A Framework to Discover Emerging Patterns for Application in Microarray Data

Anne-Laure Boulesteix Gerhard Tutz

11th March 2003

Ludwig-Maximilian-Universität Munich

Institute for Statistics, Ludwigstr.33

D–80539 Munich, Germany

email:socher@stat.uni-muenchen.de

Key Words: Emerging patterns, interactions between variables, classification trees, supervised learning

Abstract

Various supervised learning and gene selection methods have been used for cancer diagnosis. Most of these methods do not consider interactions between genes, although this might be interesting biologically and improve classification accuracy. Here we introduce a new CART-based method to discover emerging patterns. Emerging patterns are structures of the form $(X_1 > a_1) \cap (X_2 < a_2)$ that have differing frequencies in the considered classes. Interaction structures of this kind are of great interest in cancer research. Moreover, they can be used to define new variables for classification. Using simulated data sets, we show that our method allows the identification of emerging patterns with high efficiency. We also perform classification using two publicly available data sets (leukemia and colon cancer). For each data set, the method allows efficient classification as well as the identification of interesting patterns.

1 Introduction

In recent years, microarrays have become a very popular technique to measure the expression levels of thousands of genes simultaneously. In cancer medicine, microarrays are a potentially efficient tool to perform reliable diagnosis, since many genes are differentially expressed according to the tumor type. This involves sophisticated supervised learning methods in the 'small n large p' framework, where n is the number of patients (observations) and p the number of genes (variables). Microarrays are believed to allow finer and more reliable identification of tumor types than the usual clinical methods. This is of crucial importance, because a very fine diagnosis is necessary for the efficient treatment of patients. Thus, a large amount of literature deals with supervised learning methods with application to microarray gene expression data.

Emerging patterns (denoted as EPs in the whole paper) are particular interaction structures. They were first introduced in the computer science and data mining framework by (Dong and Li, 1999) and then applied to microarray data by (Li and Wong, 2001, 2002). Here, we are interested in the statistical background of emerging patterns and redefine emerging patterns for this purpose. Subsequently, we propose a new simple CART-based method to discover relevant emerging patterns as well as a classification scheme to use these emerging patterns for supervised learning. In the following, we will consider only data sets with two classes denoted as class 1 and class 2.

For the illustration of the concept of EPs, let us imagine that most patients with low (or high) expression levels of gene A and gene B belong to class k ($k = 1, 2$) and that this is the case for most patients from class k, as is depicted in Figure 1. This would be a valuable piece of information, both for statisticians who want to classify the samples and for biologists who try to determine the function of genes and the mechanisms of cancer. But this kind of pattern is quite difficult to detect. In particular, standard methods for gene selection (such as t-test) and classification miss genes involved in such interactions. Emerging patterns are patterns of this type.

EPs were introduced by (Dong and Li, 1999) who define them as 'itemsets whose supports increase signinficantly from one data set D_1 to another, D_2. Examples of emerging patterns are

$$(gene_A \geq 1.023) \cap (gene_B \geq 0.789),$$

$$(gene_C \geq 1.156) \cap (gene_D \geq 0.913).$$

with the relative frequencies in D_1 and D_2

EPs	D_1	D_2
$(gene_A \geq 1.023) \cap (gene_B \geq 0.789)$	0%	100%
$(gene_C \geq 1.156) \cap (gene_D \geq 0.913)$	85.7%	6.25%

EPs of the kind $(gene_C \geq 1.156) \cap (gene_D \geq 0.913)$, for which the frequency in class 1 is larger than in class 2, will be denoted as EPs of type 1. EPs of type $(gene_A \geq 1.023) \cap (gene_B \geq 0.789)$, for which the frequency is larger in class 2 than in class 1, will be denoted as EPs of type 2. In general let $n_1 = |D_1|$, $n_2 = |D_2|$ denote the sample sizes of data sets D_1 and D_2. For a specific pattern P the counts $n_{P,1}$ and $n_{P,2}$ within data sets D_1 and D_2 yield the simple structure

$$\begin{array}{c|cc|} & D_1 & D_2 \\ \cline{2-3} P & n_{P,1} & n_{P,2} \\ \cline{2-3} \end{array}$$

The growth rate from D_1 to D_2 is defined as

$$GR_{D_1 D_2}(P) = \frac{n_{P,2}/n_2}{n_{P,1}/n_1}.$$

by using the convention $0/0 = 1$ and $c/0 = \infty$ for $c \neq 0$. A large value of $GR_{D_1 D_2}(P)$ indicates an EP of type 2. The growth rate from D_2 to D_1

$$GR_{D_2 D_1}(P) = \frac{n_{P,1}/n_1}{n_{P,2}/n_2}$$

is simply the inverse of $GR_{D_1 D_2}$. A large value of $GR_{D_2 D_1}(P)$ indicates an EP of type 1.

For $\rho > 1$, Dong and Li (1999) call P a ρ-emerging pattern from D_1 to D_2 if

$$GR_{D_1 D_2}(P) \geq \rho.$$

and a ρ-emerging pattern from D_2 to D_1 if

$$GR_{D_2 D_1}(P) \geq \rho$$

In the original paper (Dong and Li, 1999), EPs are defined as itemsets (containing possibly more than 2 items) with high growth rate from D_1 to D_2 or from D_2 and

D_1. In (Li and Wong, 2001, 2002), the same authors only look for EPs with infinite growthrate. Thus, an EP contains patients from only one of the two classes. Subsequently, the EPs are ordered according to the number of patients they contain. The discovering method of (Li and Wong, 2002) includes two steps: a pre-screening step leaving only 35 genes and a discovering step using a enumeration-based algorithm described in (Dong and Li, 1999).

The rest of the paper is organized as follows. In Section 2, we give our own definition of emerging patterns and present our discovering method as well as the associated classification method. In Section 3, we validate our method with simulated data sets and test it on two publicly available 'benchmark' microarray data sets. In Section 4, we compare our classification results to the results of other supervised learning methods and our emerging patterns to those of Li and Wong (2002).

2 Methods

2.1 An alternative definition of emerging patterns

In our view, the definition adopted by (Li and Wong, 2002) is inconvenient for two major reasons. First, it is very restrictive to require infinite growth rate, because microarray data are noisy. Second, it would be advantageous to replace the two-step evaluating method (considering successively the growth rate and the size of the pattern) by a single statistical criterion. We suggest an alternative definition of an EP as a pattern of the form

$$P = (gene_{i_1} \diamond a_1) \cap \ldots \cap (gene_{i_k} \diamond a_k)$$

(where $\diamond$ stands for $\leq$ or $>$), for which the hypothesis $p(P|D_1) = p(P|D_2)$ can be rejected to a certain confidence level. This definition is associated to a test statistic that has to be chosen. Note that for a pattern containing few samples, this hypothesis can not be rejected even if the pattern contains only patients of the same class. Thus, this criterion effectively replaces the two-steps evaluating method of (Li and Wong, 2002). The number of genes k is denoted as the order of the EP ($k \geq 1$).

