\right] \Big/ \\ & (|K| + |J|), \end{aligned}$$

where $S(J(i)), S(K(j)) > 0$ and $R(J(i), K(j))$ denotes the redundancy between two patterns $J(i)$ and $K(j)$.

The relative feature significance between two features is calculated based on pattern significance and pattern distance by the minimum redundancy-maximum relevance (MRMR) framework [75]. The relative feature significance is used to identify the relationship between features and to reduce mutual redundancy within the feature set at the same time. However, the ideal redundancy measure $R(J(i), K(j))$ is hard to obtain. In this paper, we use approximated redundancy defined by distance between patterns [84]. The following equation is used to approximate R.

$$R(J(i), K(j)) = (1 - D(J(i), K(j))) \times \min(S(J(i)), S(K(j))). \tag{5.3}$$

The distance measure between two patterns can be obtained based on the pattern structure, or based on the distribution of the patterns, or based on the data used

Initialization
- Feature set: F = {f_1, f_2, …, f_r}

Iteration
- Discretize training samples
- Repeat until F=[ ]
  - Compute relative feature significance $S_i(\cdot|f_i)$ of f_i using Def. 1-3
  - Find the feature subset Sub_i with the highest significance score S_i
  - Train SVM with the training samples with features contained in Sub_i
  - Compute the weight vector **w** using Eq. (2)
  - Compute the weight W_i using Eq. (5)
- Compute the average of feature weights obtained from Sub_1, … Sub_r
- Find the feature subset Sub_{final} with the highest weights
- Backward feature selection with features in Sub_{final}
  - Classify the test samples with selected features

Identification of feature set
- Find minimum feature set Set_{min} and maximum feature set Set_{max} which give the best accuracy
- Set the features in Set_{min} as representative features
- Repeat until $Set_{diff} = Set_{max} - Set_{min} = \{f_{d1}, f_{d2}, \ldots, f_{dn}\} = []$
  - Compute the correlation coefficient between f_{di} and features contained in Set_{min}
  - Assign f_{di} to one of the feature in Set_{min},which has the most high correlation coefficient value, as a family member
- Analysis of measurements

Figure 5.1. A feature selection method.

in the discovery process such as the Jaccard distance. In this paper, we use the following distance measure [85].

$$D(J(i), K(j)) = 1 - \frac{|T(J(i)) \cap T(K(j))|}{|T(J(i)) \cup T(K(j))|}, \tag{5.4}$$

where $\mathcal{T} = \{t_1, t_2, ..., t_k\}$ is the transaction set, and $I(t_i) \subseteq I$ is the set of items in transaction t_i. For any itemset P, $T(P) = \{t \in \mathcal{T} | P \subseteq I(t)\}$ is the corresponding set of transactions. A distance measure D is a function mapping to a value in [0,1], where 0 means two patterns are completely relevant and 1 means two patterns are totally independent.

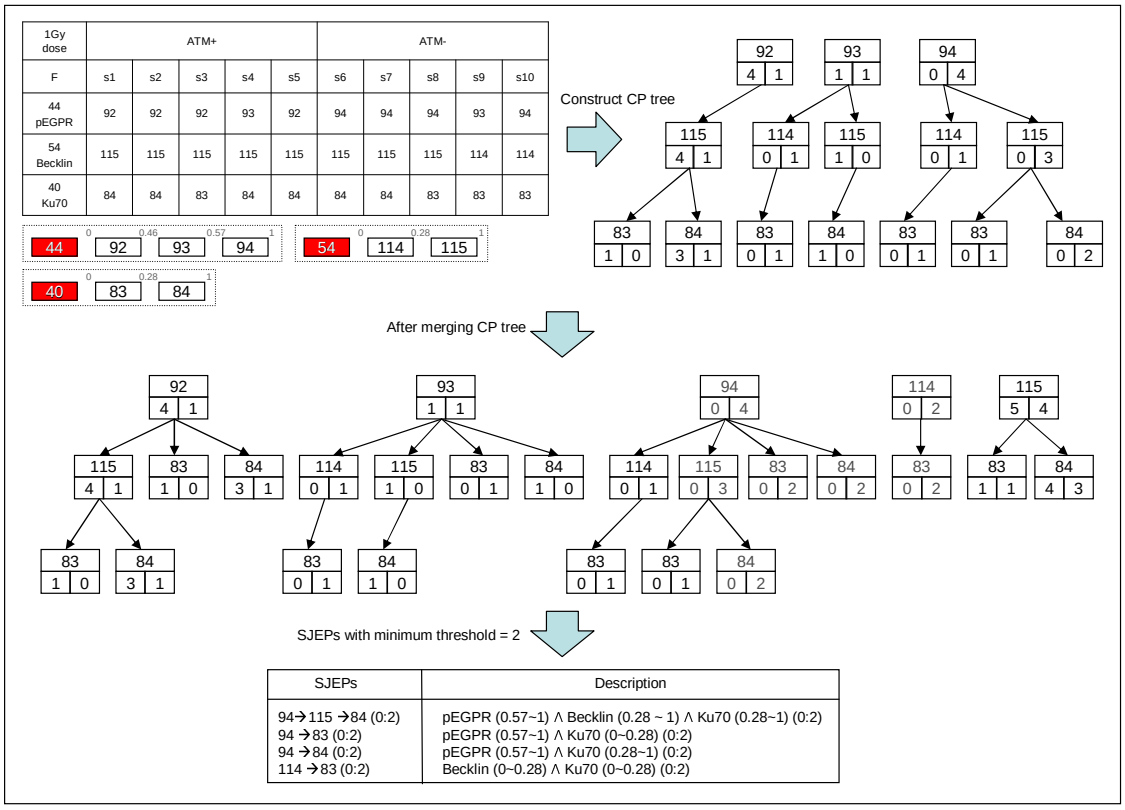


Figure 5.2. Finding SJEPs using the Contrast Pattern Tree (CP-tree).

