The robust selection of predictive genes via a simple classifier

Veronica Vinciotti[1*], Allan Tucker[1*], Paul Kellam[2] and Xiaohui Liu[1]

1.School of Information Systems, Computing and Mathematics, Brunel University, Uxbridge UB8 3PH, UK.

2.Virus Genomics and Bioinformatics Group, University College London, London W1T 4JF, UK.

Email: veronica.vinciotti@brunel.ac.uk

Phone/Fax: +44 (0)1895 265986/ +44 (0)1895 251686

* These authors contributed equally to this work.

Abstract

Identifying genes that direct the mechanism of a disease from expression data is extremely useful in understanding how that mechanism works. This in turn may lead to better diagnoses and potentially can lead to a cure for that disease. This task becomes extremely challenging when the data are characterised by only a small number of samples and a high number of dimensions, as it is often the case with gene expression data. Motivated by this challenge, we present a general framework that focuses on simplicity and data perturbation. These are the keys for the robust identification of the most predictive features in such data. Within this framework, we propose a simple selective naïve Bayes classifier discovered using a global search technique, and combine it with data perturbation to increase its robustness to small sample sizes.

An extensive validation of the method was carried out using two applied datasets from the field of microarrays and a simulated dataset, all confounded by small sample sizes and high dimensionality. The method has been shown capable of identifying genes previously confirmed or associated with prostate cancer and viral infections.

Keywords: Feature selection, naïve Bayes classifier, simulated annealing, cross-validation, microarray data.

1 INTRODUCTION

Microarray technology [1] provides an extreme example of small sample-size data, which is further confounded by very high dimensionality. Given the high costs of a single experiment, gene expression data are characterised by many dimensions, involving up to thousands of genes, but only a handful of observations, the biological samples. Therefore, when training classifiers from these data in order to identify expression patterns for a disease or a biological con-

dition, it is extremely difficult not only to distinguish samples from different conditions but also to pinpoint the genes that mostly contribute to a particular condition. This problem has been pointed out by various authors in the context of gene expression data, e.g. [2, 3].

In this paper, we develop a method for extracting a reasonable number of the most predictive genes in a biological sample, within a classification framework tailored to small-sample and high-dimensional gene expression data. This will aid biological knowledge about the distinctive features between the classes as well as help with the design of the next experiments by identifying the genes of interest. These results can then be translated in a clinical setting by defining genes that are diagnostic or prognostic of a clinical condition and for identifying mechanism of a disease. In all the steps involved in developing our method, simpler solutions will be advocated in order to increase the generalisability of the classifier as well as the robustness of the extracted features.

The first step in the direction of making the most of small-sample high-dimensional data is the choice of a *simple* classifier, that is a classifier which is defined by a small number of parameters. Despite the often unrealistic assumptions behind simple classifiers, the smaller number of parameters needed to characterise them reduces the risk of overfitting the data and, as such, can bring an increase in the predictive accuracy of the classifier on samples outside the training data [4]. This fact is supported also by studies in the context of gene expression data, where simpler classifiers have often shown superior performance to more complex classifiers [5, 6, 7]. In this paper, we explore and extend the naïve Bayes classifier, the simplest classifier within the family of Bayesian networks. This classifier has been used on various applications, including gene expression data, due to its simplicity, transparency and efficiency, e.g.[8, 9].

The extremely high dimensionality of gene expression data means that a di-

mensionality reduction method is needed in combination with the classifier in order to improve its classification accuracy as well as to identify the most predictive genes. A traditional way of dealing with high dimensionality is by using feature selection methods, where a subset of features is selected by optimizing some well-defined criterion [10]. Various feature selection methods have been suggested for gene expression data. Some of them use optimization criteria that are independent of the classification model used, the so called filter methods, some others use the classification accuracy, or other model-based scores, as the optimization criterion, the so called wrapper methods, e.g. [5, 6, 11, 12, 13, 14, 15]. A few studies have attempted a comparison between different feature selection methods on gene expression data [16, 17, 18].