2.2 Pre-Screening with empirical distribution function

Our method for discovering EPs with trees is computationally intensive if applied to all genes simultaneously. Thus, a pre-screening or gene selection step is necessary. As is seen from Figure 1, an emerging pattern involves genes which do not necessarily discriminate well when used alone. Thus, usual selection methods which score the genes separately may be too restrictive. These usual methods do not select all the interesting genes. On the other hand, in order to be part of an emerging pattern a variable must have some discriminatory power. Therefore, we propose a new univariate gene selection method which is as non-restrictive as possible and particularly well suited to the framework of emerging patterns. Our selection criterion for a gene is whether there exists a point where the empirical distribution function is less than α for one class and more than β for the other class or more than $1-\alpha$ for one class and less than $1-\beta$ for the other class, where α is a 'small parameter' (typically between 0 and 0.1) and β is a 'large parameter' (typically between 0.5 and 0.7). This gene selection procedure is very fast and selects most interesting genes, provided α is large enough and β is small enough. However, to be sure to get most interesting genes, one has to select uninteresting genes as well, which makes the discovering phase considerably slower. In order to reduce the number of parameters, we will always set α to 0.1.

Algorithm 1 ***: Pre-screening algorithm***

For each gene:

1. *Determine F_1 resp. F_2, the empirical distribution function of the observations from D_1 resp. D_2.*
2. *If either $\{x \in \mathbb{R} | F_i(x) < \alpha, F_{3-i}(x) > \beta\}$ or $\{x \in \mathbb{R} | F_i(x) > 1-\alpha, F_{3-i}(x) < 1-\beta\}$ is non-empty for $i = 1$ or $i = 2$, then select the gene.*

As an example, let us choose $\alpha = 0.1$ and $\beta = 0.7$. As is seen from Figure 4 the gene 456 from the leukemia data set would be selected because there exists an interval where the empirical distribution function for class 1 is smaller than α and the empirical distribution function for class 2 is larger than β. One of the points contained in this interval is marked in the panel.

2.3 Discovering emerging patterns with trees

2.3.1 Tree methodology

Classification trees are an efficient exploratory tool to detect structures in data (Breiman, 1984). These are based on recursive partitioning whereby the measurement space $\mathbb{R}^p$ is successively split into subsets. Let $x^T = (x_1, \dots, x_p) \in \mathbb{R}^p$ denote the gene expression levels of genes $1, \dots, p$. If A is a subset of $\mathbb{R}^p$ (corresponding to the partitioning of $\mathbb{R}^p$ into A and $\overline{A} = \mathbb{R}^P \setminus A$), the split of A based on the variable x_j divides A into

$$A_1(j, \mu) = \{x \in A | x_j \le \mu\},$$

$$A_2(j, \mu) = \{x \in A | x_j > \mu\}.$$

Thus the subset A is split by use of one variable, x_j, with the split simply depending on a threshold μ from the range of x_j. By starting with $A = \mathbb{R}^p$ and performing successive splittings one obtains a tree. The result of d splittings are subsets of $\mathbb{R}^p$ of the form

$$\{x | x_{i_1} \le \mu_1\} \cap \{x | x_{i_2} > \mu_2\} \cap \ldots \cap \{x | x_{i_d} \le \mu_d\}.$$

Thus a succession of splits yields a pattern. In general let a pattern P of order d be defined as a subset of $\mathbb{R}^p$ which is identified by the sequence $\{(j_s, \mathcal{I}_s), s = 1 \dots, d\}$ where $j_s \in \{1, \dots, p\}$ identifies the variable and $\mathcal{I}_s$ specifies the interval which in the simple case of binary splits has the form $\mathcal{I}_s = (-\infty, \mu_s]$ or $\mathcal{I}_s = (\mu_s, +\infty)$. The relationship between decision trees and EP is simple: a pattern is equivalent to a leaf.

For recursive partitioning, a splitting criterion has to be chosen. Possible criterions are for instance the deviance, the Gini-index, the entropy, etc (Tutz,2000). Since our goal is to find patterns for which the null-hypothesis $p(P|D_1) = p(P|D_2)$ can be rejected, we choose the deviance. The deviance, which is one of the most widely used splitting criterions for trees is very well suited because it corresponds to the fit of the model $p(P|\mathcal{D}_1) = p(P|\mathcal{D}_2)$. The deviance of a pattern P is defined as

$$D(P) = -2 \sum_{i=1}^{2} (n_{P,i} \log \frac{n_{P,i}}{n_P} + n_{\overline{P},i} \log \frac{n_{\overline{P},i}}{n_{\overline{P}}}).$$

Given a pattern P of order d, an additional split in variable j at μ yields a $(d+1)$-dimensional pattern. This new split is chosen to minimize the conditional deviance which is given by

$$D(P\cap\{x|x_j \leq \mu\}, P\cap\{x|x_j > \mu\} \;=\; D(P\cap\{x|x_j \leq \mu\}) + D(P\cap\{x|x_j > \mu\}) - D(P).$$

2.3.2 The discovering method

A tree is grown using a standard algorithm for classification trees (for instance the algorithm implemented in the package `tree` in R). If there are one or two 'good' leaf(ves) ('good' means that the within leaf deviance is low), the genes and the cut points defining this leaf are stored. Another tree is grown with all genes except the gene involved in the first splitting. An alternative is to eliminate all the genes involved in the definition of the leaf. The same is done for the second tree, and so on, until only one variable remains. Here, we limit the order of the patterns to 2, because we found out that almost all the patterns of greater order were not significant statistically due to the small number of observations. However, our method can easily be generalized for patterns of greater order, provided the number of observations is large enough to allow those patterns to be statistically significant. In this case, the gain of time produced by our method compared to the (computationally very intensive!) consideration of all possible patterns is even greater.

Algorithm 2 ***: Discovering algorithm***

1. *Initialize S as the set of all pre-screened genes.*

   While S is not empty:

2. *Grow a classification tree with the variables from S with maximal depth 2.*

3. *Select the biggest leaf with predicted class 1 (i.e. the leaf with the most observations in class 1) and the biggest leaf with predicted class 2 (i.e. the leaf with the most observations in class 2). These leaves are now called patterns.*

4. *For each selected pattern P of order 2:*

   (a) *Let P' denote the node which was splitted into P and $\overline{P}$. Thus we have $P = P' \cap P(j, \mathcal{I})$.*

(b) Test the two-sided null-hypothesis that the second splitting is unrelevant to the confidence level p_S by using Fisher's exact test.

(c) If the hypothesis can be rejected, then keep P in the set of the selected patterns. Else, replace P by the node P' in the set of the selected patterns.