Feature patterns (combination of features) identification techniques could be used to capture more underlying semantics than single feature. However, it is very hard to find meaningful patterns in large datasets like microarray data because of the huge search space. Furthermore, infrequent patterns are often irrelevant or do not improve the accuracy of the classification. To tackle these problems, we designed a Discriminative Network Pattern Identification System (DiNPIS). This framework contains three steps: feature selection, feature pattern identification, and feature pattern annotation.

5.2.2.1 A feature selection method

The responsive proteins under different IR doses and at different time points are selected by building a connection between pattern frequency (pattern support value) and discriminative measures. This method finds a feature subset for each feature which includes the minimum redundant features with strong relevance to the target class of the given feature based on a relative feature significance measure. With these feature subsets, we run the linear SVMs algorithm where two-thirds of the samples are utilized for training and the remaining one third for testing. Then, we compute the weight for certain feature f_k based on the idea proposed in [87]:

$$W_k = \begin{cases} \dfrac{|w_k|S(f_k)}{\sum_{j=1}^{d+1}|w_j|S(f_j)} \times \beta \times \delta, & \text{for } \gamma \leq \beta, \\ \left(1 - \dfrac{|w_k|S(f_k)}{\sum_{j=1}^{d+1}|w_j|S(f_j)}\right) \times (\gamma - \beta) \times \delta, & \text{for } \gamma > \beta, \end{cases} \tag{5.5}$$

where

$$\delta = \begin{cases} 1, & \text{for } \gamma \leq \beta \\ -1, & \text{for } \gamma > \beta \end{cases} \tag{5.6}$$

and β is the accuracy using testing samples, γ is a predefined threshold, and $|w_k|$ is the absolute SVM weight obtained using Eq 5.2. Each $|w_k|S(f_k)$ is normalized by dividing the summed $|w_k|S(f_k)$ value of all the features in the subset. $S(f_k)$ is the feature significance under Def 2. Different from the work in [87], in our approach, feature significance $S(f_k)$ is incorporated in the feature selection process, which reflects a feature's global discriminant power. All the proteins are ranked based on the normalized feature weights. A set of the top ranked features is selected based on the prediction accuracy for ATM+ and ATM- cells. Finally, the backward

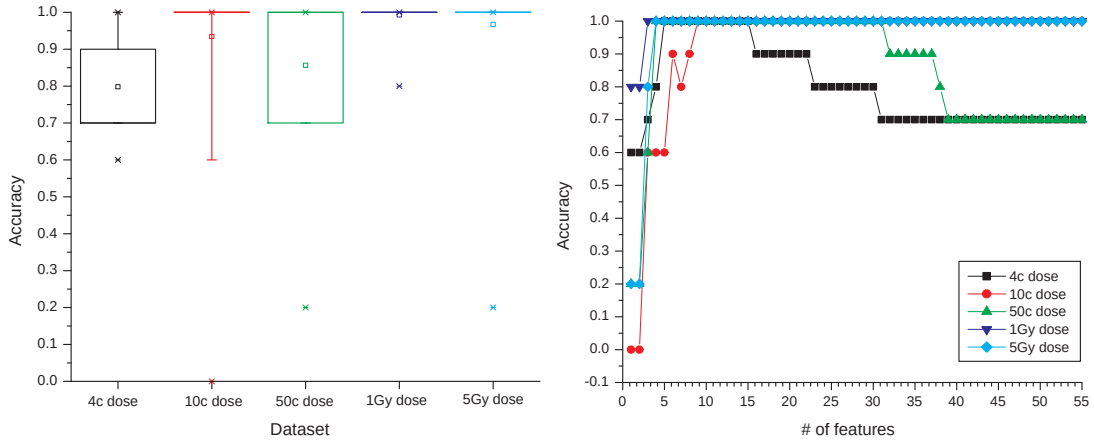


Figure 5.3. Performance of feature selection (a) Box plot of accuracy at different dose levels, (b) Accuracy of DiNPIS-FS (feature selection) when different top-ranked features are selected.

selection (elimination) is further applied to obtain a compact protein/kinase set that represents the most responsive probes.

In most computational biology applications such as diagnosis and biomarker identification, a minimum redundancy feature set that gives the best prediction accuracy is selected. However, the minimum feature set may not be reflect all the relevant proteins/kinases involved in the pathways. This can be a critical problem in our research for identifying the dynamic network responses induced by ionizing radiation at different dose levels. Thus we also consider the maximum feature set which gives the best results. The minimum feature set is used as representative features of the maximum feature set. Then each feature contained in the maximum feature set is assigned to one of these representatives as a family member using correlation coefficient. This algorithm is summarized in Fig. 5.1.

5.2.2.2 Feature Pattern Identification

Based on the most responsive proteins selected from the *feature selection* module and the calculation of relative feature (proteins/probes/antibodies) significance, we will be able to find the protein network patterns. To efficiently search all possible network patterns, we employed SJEPs mining algorithm using the contrast pattern tree (CP-tree) [88]. Jumping Emerging Patterns (JEPs) are those patterns whose supports (frequencies) increase abruptly from zero in one data set to nonzero in another data set [4]. It has the special advantages of modeling interactions among features and building powerful classifiers. In our research, a subset of JEPs, called Strong Jumping Emerging Patterns (SJEPs) is considered to remove potentially less useful JEPs while retaining those with high discriminating power.