In our experiments we use a wrapper feature selection approach where the criterion for the selection of the features is based on the likelihood of the corresponding selective naïve Bayes classifier. The likelihood itself is the most common scoring metric for Bayesian networks. However, this measure is prone to overfitting by leading to overly-complex models. As our aim is to develop a simple model by identifying only a small subset of predictive genes, we limit the feature selection method based on this score to networks with a maximum of k links. A downside of this approach, and indeed of many feature selection approaches, is that the choice of k can be difficult and computationally expensive. An alternative, which does not require to fix the number of features k a priori, is to use a score function that penalises the inclusion of many links. To this end, we explore the use of a network score based on the Minimum Description Length (MDL) to select the optimal subset of features. This is similar to the Bayesian variable selection approach of [19] in the context of a multinomial probit regression model: here a prior on the number of predictive features is chosen which penalises complex models. An MDL-based score has been previously used

in the context of gene expression data and classification by [20] to select a small number of cluster centroids that best discriminate between the sample classes.

When only few samples are available, care should be taken also in validating and assessing the model. When assessing the performance of the classifier, one has to make sure that the data used to select the features and train the classifier are not used also at the testing stage [2]. In this paper, we use cross-validation to measure the performance of our selective naïve Bayes classifier. Furthermore, to increase the robustness of the method, we repeat the feature selection method based on a stochastic optimization search a number of times. As a consequence of this, we are able to assign a confidence measure to the various features and draw more reliable conclusions about the most predictive features.

To summarise, a robust selective naïve Bayes classifier is proposed for feature selection. This method builds on a simple selective naïve Bayes classifier, but using simulated annealing to optimize a score function that favours simpler models. In addition, the method uses data perturbation in the form of cross-validation to increase the robustness in the selected features. Section 2 describes the proposed method in detail. Section 3 describes the datasets used in the analysis. In Section 4 we test the classification accuracy of the method and in Section 5 we investigate the quality of the discovered features on two biological datasets.

2 METHOD

Let $x = (x_1 \cdots x_N)$ be the vector of discretized expression levels for the N genes on one sample. By expression, we mean the log(intensity) of a gene relative to a common reference signal. Each variable of gene expression is discretized into s states, with s chosen sufficiently small to limit the number of parameters in the model. This avoids assuming an overly restrictive continuous distribution over

the gene variables and has the potential of capturing non-linear relationships. Let c be the associated class, e.g. the presence or absence of a virus or disease in the sample. Despite considering the case of binary classes in our experiments, the methodology described in this paper is not restricted to this special case. Given a training set containing a number of samples and their known true classes, the task is to find a rule that will automatically classify a new sample x to its unknown class.

A naïve Bayes classifier makes the simplifying assumption that each feature is independent of each other given the class. This corresponds to the network structure in Figure 1 and to the efficient factorization $p(x|c) = \prod_{i=1}^{N} p(x_i|c)$.

*****FIGURE 1 ABOUT HERE ******

Under the assumption of uniform priors, a Bayesian estimate of $p(x_i|c)$ is given by

$$\hat{p}(x_{ip}|c) = \frac{1 + n(x_{ip}|c)}{s + n(c)},$$

where s is the number of discretized states of the gene variable X_i, $n(x_{ip}|c)$ is the number of cases in the dataset where X_i takes on its pth unique state within the samples from class c, and $n(c) = \sum_{p=1}^{s} n(x_{ip}|c)$ is the total number of samples from class c. From $\hat{p}(x|c)$, an estimate of $p(c|x)$ is calculated using Bayes rule and the resulting classification rule assigns the sample x to the class associated to the highest estimated probability.

In this paper, we simplify the naïve Bayes structure further by only including features (i.e. links in Figure 1) which result in high scoring networks. As a first approach, we use the standard likelihood score of a network [21] to compare networks associated to different subsets of features. For our

problem, this is given by

$$\text{Lik-score} = \prod_{i=1}^{N} \prod_{c=1}^{q} \frac{(s-1)!}{(s+n(c)-1)!} \prod_{p=1}^{s} n(x_{ip}|c)! \tag{1}$$

where q is the number of classes. As this score does not inherently penalise larger sets of selected links versus smaller ones, we constrained the feature selection search to a maximum of k links.