5. *For each selected pattern, test from the global contingency table if it is significant, by using Fisher's exact test to the confidence level p_G.*

6. *Store the significant pattern(s) (and their dominant class).*

7. *Eliminate the gene involved in the first splitting of the tree from the set of variables S.*

8. *With the reduced S go to step 2.*

Alternatively, in step 7 one can eliminate all the genes involved in the discovered emerging patterns. It makes the algorithm faster, but interesting emerging patterns are potentially missed. The idea behind step 4 is that an EP of lower order is better than an EP of larger order for which consecutive splittings are unrelevant. For example, the pattern $P_{AB} = (gene_A > a) \cap (gene_B > b)$ depicted in Figure 3 has two relevant splittings, where as the splitting $(gene_C > c)$ in the pattern $P_{CD} = (gene_C > c) \cap (gene_D > d)$ is unrelevant.

Note that we use the deviance as splitting criterion for the tree algorithm and Fisher's exact test as evaluating criterion after the tree is grown. Although these two statistics correspond to the same null-hypothesis, they are different. We used the deviance criterion as splitting criterion when growing the tree, because this is the 'standard' criterion used by most tree users and it is known that it works well. However, it is better to evaluate the resulting pattern by carrying out Fisher's exact test, because the deviance test is an asymptotic test which might give biased results when applied to small leaves. In further work, one could try to implement a tree algorithm using Fisher's exact test as splitting criterion.

Finally, one obtains a list of emerging patterns of different types (1 and 2) and different orders. These can be used for classification, as described in the next paragraph.

2.4 Classification method

The idea is to define binary covariates based on EPs and apply a classical supervised learning method namely linear discriminant analysis on these new covariates. Suppose that we have a learning set $\mathcal{L}$ and a test set $\mathcal{T}$. Let n_L denote the number of observations in $\mathcal{L}$ and n_T denote the number of observations in $\mathcal{T}$.

Algorithm 3 ***: Classification algorithm***

1. *Apply the pre-screening algorithm to $\mathcal{L}$ with parameters α and β.*
2. *Detect the emerging patterns in $\mathcal{L}$ following the discovering algorithm. Let m denote the number of found emerging patterns.*
3. *Define new data matrices $\mathcal{L}'$ of dimensions $(n_L \times m)$ and $\mathcal{T}'$ of dimensions $(n_T \times m)$ as follows:*

$$\mathcal{L}'(i,j) = \begin{cases} 1\,, & \textit{if the } i\textit{-th training observation is in the } j\textit{-th EP} \\ 0\,, & \textit{otherwise} \end{cases}$$

$$\mathcal{T}'(i,j) = \begin{cases} 1\,, & \textit{if the } i\textit{-th test observation is in the } j\textit{-th EP} \\ 0\,, & \textit{otherwise} \end{cases}$$

4. *Perform linear discriminant analysis (details are given in the next paragraph) to predict the class of observations from $\mathcal{T}$ using the matrix $\mathcal{L}'$ as training data set and $\mathcal{T}'$ as test data set.*

For discriminant analysis, we make the following distributional assumptions for $\mathbf{X}^T = (X_1, \ldots, X_m)$, where the $X_j,\ j = 1, \ldots, m$ stand for the new variables:

$$\mathbf{X} \,|\, Y = 1 \sim \mathcal{N}_2(\mu_1, \lambda\mathbf{I}) \quad \text{where } \mu_1 \text{is the mean of } \mathbf{X} \text{ in class 1}$$
$$\mathbf{X} \,|\, Y = 2 \sim \mathcal{N}_2(\mu_2, \lambda\mathbf{I}) \quad \text{where } \mu_2 \text{is the mean of } \mathbf{X} \text{ in class 2}$$

where $\mathbf{I}$ is the identity matrix of dimension $(m \times m)$ and λ is a constant. It means that the variables are considered to be independent and to have the same variance λ. This simplified discriminant analysis method is also referred as nearest centroid. The assumptions are very strong, but the performance is better than by estimating many parameters on the basis of few observations.

3 Results

3.1 Simulated data

Generation of the data matrix

We generate a simulated data matrix containing n_{obs} observations and a total of n_g genes. Let $n_1 = n_2 = n_{obs}/2$ denote the numbers of observations in each class. Let n_{EP1} resp. n_{EP2} denote the numbers of EPs of type 1 resp. 2. Thus, $2n_{EP1}$ genes are involved in a EP of type 1, $2n_{EP2}$ are involved in a EP of type 2, and the remaining $n_g - 2 \cdot (n_{EP1} + n_{EP2})$ are not involved in any EP. To simplify, the expression levels are all in $[0, 1]$. Genes which are not involved in EPs are randomly generated according to the uniform distribution between 0 and 1. Only EPs of order 2 are generated.

To generate a pair of genes forming an EP of type 1, the following algorithm is run. EPs of type 2 are generated in the same way: one just has to exchange 1 and 2 in the algorithm.

Algorithm 4 ***: Algorithm for EP generation***

1. *The threshold values for both involved genes are randomly generated according to the uniform distribution between 0.25 and 0.75. The senses ('$>$' or '$<$') are also randomly chosen.*

2. *The two-dimensional expression level is generated according to the uniform distribution for each observation:*

   - *in the EP with probability q for observations from class 1 and $1 - q$ for observations from class 2.*
   - *outside the EP with probability $1 - q$ for observations from class 1 and q for observations from class 2.*

   where q is parameter reflecting the theoretical goodness of the EP.

The method for prescreening genes and discovering EPs was tested with several combinations of parameters (α,β,p_G,p_S). We decided to fix α to 0.1.

Results of the discovering method on simulated data

The total number of possible gene pairs is $p(p-1)/2$, where p is the number of genes. For each gene pair, let us define two binary variables: e, which equals 0 if the pair does not form an EP and 1 if the pair forms an EP, and d, which equals 0 if the pair is not detected as an EP by our method and 1 if it is detected as an EP. For each parameter combination, we simulate 100 data matrices and apply our discovering method (including the pre-screening step) to these 100 matrices successively. For each simulated data matrix, we define

n_{ed}: number of detected EPs

$n_{\overline{e}d}$: number of non-EP gene pairs which were detected as EPs

$n_{e\overline{d}}$: number of EPs which have not been detected

$n_{\overline{ed}}$: number of non-EP gene pairs which were not detected as EPs

If an EP of type 1 resp. 2 is diagnosed as EP of type 2 resp. 1, the EP is not considered as detected: it will be counted in $n_{e\overline{d}}$. For each parameter combination, we are interested in the contingency table

	d	$\overline{d}$	Σ
e	$med(n_{e,d})$	$med(n_{e,\overline{d}})$	n_{EP}
$\overline{e}$	$med(n_{\overline{e},d})$	$med(n_{\overline{e},\overline{d}})$	$p(p-1)/2 - n_{EP}$

where med is the median over the 100 random data matrices. We define the hit rate as the median proportion of detected EPs among the n_{EP} real EPs

$$\frac{med(n_{e,d})}{n_{EP}}$$

Similarly, the false alarm rate is defined as the median proportion of pairs which were discovered as EPs among the non-EP pairs

$$\frac{med(n_{\overline{e},d})}{p(p-1)/2 - n_{EP}}$$

The hit rate and the false alarm rate are given in Table 4 for different β, p_S and p_G. The boxplots of n_{ed}/n_{EP} and $n_{\overline{e}d}/n_{EP}$ are represented in Figure 4 and 5. They reveal some interesting features. First, β does not seem to have much influence on both rates, which indicated that a high β can select most important genes. Second, both the false alarm rate and the hit rate seem to decrease when p_S is increases. Third, a high p_G parameter seems to decrease the false alarm rate sensibly but not the hit rate. Thus, the main conclusion of these simulations is that p_G should be quite high, but that the other parameters do not have any great influence on the accuracy of the discovering method. However, the influence of β is expected to be greater in real data sets, where the distinction between 'good' and 'bad' genes is not that clear.