A CP-tree is an ordered multi-path tree structure. Each node of the CP-tree shows a variable number of items (expression intervals of proteins). The expression levels at each node are ordered in terms of pattern significance. The branches of the tree reflect parent-child relationship. The cp-tree is constructed using three operations: *createTree*, *mergeTree*, and *mineTree*. In *createTree* operation, In *createTree* operation, the CP-tree is constructed by using the new ordering of each transaction based on the feature rank identified by Eq. (5), while the original work on CP-tree reorders transactions based on the feature support value [88]. The order of a CP-tree is very important to extract SJEPs. However, there are some critical issues when we use only the feature support value for reordering. First, there are many cases that the support values of features are equivalent. Second, feature support value itself is not enough to rank features. Therefore, reordering based on the feature weight has the strong advantage to efficiently extract SJEPs. In *mergeTree* operation, the CP-tree is extended to find the complete set of paths (SJEPs). Finally, in *mine-*

Tree operation, we start to search the CP-tree depth-first for SJEPs. Because every training instance is sorted by its rank when inserting into the CP-tree, items with high rank, which are more likely to appear in an SJEP, are closer to the root. Using the predefined order, we can produce the complete set of paths (item sets) systematically through depth-first searches of the CP-tree. After completing the search of the CP-tree, we select only minimal patterns by filtering out the patterns that are supersets of others. The remaining minimal ones are SJEPs since they satisfy the minimum support threshold. Fig. 5.2 shows an example of finding SJEPs using the CP-tree. In this figure, three selected probes are given: pEGPR, Belklin, and Ku70. As can be seen from the upper left table in Fig. 5.2, antibody pEGPR has three items numbered as 92, 93, and 94. Inside each node of the CP-tree, the top number indicates the item number, the lower left number shows the support value for ATM+, and the lower right number shows the support value for ATM- at the current tree level. The final selected protein motif patterns are listed as SJEPs at the bottom of the figure. As an example, one SJEP is composed of items $94 \rightarrow 115 \rightarrow 84$ with support value as 0 for ATM+ and 2 for ATM-.

5.2.2.3 Feature Pattern Annotation

The last step of the DiNPIS framework is to provide protein pattern annotation, which is important to assign a set of characteristics to feature patterns and thus to obtain relevant information for the interpretation of experimental results. Our goal is to generate annotations to provide information such as protein significance, relative protein significance, protein prediction ability (classification accuracy of different dosages), protein network motif significance, dependence relationship among proteins, and so on.

The feature pattern annotation is composed of four parts: experiment information, feature pattern analysis information, interaction diagrams of the representative features, and SJEP information. Experiment information includes experiment name, data description, summary of responsive feature selection result, summary of feature pattern identification, and descriptions of selected proteins. Feature pattern analysis information includes the graphical view of interactions among features for each stage of different time points, feature weights showing the importance of individual proteins, relative feature significance reflecting feature dependence, and radiation stage prediction information with selected features. Interaction diagram shows the relationships between the representative features generated by SJEPs. SJEP information shows all SJEPs (the protein network modules) identified from the CP-tree.

Table 5.1. Data description

Dataset	# of classes	# of samples	# of features	Description
Data1	2	10 (5/5)	55	4c dose, 5 time points
Data2	2	10 (5/5)	55	10c dose, 5 time points
Data3	2	10 (5/5)	55	50c dose, 5 time points
Data4	2	10 (5/5)	55	1Gy dose, 5 time points
Data5	2	10 (5/5)	55	5Gy dose, 5 time points
Data6	2	10 (5/5)	55	1hr, 5 doses
Data7	2	10 (5/5)	55	6hr, 5 doses
Data8	2	10 (5/5)	55	24hr, 5 doses
Data9	2	10 (5/5)	55	48hr, 5 doses
Data10	2	10 (5/5)	55	72hr, 5 doses
Data11	2	50 (25/25)	55	all times, all doses

5.3 Experiments

We applied quantum dot reverse-phase protein microarray [80] to profile the dynamic responses of several signaling pathways, including DNA damage, DNA repair, and cell cycle checkpoints, to low dose of Ionizing Radiation (IR) [81],[82].

Table 5.2. The number of minimum and maximum responsive protein sets under different doses and at different time points

Dataset	DFPIS-Feature selection					SVM-RFE				
	# of features		accuracy	sensitivity	specificity	# of features		accuracy	sensitivity	specificity
	min	max				min	max			
Data1	5	15	100	100	100	18	26	84	76	91
Data2	5	55	100	100	100	5	55	100	100	100
Data3	4	31	100	100	100	7	18	80	80	80
Data4	3	55	100	100	100	15	55	90	80	100
Data5	4	55	100	100	100	7	55	100	100	100
Data6	5	55	100	100	100	8	55	100	100	100
Data7	6	55	100	100	100	14	55	100	100	100
Data8	3	55	100	100	100	12	55	80	80	80
Data9	7	21	100	100	100	39	55	90	80	100
Data10	3	55	100	100	100	7	55	100	100	100
Data11	7	55	100	100	100	10	55	100	100	100

Table 5.3. Comparison of interactions for 4 cGy and 5 Gy dose

*rep : representative

No	4c dose	Mapping to 5 Gy dose network	5Gy dose	No	5Gy dose	Mapping to 4c dose network	4c dose
1	33 → 0	44 (33's rep)→44 (0's rep)	44 in rep set	1	46→54	25 (46's rep)→25 (54's rep)	25 in rep set
2	0→20	44 (0's rep)→46 (20's rep)	46→44	2	54→44	25 (54's rep)→0 (44's rep)	0→25
3	20→25	46 (20's rep)→54(25's rep)	46→54	3	44→1	0 (44's rep)→5 (1's rep)	0→20→25→5, 0→20→5, 0→25→5
4	25→5	54 (25's rep)→54(5's rep)	54 in rep set				
5	0→25	44 (0's rep)→54(25's rep)	55→44				
6	33→20	44 (33's rep)→46(20's rep)	46→44	4	46→44	25 (46's rep)→0 (44's rep)	0→25
7	20→5	46 (20's rep)→54(5's rep)	46→54	5	54→1	25 (54's rep)→5 (1's rep)	25→5

ATM-deficient (ATM-) and -proficient (transfected with full length ATM construct, ATM+) cells were treated with different doses of IR and cell lysates were collected at different time-points, serial diluted and spotted on an array in triplicate. The intensities of all antibodies were normalized relative to those of control and then were normalized to have from zero to one. The arrays were then probed with specific antibodies. 55 antibodies have been evaluated for the dynamic change of the network (see the lower part of Fig. 5.5. The five applied doses are 4 cGy, 10 cGy, 50 cGy, 1 Gy, and 5 Gy. Both types of cells for each dosage were observed at 1 h (hour), 6 h, 24 h, 48 h, and 72 h.