An alternative to this is to modify the score by including a penalization term to the likelihood function. To this end, we have explored the use of an MDL-based score to select the optimal subset of features in a selective naïve Bayes classifier. This score has been previously used to learn Bayesian networks [22, 23]. The main idea behind it is to compute the description length of the network as the sum of the description length of the model and the description length of the data given the model. For our problem, this is given by

$$\begin{aligned} \text{MDL-score} &= \text{DL}_{Model} + \text{DL}_{Data} \\ \text{DL}_{Model} &= 2N \log(N+1) + q(s-1)N \frac{\log M}{2} \\ \text{DL}_{Data} &= M \sum_{i} H(X_i|C) \end{aligned} \tag{2}$$

where $H(X_i|C) = -\sum_{p,c} n(x_{ip}, c) \log n(x_{ip}|c)$ and M is the number of biological samples. More complex networks are expected to lead to a shorter description of the data given the model at the expense of a longer description needed to describe the model itself. In this way, the score (2) provides a trade-off between complexity of the model and goodness-of-fit to the data. Furthermore, the factor $\frac{\log M}{2}$ in equation (2) is a penalty term based upon the sample size [24]. For datasets where the sample size is particularly small, this penalty will have a more pronounced effect.

The next step in our method is to efficiently find the network or set of fea-

tures that optimizes the score (1) or (2). We make use of a simulated annealing approach [25] to find the set of features that maximizes the network score. This *global* optimization search was made possible by the simplicity of the model and was chosen with the aim of efficiently tackling the problem of high dimensionality without suffering from local optima, which many greedy searches might suffer from. These include for example the selective naïve Bayes of [26] and the sequential hill-climb approach of [12]. The main idea behind our method is to make small changes to the classifier structure and then score the network. The changes involve using three operators, `add`, `delete` and `swap`, which randomly add a link, remove a link and swap a link in the classifier, respectively. These changes can be constrained to networks with a maximum of k links when using the likelihood score (1). In contrast to this, when the MDL score (2) is used, k is automatically selected. Finally, note that if there are no links in the network then only the `add` operator can be applied whereas if the number of links is equal to k then only `swap` and `delete` can be applied.

The optimization algorithm is documented fully below, where D represents the input data, the initial annealing temperature is denoted by t_0, the cooling parameter for the temperature by c, the maximum number of scoring function calls by `maxfc` and the score of a network by `score(nbc)`, computed either by the log of the likelihood in (1) or by the MDL in (2). $R(0,1)$ is a uniform random number generator with limits 0 and 1.

```
Input:  t0,maxfc,D
        fc = 0, t = t0
        Initialise nbc to a naïve Bayes
        classifier with no links
        result = nbc
        oldscore = score(nbc)
        While fc ≤ maxfc do
          For each operator do
                Apply operator to nbc
                newscore = score(nbc)
```

```
                fc = fc + 1
                dscore = newscore − oldscore
                If newscore > oldscore then
                   result = nbc
                Else If R(0, 1) < e^(dscore/t) Then
                   Undo the operator
                End If
                End For
            t = t × c
        End While
Output: result
```

Algorithm 1: Simulated annealing for feature selection in naïve Bayes.

For all our experiments, we set t_0 to 1. This was based upon the initial scores when applied to the two datasets investigated in this paper (we have generally found that a good starting temperature is similar to the changes in score in the early iterations) and allowed upward transitions in the early and mid-stages of the learning procedure. `maxfc` was set to 10000 as this was found through empirical analysis to ensure that convergence has occurred on the datasets explored. c was set to 0.999, calculated to make the temperature after `maxfc` iterations suitably close to zero in order to ensure that changes resulting in a worse solution would no longer be retained.

We use m-fold cross-validation to assess the performance of the selective naïve Bayes classifier. Furthermore, for each fold, we repeat the feature search 10 times, due to the stochastic nature of our simulated annealing (SA) algorithm. For each of these runs, the frequency count is maintained for each link in all networks generated on the training data for the corresponding fold and the classifiers tested on the portion of data taken out. In this way we are able to produce a confidence measure for each feature in the dataset based on the different training samples generated by each fold, leading to a more robust detection of the most predictive features. This is similar to the method used by [27], where the confidence measure on links in a Bayesian network is achieved

by bootstrapping the data.