3.2 Results of the classification method on real data sets

Study design

We tested our classification method by dividing randomly the data set into a learning set $\mathcal{L}$ and a test set $\mathcal{T}$, following the procedure described in (Dudoit *et al.*, 2002). Since our method is very sensitive to the size of the training set, we chose to use test sets containing only 10 observations. The entire procedure including pre-screening is repeated 50 times. For each parameter combination, we used the same partitions, so as to eliminate possible causes of variations. For each data set, we give the mean classification error rate over the 50 runs in a table. If the value is marked with a star, it indicates that for at least one partition, no EP was found, thus making the discrimination impossible. In this case, the mean is calculated using only the partitions that yielded at least one EP.

In our study, we set $\alpha = 0.1$ (first pre-screening parameter) and $p_S = 10^{-4}$ (confidence level for the testing of the second splitting). β (second pre-screening parameter) and p_G (confidence level for the global testing of the leaves) can vary.

Colon data set

The colon data set (Alon, 1999) is an Affymetrix data set containing 2000 genes for 22 normal and 40 cancer samples. The 2000 genes are already selected genes (the authors do not explain how). We carried out a base 10 logarithmic transformation and normalized the samples to mean 0 and variance 1. Furthermore, 3 genes appear 4 times in the data set, with exactly the same expression levels for all 62 samples. We eliminated these lines in the data matrix.

Table 2 contains the mean error rate over the 50 partitions for different values of p_G and β. As expected, the number of found EPs depends highly on both parameters β and p_G. It seems that the classification accuracy increases when both β decreases and p_G increases. The fact that a low β increases the accuracy is not surprising, since more potential good genes are then selected. The fact that low p_G decreases the accuracy is more difficult to explain, since a stronger selection criterion for the EPs should prevent the selection of unrelevant EPs. A possible explanation is the robustness of classifiers based on more EPs. Indeed, the boxplots representing the number of found EPs (Figure 6) shows that for high p_G this number is very low and very variable.

Leukemia data set

The leukemia data set (Golub, 1999) is an Affymetrix data set containing the expression levels of 7129 genes for 72 leukemia patients. We put the training set (38 samples: 27 ALL and 11 AML) together with the test set (34 samples: 20 ALL and 14 AML). We applied the usual pre-processing method as described in (Dudoit *et al.*,2002): thresholding at 100 and 16000, filtering (thus obtaining 3571 genes left), base 10 logarithmic transformation, standardization of each sample to mean 0 and variance 1.

Table 4 contains the mean error rate over the 50 partitions for different values of p_G and β. As for the colon data set, the classification accuracy increases with p_G. But low values for β do not seem to yield much better accuracy. The number of found EPs is less dependent on the parameter β than for the colon data set, as can be seen in Figure 7.

Comparison with other supervised learning methods

Using the same study design, we tested 3 of the most usual classification methods: diagonal linear discriminant analysis as described in (Dudoit *et al.*,2002), nearest-neighbors with $k = 3$ and Support Vector Machine (SVM). We chose these 3 methods because DLDA and kNN gave very good when applied to several microarray data sets (Dudoit *et al.*,2002), and SVM is also believed to give very good results (Furey *et al.*,2000). For the k-nearest-neighbor method, we used the `R` programm `knn` from the library `class` and chose the euclidian distance as distance metric. For support vector machines, we used the `R` programm `svm` from the library `e1071`. Since these methods work much better with few genes, we performed a preliminary gene selection using the robust Wilcoxon-statistic, because it gave better results that the other simple selection methods we tried (t-statistic, Pearson's correlation and Ben-Dor's $TNoM$ score). The results are in Table 3 for the colon data and in Table 5 for the leukemia data. (leukemia). For the colon data set, the results are a bit better than the results with our method. For the leukemia data set, our results are better. Thus, our classification results are similar to those of the best usual methods. In addition, our method allows to discover interesting structures at the same time, which is actually a great advantage over the usual approaches which do nothing more than filtering highly differentially expressed genes and classifying the samples.

3.3 Analysis of the EPs found in the colon data set

Here the discovering algorithm is run on the whole data set. We set β to 0.3 and give in table 6 only the EPs with a p-value lower than 10^{-11}, which are the most significant. The numbers of EPs of type 1 and 2 are approximately equal, which is quite a good thing for classification purpose. Another interesting point is that not all the genes involved in these EPs are good individually (data not shown), especially the genes involved in the second splitting. Thus, we conclude that it is restrictive to select genes on the basis of some individual score like the t-statistic or the Wilcoxon-statistic and consider all the other genes as uninteresting.

Our EPs for the colon data set are very different from Li and Wong's. Several reasons can be put forward. First, Li and Wong looked only for what could be referred as 'perfect EPs', i.e. EPs with infinite growth rate. From the statistical point of view, it makes sense to consider non-perfect EPs as well, especially when having to do with noisy data like microarray data. Second, the EPs they found are quite long and possibly containing highly correlated genes. For instance, in an EP containing 7 genes, some of the genes probably do not bring much, just eliminating one or two observations. There is no reason why this should also be the case with an independent data set. Thus, our EPs are shorter, because we prefer to focus on the statistical relevance and reproducibility. Last, Li and Wong performed a pre-screening leaving only 35 genes, which is quite restrictive. In particular, the pre-screening eliminates genes which do not perform well individually. Since EPs are made of several genes interacting together, it is questionable whether such a dramatic pre-screening is suited. Indeed, in our emerging patterns, we noticed that the gene involved in the second splitting is not always a gene which performs good individually.

4 Concluding remarks

In summary, emerging patterns are a valuable tool for supervised learning as well as for exploring interactions between genes. Our quite intuitive and fast CART-based method allows to discover a large proportion of them, with focus on statistical aspects which had been overseen so far. Additionally, we demonstrated that it is not necessary to perform a strong pre-screening before classification and that some genes performing poorly individually may be useful in association with other genes. It would be of great interest to run our discovering algorithm on data sets with more observations, thus allowing EPs of greater order. A related issue is the study of the statistical relevance in the multiple testing framework. Another important topic is the generalization to the multi-class case which often occurs in practice.