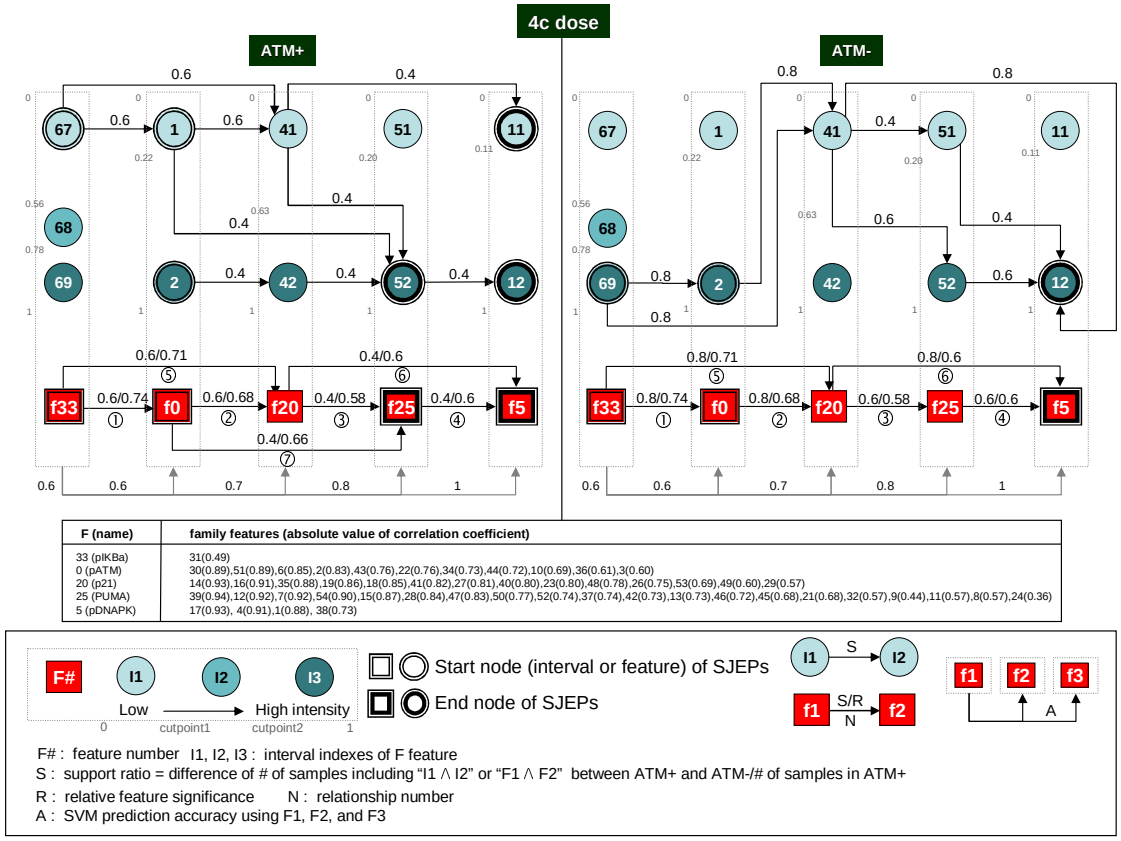
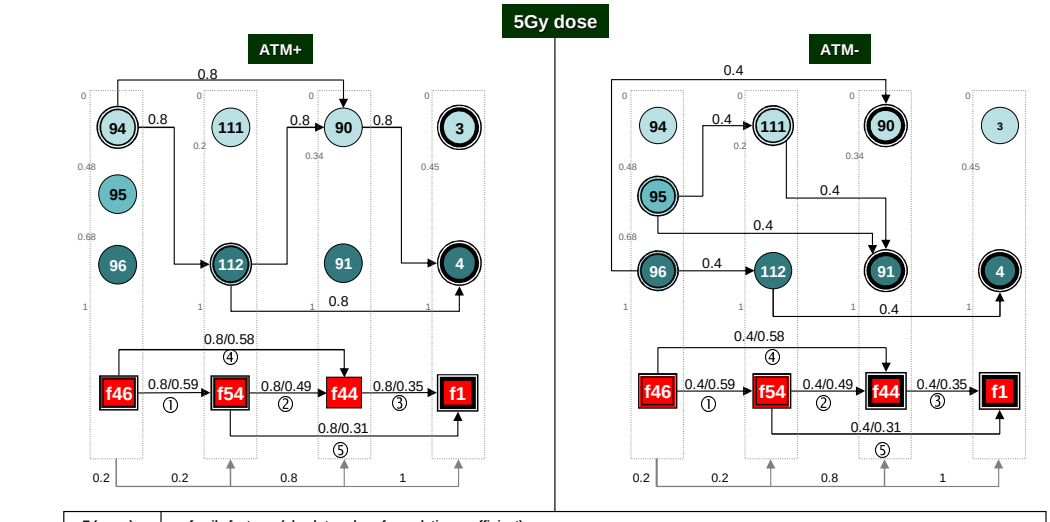


Figure 5.4. Interaction diagram of five representative probes on Data1 using 4c dose.

To test the performance of the proposed DiNPIS algorithm, classification was carried out by the linear SVM (soft margin C=1) and LOOCV (leave-one-out cross validation) evaluation was employed because of small number of samples.

Table 5.1 shows the data sets used in this experiment. These data sets treat intensities of certain dose at five different time points, intensities of all different dose level at certain time, and intensities of all different dose level at all time points as samples and have 55 antibodies as features. The classes of these datasets are labeled as either ATM-proficient (ATM+) or ATM-deficient (ATM-).