Acknowledgements

We thank Korbinian Strimmer for critical comments and discussion.

References

[1] Agresti,A. (2002), Categorical Data Analysis, *Wiley Series in Probability and Mathematical Statistics*

[2] Alon,U., Barkai,N., Notterman,D.A., Gish,K., Ybarra,S., Mack,D., Levine,A.J. (1999), Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays, *Cell Biology*, **96**, 6745 – 6750

[3] Ben-Dor A., Bruhn L., Friedman N., Nachman I., Schummer M., Yakhini Z. (2000), Tissue Classification with Gene Expression Profiles

[4] Breiman L., Friedman J., Olshen R., Stone C. (1984), Classification and regression trees, *Wadsworth, Belmont, CA*

[5] Dong,G., Li,J. (1999), Efficient Mining of Emerging Patterns: Discovering Trends and Differences, *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, ACM Press, San Diego, CA, pp.43 – 52

[6] Dudoit,S., Fridlyand,J., Speed,T.P. (2002), Comparison of Discrimination Methods for the Classification of Tumors using Gene Expression Data, *JASA*, **97**, No.457, 77 – 87

[7] Furey,T., Cristianini,N., Duffy,N., Bednarski,D.W., Schummer,M., Haussler, D. (2000), Support vector machine classification and validation of cancer tissue samples using microarray expression data, *Bioinformatics*, **16**, 906 – 914

[8] Golub,T.R., Slonim,D.K., Tamayo,P., Huard,C., Gaasenbeek,M., Mesirov,J.P., Coller,H., Loh,M.L., Downing,J.R., Caligiuri,M.A., Bloomfield,C.D., Lander,E.S. (1999), Molecular Classification of Cancer: Class Discovery and Class Prediction by Gene Expression Monitoring, *Science*, **286**, 531 – 537

[9] Kahn,J., Wei,J.S., Ringner,M., Saal,L.H., Ladanyi,M., Westermann,F., Berthold,F., Schwab,M., Antonescu,C.R., Peterson,C., Meltzer,P.S. (2001), Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks, *Nature Medicine*, **7**, Number 6

[10] Li,J., Wong,L. (2001), Emerging Patterns and Gene Expression Data, *Genome Informatics*, **12**, 3 – 13

[11] Li,J., Wong,L. (2002), Identifying good diagnostic gene groups from gene expression profiles using the concept of emerging patterns, *Bioinformatics*, **18**, 725 – 734

[12] Li,J., Liu,H., Downing,J.R., Yeoh,A.E., Wong,L. (2003), Simple rules underlying gene expression profiles of more than six subtypes of acute lymphoblastic leukemia (ALL) patients, *Bioinformatics*, **19**, 71 – 78

[13] Singh,D., Febbo,P.G., Ross,K., Jackson,D.G., Manola,J., Ladd,C., Tamayo,P., Renshaw,A.A., D'Amico,A.V., Richie,J.P., Lander,E.S., Loda,M., Kantoff,P.W., Golub,T.R., Sellers,W.R., (2002), Gene expression correlates of clinical prostate cancer behavior, *Cancer Cell*, **1**

[14] Tutz,G. (2000), Die Analyse kategorialer Daten, *Oldenbourg Verlag*

β	p_G	p_S	hit rate	false alarm rate
0.3	10^{-16}	10^{-4}	0.55	$4.1 \cdot 10^{-5}$
0.4	10^{-16}	10^{-4}	0.55	$4.2 \cdot 10^{-5}$
0.3	10^{-18}	10^{-4}	0.55	$2.8 \cdot 10^{-5}$
0.4	10^{-18}	10^{-4}	0.50	$2.8 \cdot 10^{-5}$
0.3	10^{-20}	10^{-4}	0.45	$2 \cdot 10^{-5}$
0.4	10^{-20}	10^{-4}	0.45	$2 \cdot 10^{-5}$
0.3	10^{-16}	10^{-8}	0.45	$2.9 \cdot 10^{-5}$
0.4	10^{-16}	10^{-8}	0.50	$2.9 \cdot 10^{-5}$
0.3	10^{-18}	10^{-8}	0.45	$1.8 \cdot 10^{-5}$
0.4	10^{-18}	10^{-8}	0.45	$2 \cdot 10^{-5}$
0.3	10^{-20}	10^{-8}	0.425	$1.2 \cdot 10^{-5}$
0.4	10^{-20}	10^{-8}	0.40	$1.2 \cdot 10^{-5}$

Table 1: Hit rate and false alarm rate for various parameter combinations

	$p_G = 10^{-8}$	$p_G = 10^{-9}$	$p_G = 10^{-10}$
$\beta = 0.3$	0.134	0.128	0.158
$\beta = 0.4$	0.138	0.146	$0.165*$
$\beta = 0.5$	0.146	0.158	$0.206*$
$\beta = 0.6$	0.192	0.206	$0.234*$

Table 2: Mean error rate over the 50 runs for the colon data with our method

	LDA	$3-\text{NN}$	SVM
10 genes	0.120	0.124	0.118
20 genes	0.122	0.152	0.122
50 genes	0.122	0.164	0.114
100 genes	0.126	0.150	0.122
200 genes	0.128	0.160	0.122

Table 3: Mean error rate over the 50 runs for the colon data with usual classification methods

	$p_G = 10^{-13}$	$p_G = 10^{-14}$	$p_G = 10^{-15}$
$\beta = 0.4$	0.028	0.028	0.044
$\beta = 0.5$	0.026	0.030	0.046
$\beta = 0.6$	0.024	0.030	0.038
$\beta = 0.7$	0.026	0.032	$0.049*$

Table 4: Mean error rate over the 50 random partitions for the leukemia data with our method

	LDA	$3-\text{NN}$	SVM
10 genes	0.040	0.044	0.048
20 genes	0.030	0.036	0.040
50 genes	0.028	0.040	0.052
100 genes	0.034	0.044	0.042
200 genes	0.032	0.044	0.036

Table 5: Mean error rate over the 50 random partitions for the leukemia data with usual classification methods