F (name)	family features (absolute value of correlation coefficient)
46 (pSmad3)	36(0.86),51(0.75),17(0.68),15(0.71),20(0.62),43(0.60),14(0.60)
54 (Becklin)	5(0.89),18(0.83),25(0.83),41(0.83),52(0.76),13(0.76),24(0.69),8(0.65),29(0.64),7(0.60),34(0.32),9(0.23)
44 (pEGFR)	0(0.80),31(0.80),33(0.79),3(0.77),30(0.76),12(0.73),6(0.72),28(0.71),38(0.70),37(0.68),42(0.66),11(0.65),27(0.62),45(0.62),19(0.58),47(0.44),21(0.43),22(0.40)
1 (pBRCA1)	23(0.86),50(0.77),35(0.76),16(0.76),39(0.75),26(0.75),10(0.75),4(0.73),32(0.71),53(0.71),48(0.71),40(0.70),49(0.64),2(0.56)

[Feature Information]

No	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
Name	pATM	pBRCA1	Chk1	Chk2	pChk2	pDNAPK	Src	pSrc	Erk	pERK	Akt	pAKT	PTEN	mTOR	p-mTOR	CynB1	CynD1	CynD3	cdk4	p16
No	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39
Name	p21	p53	p-p53	MDM2	MDMX	PUMA	pRaf	pJNK	GSK3b	pGSK3b	NFkBp65	pNFkBp65	IKBa	pIKBa	pBcl2	Cleaved_Casp3	Cleaved_Casp9	sClusterin	Klotho	gH2AX
No	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54					
Name	Ku70	IGFR1	pIGFR1	EGFR	pEGFR	Smad3	pSmad3	pDAB2IP	IRS1	pIRS1	cAbl	pAbl	Stat3	pStat3	Becklin					

Figure 5.5. Interaction diagram of four representative probes on Data5 using 5Gy dose.

5.3.1 Computational analysis: feature selection

The discovery of different responsive probe sets for different dosages and at different time points are given in Tab. 5.2. In this table, the minimum feature set indicates the list of selected features by DiNPIS feature selection, and the maximum feature set indicates all the other relevant probes with respect to each selected probe in the minimum set. This table shows that A-T cells had been significantly effected by low dose IR as well as high dose IR. However, we note that only $5 \sim 15$ features were selected in Data1 under 4c dose. It shows that many of features significantly effected by high dose IR have been functioned by row dose IR not as much as by high dose IR. We also could observe different effects on low dose IR and high dose IR in Fig. 5.3. To evaluate the performance of our algorithm, we carried out comparison

experiments with SVM-RFE feature selection. As seen from Tab. 5.2, the accuracy rates using DiNPIS-FS generally outperform the SVM-RFE.

5.3.2 Computational analysis: feature pattern identification

To analyze the dynamic network responses induced by different IR levels, we give examples on two feature interaction diagrams on Data1 using 4c dose and Data5 using 5Gy dose in feature pattern annotation.

We found six SJEPs for both ATM+ and ATM- on Data1. From these patterns, seven relationships between five representative features were found. As shown in Fig. 5.4, the first and sixth feature relationships were found in both classes. However, note that fluorescence intensities of features are expressed differently during these interactions. For instance, the dependency of feature f0 (pATM) causes the intensity of f20 (p21) go up in ATM+ class but leads it down in ATM- class. The seventh relationship disappeared in ATM-. According to the support ratio and the relative feature significance assigned to each relationship, the first, second, fifth, and the sixth relationships are sightly stronger than the third, fourth, and seventh relationships.

We found three SJEPs for ATM+ and five SJEPs for ATM- on Data5. From these patterns, five relationships between four representative features were founded. As shown in Fig. 5.5, all of the five feature relationships were founded in both classes. However, note that fluorescence intensities of features are expressed differently during these interactions except for the fifth relationship. According to the support ratio assigned to each relationship, the strength of all relationships in ATM+ is slightly reduced in ATM-.

5.3.3 Biological observations

As shown in Tab. 5.3, we investigated whether interactions of selected features at different dose IR levels are related to each other.

First, all of the four representative probes including pSmad3, Becklin, pEGFR, and pBRCA1 on Data5 (5 Gy dose) were found in the maximum feature set on Data1 (4 cGy dose). It shows that these antibodies still play an important role under low dose IR level. Second, all of the relationships in Data5 are related to those of Data1. In DiNPIS-feature pattern identification, we assume that a family member has the same or similar relationships as the ones of its representative features. Thus five relationships on Data5 were matched with similar five relationships on Data1 in Tab. 5.3. For instance, the fifth relationship in Data5 was assigned to the fourth relationship in Data1 since feature f25 (PUMA) was a representative of f54 (Becklin) that has a 0.90 correlation coefficient with f25, f5 (pDNAPK) was a representative of f1 (pBRCA1) holding a 0.88 correlation coefficient with f5, and there exists the fourth relationship between f25 and f5 in Data1.

Finally, we observe some reverse relationships. As an example, the second relationship in Data5 corresponds to the reverse of the seventh relationship in Data1. In our research, teh direction of dependence was determined by a new feature rank identified in DiNPIS-feature selection. Thus this reverse relationship can be identified if major features are changed as dose IR levels are changed. However, to provide more information about directions of relationships, we need further study by considering all possible directions of relationships.

5.4 Conclusion

This paper presented exploratory work on identifying signaling molecules under low dose ionizing radiation by using reverse phase protein array (RPPM) in conjunction with quantum dot. A computational framework, Discriminative Network Pattern Identification System (DiNPIS), is developed to recognize the contributing network motifs in different pathways and to take into the consideration of protein dependence. For feature selection, the most responsive proteins at different time points are identified. The interaction patterns of those selected probes are discovered by employing SJEP pattern mining based on a contrast pattern-tree. The last step of feature pattern annotation provides a complete pattern characterization such as single probe significance, relative pair-wise probe dependence, and pattern significance. The pilot study does reveal the quantitative change of different protein/kinase expression levels in different patterns. For future work, we plan to increase the sample size and the number of probes. In addition, we will investigate and biologically validate the individual signaling pathways affected under different dose and in time series.

REFERENCES

[1] X. Li, S. Rao, Y. Wang, and B. Gong, "Gene mining: a novel and powerful ensemble decision approach to hunting for disease genes using microarray expression profiling," *Nucleic Acids Research*, vol. 32, no. 9, pp. 2685–2694, 2004.