Gene 1	Gene 2	Freq. in D_1	Freq. in D_2
$H06524 \in [-0.54, +\infty)$	$Z50753 \in [0.16, +\infty)$	1	0.075
$H11084 \in [-\infty, 0.33)$	$Z50753 \in [-0.55, +\infty)$	0.91	0.025
$U04953 \in [-\infty, 0.07)$	$M63391 \in [1.17, +\infty)$	0.86	0
$R81330 \in [-0.45, +\infty)$	$R36977 \in [-\infty, -0.08)$	0.91	0.05
$M82919 \in [-1.05, +\infty)$	$X12369 \in [0.49, +\infty)$	0.82	0
$T51493 \in (-\infty, -0.77]$	$R64115 \in (-\infty, 0.58]$	0.91	0.05
$T64467 \in [0.67, +\infty)$	$H72234 \in (-\infty, -0.10]$	0.82	0
$U04953 \in (-\infty, 0.07]$	$R60883 \in [-0.38, +\infty)$	0.91	0.025
$T64467 \in [0.67, +\infty)$	$T51493 \in (-\infty, -0.72]$	0.82	0
$R55310 \in [0.32, +\infty)$	$U09564 \in (-\infty, -0.15]$	0.82	0
$R55310 \in [0.32, +\infty)$	$H72965 \in (-\infty, -0.51]$	0.86	0
$L38810 \in (-\infty, 1.48]$	$M76378 \in (\infty, 1.19]$	0	0.9
$X87159 \in (-\infty, 0.68]$	$X63629 \in [-0.90, +\infty)$	0	0.875
$D14812 \in [0.20, +\infty)$	$U25138 \in (-\infty, -0.44]$	0.14	0.975
$T62947 \in [-1.06, +\infty)$	$M76378 \in (-\infty, 1.18]$	0	0.875
$T62947 \in [-1.12, +\infty)$	$H20709 \in (-\infty, 2.80]$	0.05	0.925
$T41207 \in (-\infty, -0.11]$	$T92451 \in (-\infty, 1.94]$	0.05	0.925
$T71025 \in (-\infty, 2.19]$	$H08393 \in [-1.19, +\infty)$	0	0.875
$M91463 \in (-\infty, -0.59]$	$R44418 \in (-\infty, 0.54]$	0.14	0.975

Table 6: Emerging Patterns for the colon data set

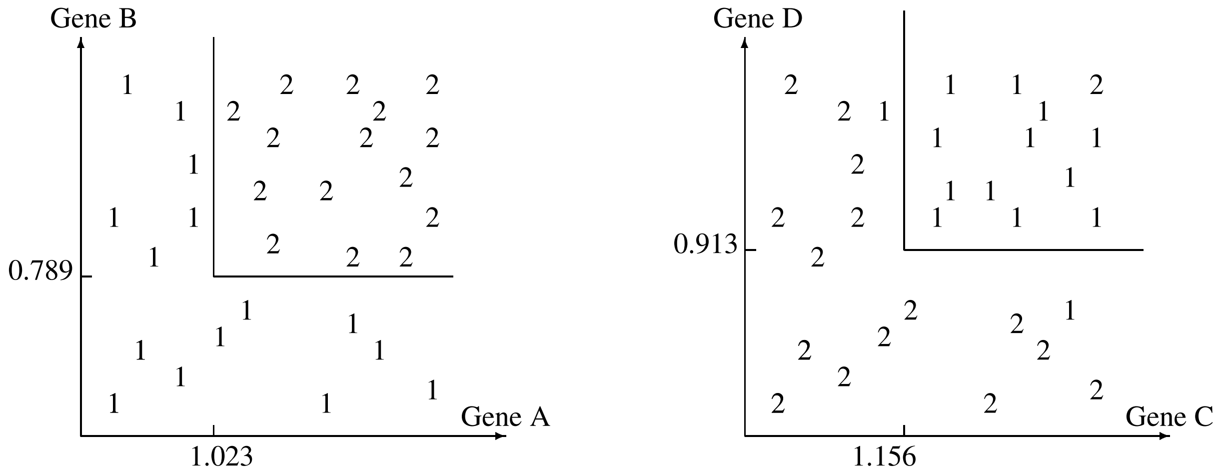

Figure 1: Examples for possible configurations for two genes with ’2’ denoting cancer tissue and ’1’ normal

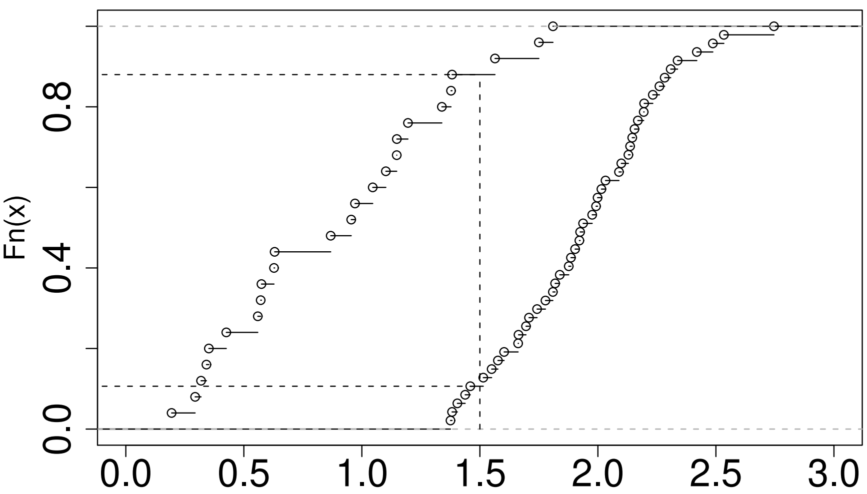

Figure 2: Empirical distribution of gene 456 (from the leukemia data set) in class 1 and class 2

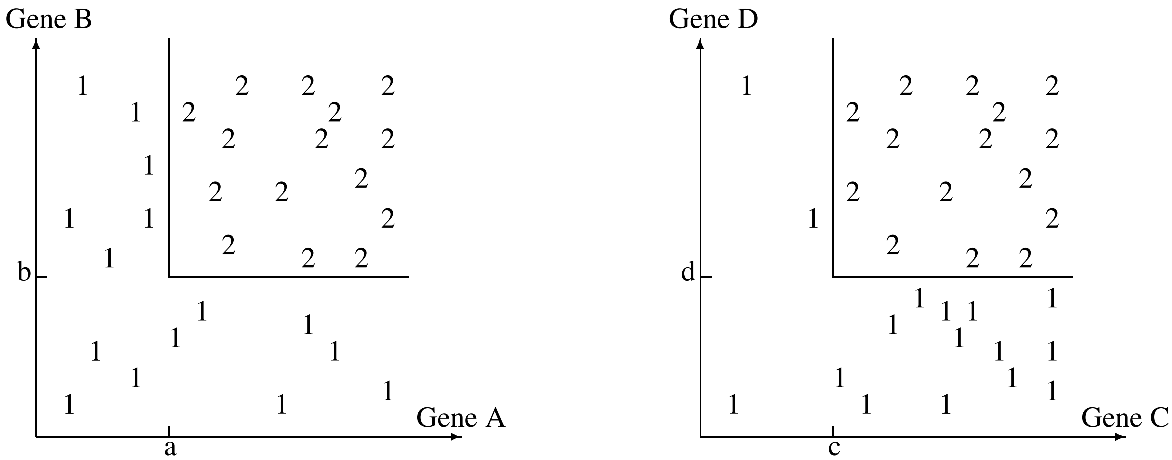

Figure 3: Examples of a relevant EP and an unrelevant EP

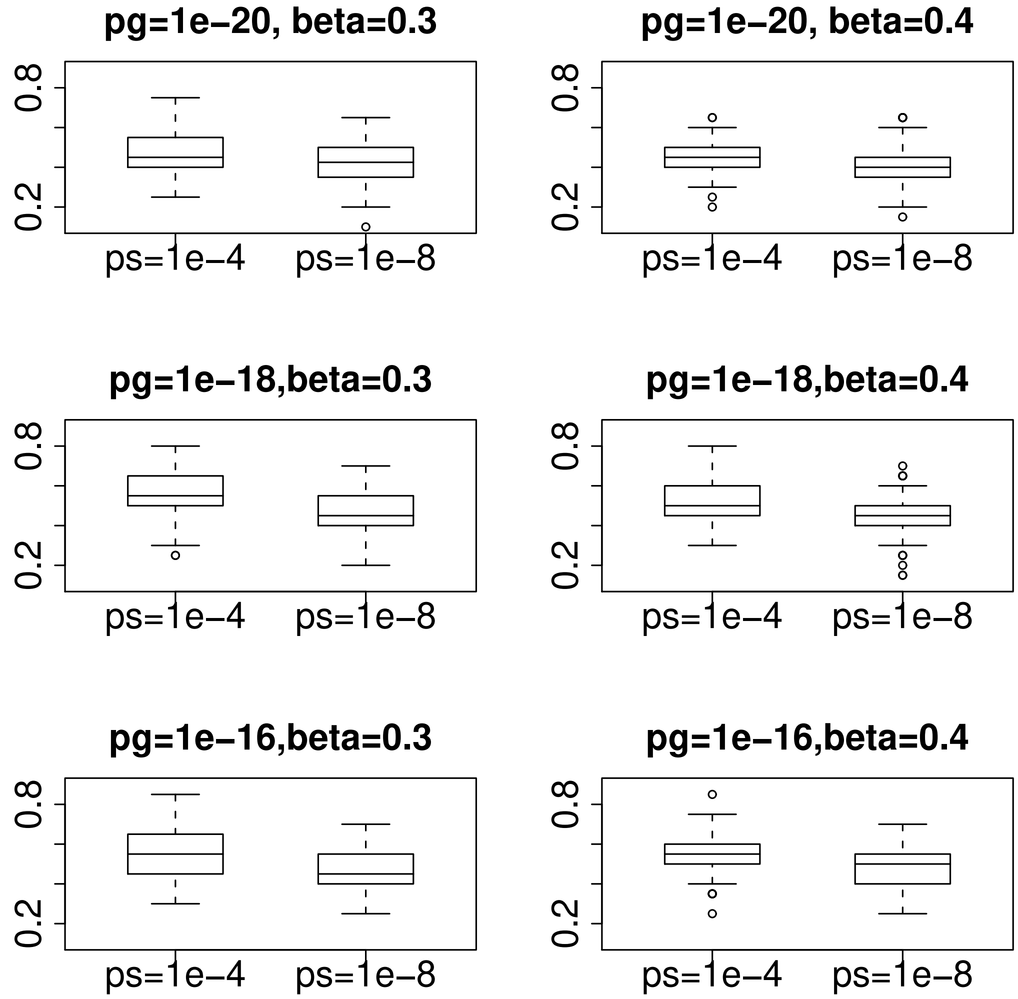

Figure 4: Boxplots of the discovery rate over the 100 random data matrices for different parameter combinations

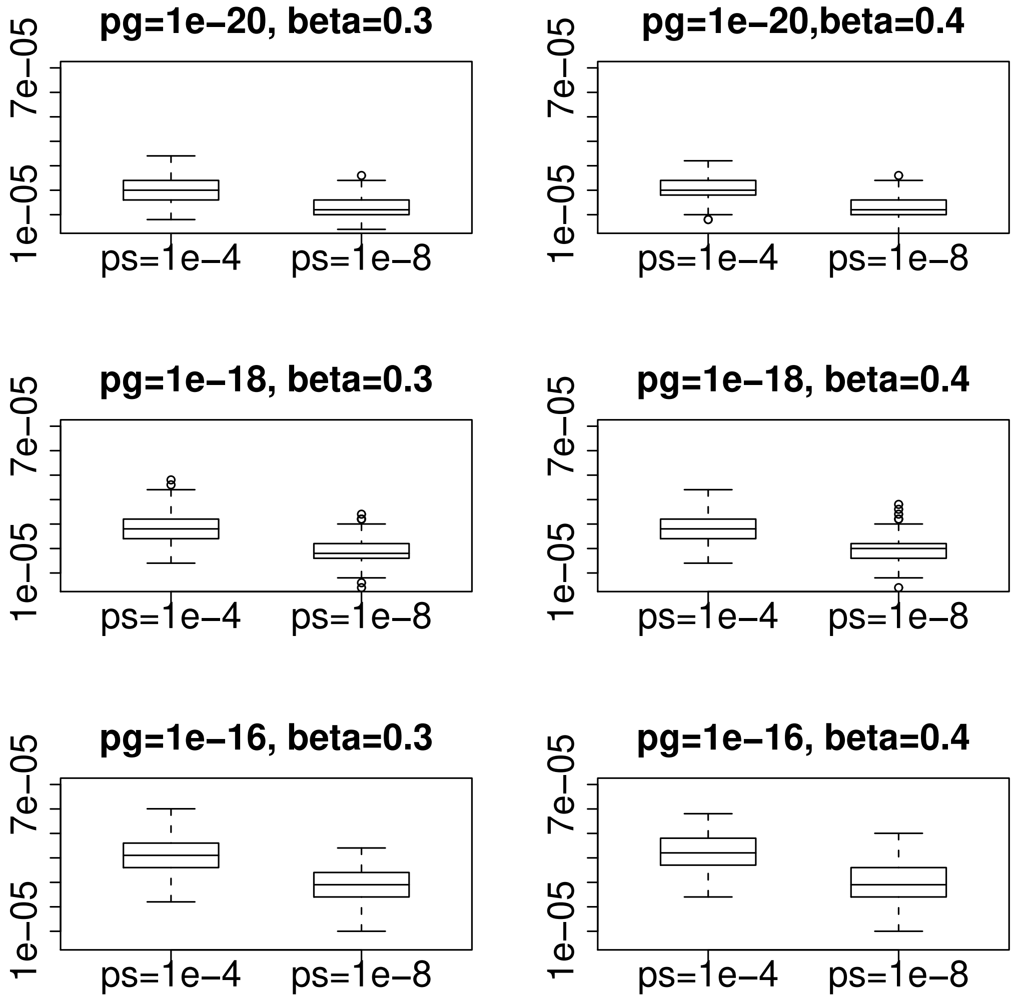

Figure 5: Boxplots of the false alarm rate over the 100 random data matrices for different parameter combinations