[2] L. Parsons, E. Haque, and H. Liu, "Subspace clustering for high dimensional data: A review," in *ACM SIGKDD Explorations Newsletter*, vol. 6, 2004, pp. 90–115.

[3] A. Boulesteix, G. Tutz, and K. Strimmer, "A cart-based approach to discover emerging patterns in microarray data," *Bioinformatics*, vol. 19, no. 18, pp. 2465–2472, December 2003.

[4] J. Li, G. Dong, and K. Ramamohanarao, "Making use of the most expressive jumping emerging patterns for classfication," *Knowledge and Information Systems*, vol. 3, no. 2, pp. 131–145, 2001.

[5] J. Li and L. Wong, "Identifying good diagnostic genes or genes groups from gene expression data by using the concept of emerging patterns," *Bioinformatics*, vol. 18, no. 5, pp. 725–734, 2002.

[6] L. Yu, F. Chung, S. Chan, and S. Yuen, "Using emerging pattern based projected clustering and gene expression data for cancer detection," in *Proceedings of the second conference on Asia-Pacific bioinformatics*, 2004, pp. 75–84.

[7] M. Law, M. Figueiredo, and A. Jain, "Simultaneous feature selection and clustering using a mixture model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1154–1166, 2004.

[8] G. Dong and J. Li, "Efficient mining of emerging patterns: Discovering trends and differences," in *Proceedings of the Fifth International Conference on Knowledge Discovery and Data Mining*, 1999, pp. 43–52.

[9] A. Soulet, B. Cremilleux, and F. Rioult, "Condensed representation of emerging patterns," in *Proceedings of the 8th conference on Asia-Pacific Knowledge Discovery and Data Mining*, 2004, pp. 127–132.

[10] U. Fayyad and K. Irani, "Multi-interval discretization of continuous-valued attributes for classification learning," in *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, 1993, pp. 1022–1029.

[11] P.Pudil, J. Novovicova, and J. Kittler, "Feature selection based on the approximation of class densities by finite mixtures of the special type," *Pattern Recognition*, vol. 28, no. 9, pp. 1389–1398, 1995.

[12] S. Vaithyanathan and B. Dom, "Generalized model selection for unsupervised learning in high dimensions," in *The Neural Information Processing Systems*, 1999, pp. 970–976.

[13] M. Figueiredo and A. Jain, "Unsupervised learning of finite mixture models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 3, pp. 381–396, March 2002.

[14] G. Celeux, S. Chretien, F. Forbes, and A. Mkhadri, "A component wise am algorithm for mixtures," INRIA, Tech. Rep. RR-3846, August 1999.

[15] U. Alon, N. Barkai, D. Notterman, K. Gish, S. Ybarra, D. Mack, and A. Levine, "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," in *Proceedings of the National Academy of Sciences*, 1999, pp. 6745–6750.

[16] D. Singh, P. Febbo, K. Ross, D. Jackson, J. Manola, C. Ladd, P. Tamayo, A. Renshaw, A. DAmico, J. Richie, E. Lander, M. Loda, P. Kantoff, T. Golub,

and W. Sellers, "Gene expression correlates of clinical prostate cancer behavior," *Cancer cell*, vol. 1, no. 2, pp. 203–209, March 2002.

[17] G. McLachlan, R. Bean, and D. Peel, "A mixture model-based approach to the clustering of microarray expression data," *Bioinformatics*, vol. 18, no. 3, pp. 413–422, 2002.

[18] C. Aggarwal and P. Yu, "Finding generalized projected clusters in high dimensional spaces," in *Proceedings of ACM SIGMOD conference*, 2000, pp. 70–81.

[19] A. Ben-Dor, R. Shamir, and Z. Yakhini, "Clustering gene expression patterns," *Journal of Computational Biology*, vol. 6, no. 3-4, pp. 281–297, October 1999.

[20] A. Ben-Dor, L. Bruhn, N. Friedman, I. Nachman, M. Schummer, and Z. Yakhini, "Tissue classification with gene expression profiles," *Journal of Computational Biology*, vol. 7, no. 3-4, pp. 559–583, August 2000.

[21] S. Cal, V. Quesada, M. Llamazares, A. Diaz-Perales, C. Garabaya, and C. Lopez-Otin, "Human polyserase-2, a novel enzyme with three tandem serine protease domains in a single polypeptide chain," *Journal of Biological Chemistry*, vol. 280, no. 3, pp. 1953–1961, 2005.

[22] N. Durany, J. Joseph, E. Campo, R. Molina, and J. Carreras, "Phosphoglycerate mutase, 2,3-bisphosphoglycerate phosphatase and enolase activity and isoenzymes in lung, colon and liver carcinomas," *Journal of Cancer*, vol. 75, no. 7, pp. 969–977, 1997.

[23] Y. Yap, X. Zhang, M. Ling, X. Wang, Y. Wong, and A. Danchin, "Classification between normal and tumor tissues based on the pair-wise gene expression ratio," *BMC Cancer*, vol. 4, no. 1, p. 72, 2004.

[24] Y. Tokuda, Y. Satoh, C. Fujiyama, H. S. S. Toda, and Z. Masaki, "Prostate cancer cell growth is modulated by adipocyte-cancer cell interaction," *British Journal of Urology International*, vol. 21, no. 7, p. 716, 2003.

[25] J. Luo, Y. Yu, K. Cieply, F. Lin, P. Deflavia, R. Dhir, S. Finkelstein, G. Michalopoulos, and M. Becich, "Gene expression analysis of prostate cancers," *Molecular Carcinogenesis*, vol. 33, no. 1, pp. 25–35, 2002.

[26] H. Akaike, "A new look at the statistical model identification," *IEEE Transactions on Automatic Control*, vol. Ac(19), pp. 716–723, 1974.

[27] G. Schwarz, "Estimating the dimension of a model," *Annals of Statistics*, vol. 6, pp. 461–465, 1978.

[28] M. Browne, "Cross-validation methods," *Journal of Mathematical Psychology*, vol. 44, pp. 108–132, 2000.

[29] P. Grunwald, "Model selection based on minimum description length," *Journal of Mathematical Psychology*, vol. 44, pp. 133–152, 2000.

[30] J. Oliver, R. Baxter, and C. Wallace, "Unsupervised learning using mml," in *Proceedings of the 13th International Conference Machine Learning*, 1996, pp. 364–372.

[31] H. Bozdogan, "Akaikes information criterion and recent developments in information complexity," *Journal of Mathematical Psychology*, vol. 44, pp. 62–91, 2000.

[32] C. Biernacki, G. Celeux, and G. Govaert, "Assessing a mixture model for clustering with the integrated classification likelihood," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 7, pp. 719–725, 2000.

[33] G. McLachlan and D. Peel, *Finite Mixture Models*. Wiley-Interscience, 2000.

[34] H. Wang, B. Lue, Q. Zhang, and S. Wei, "Estimation for the number of components in a mixture model using stepwise split-and-merge em algorithm," *Pattern Recognition Letter*, vol. 25, pp. 1799–1809, 2004.

[35] N. Ueda, R. Nakano, Z. Ghahramani, and G. Hinton, "Split and merge em algorithm for improving gaussian mixture density estimates," in *Proceedings of IEEE Signal Processing Society Workshop*, 1998, pp. 274–282.

[36] N. Ueda and R. Nakano, "Em algorithm with split and merge operations for mixture models," *Systems and Computers in Japan*, vol. 31, no. 5, pp. 1–11, 2000.

[37] X. D. Luna and K. Skouras, "Choosing a model selection strategy," *Scandinavian Journal of Statistics*, vol. 30, pp. 113–128, 2003.

[38] R. Pintelon and J. Schoukens, *System Identification. A frequency domain approach.* Wiley-IEEE Press, 2001.

[39] P. Grunwald, *Advances in Minimum Description Length: Theory and Applications.* MIT Press, 2005, ch. Chapters 1 and 2 : A Tutorial Introduction to the Minimum Description Principle, pp. 1–80.

[40] M. Figueiredo, J. M. N. Leitao, and A. K. Jain, "On fitting mixture models," in *Energy Minimization Methods in Computer Vision and Pattern Recognition*, 1999, pp. 54–69.

[41] K. Fukunaga, *Introduction to Statistical Pattern Recognition, Second Edition.* Academic Press, 1990.

[42] P.Pudil, J. Novovicova, and J. Kittler, "Floating search methods in feature selection," *Pattern Recognition Letter*, vol. 15, no. 11, pp. 1119–1125, 1994.

[43] K. Mao, "Orthogonal forward selection and backward elimination algorithms for feature subset selection," *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics*, vol. 34, no. 1, pp. 629–634, 2004.

[44] L. Talavera, "Dependency-based feature selection for clustering symbolic data," *Intelligent Data Analysis*, vol. 4, pp. 19–28, 2000.

[45] J. Pena, J. Lazano, P. Larranaga, and I. Inza, "Dimensionality reduction in unsupervised learning of conditional gaussian networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, pp. 590–603, 2001.

[46] J. Dy and C. Brodley, "Feature subset selection and order identification for unsupervised learning," in *Proceedings of the 17th International Conference on Machine Learning*, 2000, pp. 247–254.

[47] S. Vaithyanathan and B. Dom, "Hierarchical unsupervised learning," in *Proceedings of the 17th International Conference on Machine Learning*, 2000, pp. 1039–1046.

[48] N. Sondberg-Madsen, C. Thomsen, and J. Pena, "Unsupervised feature subset selection," in *Proceedings of the Workshop on Probabilistic Graphical Models for Classification (within ECML/PKDD)*, 2003, pp. 71–82.

[49] Y. Kim, J. Oh, and J. Gao, "Emerging pattern based subspace clustering of microarray gene expression data using mixture models," in *Proceedings of the International Conference Advances In Bioinformatics And Its Applications*, vol. 8, 2004, pp. 13–24.

[50] T. Hastie, R. Tibshirani, M. Eisen, A. Alizadeh, R. Levy, L.Staudt, W. Chan, D. Botstein, and P. Brown, "'gene shaving' as a method for identifying distinct sets of genes with similar expression patterns," *Genome Biology*, vol. 1, no. 2, pp. 0003.1–0003.21, 2000.

[51] C. Ding, "Unsupervised feature selection via two-way ordering in gene expression analysis," *Bioinformatics*, vol. 19, no. 10, pp. 1259–1266, 2003.

[52] G. Dunteman, *Principal Components Analysis (Quantitative Applications in the Social Sciences)*. Sage Publications, 1989.

[53] G. McCabe, "Principal variables," *Technometrics*, vol. 26, pp. 127–134, 1984.

[54] I. Jolliffe, *Principal Component Analysis*. Springer, 1986.

[55] W. Krzanowski, "Selection of variables to preserve multivariate data structure, using principal component analysis," *Applied Statistics-Journal of the Royal Statistical Society Series C*, vol. 36, pp. 22–33, 1987.

[56] ——, "A stopping rule for structure- preserving variable selection," *Statistics and Computing*, vol. 6, pp. 51–56, 1996.

[57] K. Mao, "Identifying critical variables of principal components for unsupervised feature selection," *IEEE Transactions on Systems Man and Cybernetics*, vol. 35, no. 2, pp. 339–344, 2005.

[58] D. Frossyniotis, A. Likas, and A. Stafylopatis, "A clustering method based on boosting," *Pattern Recognition Letters*, vol. 25, pp. 641–654, 2004.

[59] U. Maulik and S. Bandyopadhyay, "Performance evaluation of some clustering algorithms and validity indices," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 12, pp. 1650–1654, 2002.

[60] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the em algorithm (with discussion," *Journal of the Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.

[61] G. McLachlan and T. Krishnan, *The EM Algorithm and Extensions*. Wiley-Interscience, 1997.

[62] D. Frossyniotis, M. Pertselakis, and A. Stafylopatis, "A multi-clustering fusion algorithm," in *Proceedings of the 2nd Hellenic Conference on AI*, 2002, pp. 225–236.