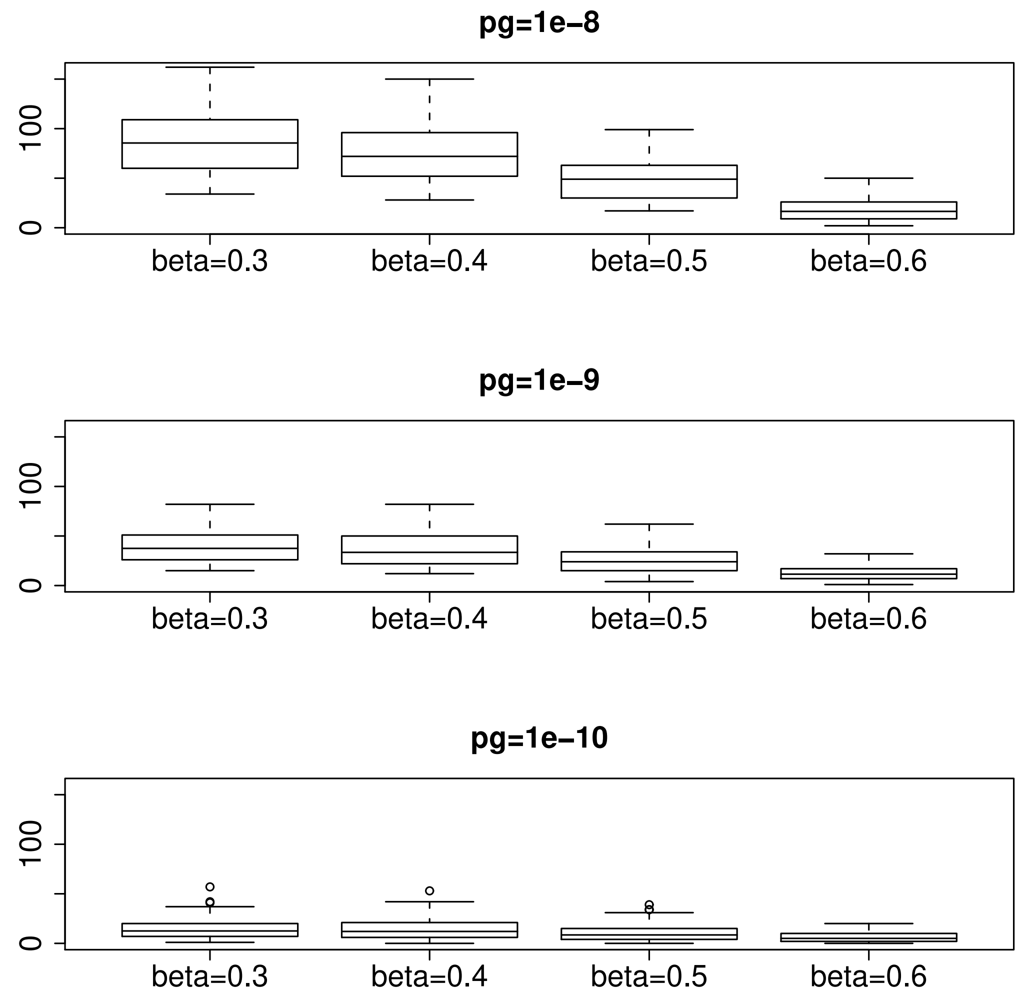

Figure 6: Boxplots of the number of found EPs over the 50 runs for the colon data set

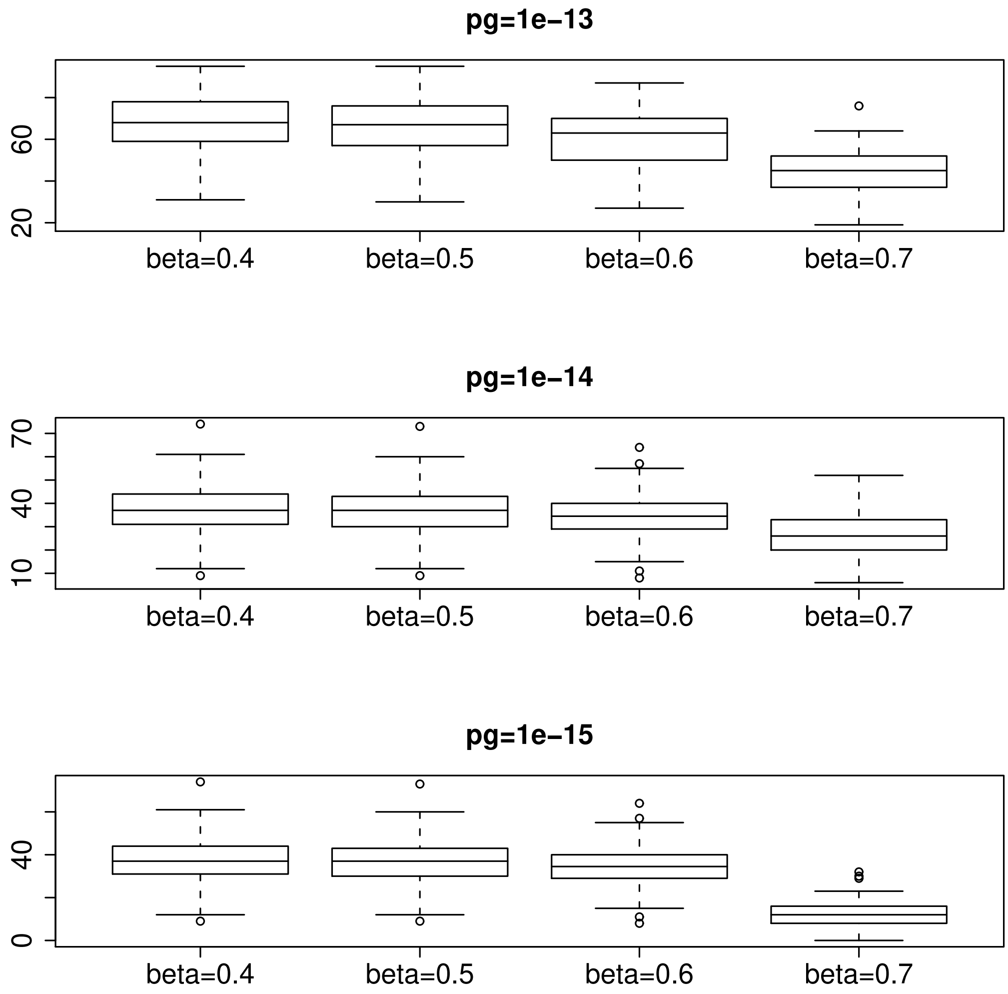

Figure 7: Boxplots of the number of found EPs over the 50 runs for the leukemia data set