[63] Y. Freund and R. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of Computer and System Sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[64] R. Schapire and Y. Singer, "Improved boosting algorithms using confidence-rated predictions," *Machine Learning*, vol. 37, no. 3, pp. 297–336, 1999.

[65] Y. Kim and J. Gao, "A new semi-supervised subspace clustering algorithm on fitting mixture models," in *Proceedings of the IEEE Symposium on Computational Intelligence In Bioinformatics and Computational Biology*, 2005, pp. 1–8.

[66] K. Yeung and W. Ruzzo, "Principal component analysis for clustering gene expression data," *Bioinformatics*, vol. 17, no. 9, pp. 763–774, 2001.

[67] P. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 1987.

[68] A. Bensaid, L. Hall, J. Bezdek, L. Clarke, M. Silbiger, J. Arrington, and R. Murtagh, "Validity-guided (re)clustering with applications to image segmentation," *IEEE Transactions on Fuzzy Systems*, vol. 4, pp. 112–123, 1996.

[69] J. Dy and C. Brodley, "Feature selection for unsupervised learning," *Journal of Machine Learning Research*, vol. 5, pp. 845–889, 2004.

[70] T. Golub, D. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. Mesirov, H. Coller, M. Loh, J. Downing, M. Caligiuri, C. Bloomfield, and E. Lander, "Molecular classification of cancer: class discovery and class prediction by gene expression monitoring," *Science*, vol. 286, pp. 531–537, 1999.

[71] R. Kohavi and G. John, "Wrapper for feature subset selection," *Artificial Intelligence*, vol. 97, no. 1-2, pp. 273–324, 1997.

[72] S. Das, "Filters: Wrappers and a boosting-based hybrid for feature selection," in *Proceedings of the 18th International Conference on Machine Learning*, 2001, pp. 74–81.

[73] H. Liu and L. Yu, "Toward integrating feature selection algorithms for classification and clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 4, pp. 491–502, 2005.

[74] H. Chung, H. Liu, S. Brown, C. McMunn-Coffran, C. Kao, and D. frank Hsu, "Identifying significant genes from microarray data," in *Proceedings of the 4th IEEE Symposium on Bioinformatics and Bioengineering*, 2004, pp. 358– 365.

[75] C. Ding and H. Peng, "Minimum redundancy feature selection from microarray gene expression data," *Journal of Bioinformatics and Computational Biology*, vol. 3, no. 2, pp. 185–205, 2005.

[76] L. Yu and H. Liu, "Efficient feature selection via analysis of relevance and redundancy," *Journal of Machine Learning Research*, vol. 5, pp. 1205–1224, 2004.

[77] F. L. H. Peng and C. Ding, "Feature selection based on mutual information:criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[78] P. Mitra, Murthy, and S. Pal, "Unsupervised feature selection using feature similarity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 301–312, 2002.

[79] L. Yu and H. Liu, "Redundancy based feature selection for microarray data," in *Proceedings of the 10th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2004, pp. 22–25.

[80] D. Geho, N. Lahar, P. Gurnani, M. Huebschman, P. Herrmann, V. Espina, A. Shi, J. Wulfkuhle, H. Garner, E. Petricoin, L. Liotta, and K. Rosenblatt, "Pegylated, steptavidin-conjugated quantum dots are effective detection elements for reverse-phase protein microarrays," *Bioconjug. Chem.*, vol. 16, no. 3, pp. 559–566, 2005.

[81] F. Marchetti, M. Coleman, I. Jones, and A. Wyrobek, "Candidate protein biodosimeters of human exposure to ionizing radiation," *International Journal of Radiation Biology*, vol. 82, no. 9, pp. 605–639, 2006.

[82] Y. Ziv, A. Bar-Shira, I. Pecker, P. Russell, T. Jorgensen, I. Tsarfati, and Y. Shiloh, "Recombinant atm protein complements the cellular a-t phenotype," *Oncogene*, vol. 15, pp. 159–167, 1997.

[83] H. Cheng, X. Yan, J. Han, and C. Hsu, "Discriminative frequent pattern analysis for effective classification," in *Proceedings of the 2007 IEEE International Conference on Data Engineering (ICDE 07)*, Istanbul, Turkey, April 2007.

[84] D. Xin, H. Cheng, X. Yan, and J. Han, "Extracting redundancy-aware top-k patterns," in *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, August 2006, pp. 20–23.

[85] D. Xin, J. Han, X. Yan, and H. Cheng, "On compressing frequent patterns," *Data and Knowledge Engineering*, vol. 60, pp. 5–29, 2007.

[86] C. Burges, "A tutorial on support vector machines for pattern recognition," *Data Mining and Knowledge Discovery*, vol. 2, pp. 121–167, 1998.

[87] J. Oh, A. Nandi, P. Gurnani, P. Bryant-Greenwood, K. P. Rosenblatt, and J. Gao, "Prediction of labor for pregnant women using high-resolution mass spectrometry data," in *Proceedings of IEEE Symposium on Bioinformatics and Bioengineering (IEEE BIBE)*, 2006, pp. 332–339.

[88] H. Fan and K. Ramamohanarao, "Fast discovery and the generalization of strong jumping emerging patterns for building compact and accurate classifiers," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 6, pp. 721–737, 2006.

BIOGRAPHICAL STATEMENT

Young Bun Kim was born in Seoul, South Korea, in 1972. She received her B.S. and M.S. degrees from Soongsil University, South Korea, in 1995 and 1997 respectively, all in Computer Science, Ph.D. degree from The University of Texas at Arlington in 2008, in Computer Science and Engineering.

During her four-year employment at LG-EDS and at Motorola from 1996 to 2001, she carried out many projects based on Object Oriented Technology, and gained plenty of working experience as a software engineer. During two-year stint at Sun Microsystems from 2001 to 2002 as a senior consultant, she provided consulting services related to corporate education solutions.

Her primary research interest is to develop machine learning methods and their applications to bioinformatics and system biology problems. Her work covers the following areas: feature (biomarker) selection, class prediction and classification and feature patterns (gene/protein interactions) identification.