We call this method the Robust Selective Naïve Bayes classifier (RSN) to emphasise the combination of the feature selection procedure with the confidence measure associated to each feature. The algorithm below gives a description of the overall procedure.

```
Input:  m,r,D,M
        Initialise all counts in p to 0
        and accuracy to 0
        For i = 1 to m
           D' = D with M/m samples randomly
           removed and placed into U'
           For j = 1 to r
             Apply SA on D' to learn NBC links
             For each gene g discovered in NBC
                p[g] = p[g] + 1
             End g
             Use U' as test set to assess
             the performance of the classifier
                accuracy = accuracy + accuracy[U']
           End j
        End i
        p = p/(m * r), accuracy = accuracy/(m * r)
Output:  p, accuracy
```

Algorithm 2: Robust Selective Naïve Bayes classifier

where m is the measure of perturbation (e.g. 10 to randomly delete a tenth of the data at a time), M is the number of samples, D is the dataset, *accuracy* gives the average performance of the classifier using m-fold cross-validation, where each fold involves r repeats of the stochastic search algorithm (here SA), and p is a vector of counts for each gene, that is the proportion of times that a gene has been selected by the RSN method.

3 THE DATA

In this section, we describe the datasets used to validate and assess the RSN method. These include two real datasets from the field of gene expression data and a simulated dataset.

3.1 Prostate Cancer

This dataset has recently been included in the Stanford Microarray Database and is described in [28]. It consists of 112 samples, 41 of which are from normal prostate specimens and the remaining 71 from primary prostate tumours and lymph nodes metastases. We used filtered data containing 1410 genes.

For the RSN classifier, each variable of gene expression was discretised into four states using a frequency-based method whereby each resulting discrete state appears an equal number of times.

3.2 B-cell Lymphomas

This dataset consists of a series of 26 arrays from 584 filtered genes measuring gene expression difference across a set of human B-cell lymphomas and leukaemias [29]. The 584 genes were filtered from a total of 1987, by removing the genes with missing data. Each probe on the array detected a single gene transcript. For each sample, we know whether a virus is present or not. This is what we wish to classify. The data is equally balanced with 50% of cases having the virus present.

Prior to learning the RSN classifier, the data were discretised into two states, based upon whether the expression value was positive or negative. The choice of two states was decided for this dataset with the aim of keeping the number of parameters as low as possible (due to the extremely small number of samples).

3.3 Simulated Data

To assess the sensitivity and specificity of RSN, we make use of a simulated dataset. We have generated the data with the aim of reflecting the sort of features found within real microarray data. The dataset consists of 100 samples, 50 for each of the two classes, and 1000 features. For 30 of these features, we have drawn the gene expression profiles associated with the two classes from two different distributions, respectively, that is we have assumed that these features differentiate between the two classes. The two distributions were chosen to be normal, with mean and standard deviations randomly selected from a uniform distribution with range of [-1,1] and [0.1,1.5], respectively. The ranges were identified using the predictive genes found in the B-cell data, with the aim of reflecting the degree of overlapping between the two classes found in real datasets. The remaining 970 gene expression profiles were drawn from a single standard normal distribution, that is we have assumed that these features do not separate the two classes. The mean and variance of this distribution were also set approximately as the ones found in the B-cell data. Note that the choice of a normal distribution is appropriate for the microarray data in our studies, as the gene expression is reported in the log-scale.

4 RESULTS

We validate and assess the performance of the RSN classifier in different ways. Firstly, we explore the effect of reducing sample size on prediction accuracy by applying the method to the prostate cancer dataset, where a reasonable number of samples is available. Secondly, we use the simulated dataset to test how well our feature selection method detects the most predictive features in the data. Finally, we assess the classification accuracy of the method and compare it with

other standard feature selection methods for classification.

4.1 Exploring the effect of sample size on classification accuracy

We use the prostate cancer dataset to show the effect of sample size on accuracy of the RSN classifier when compared with the standard naïve Bayes classifier. We sequentially reduce the size of the data, by randomly deleting a tenth of the observation. At each step, a selective naïve Bayes classifier is learnt on the corresponding data and the accuracy estimated using Algorithm 2 (with $m, r = 10$). Figure 2 shows the classification accuracy of selective naïve Bayes models based on the likelihood score (1) and limited to a maximum of 5 links, 50 links, 500 links and the standard naïve Bayes (1410 links), respectively. It can be seen that the complexity of the model is detrimental to the classifier performance as sample size decreases.

*****FIGURE 2 ABOUT HERE ******

These results imply that a simple model is preferable when only a small sample is available as it reduces the risk of overfitting. This was the expectation and main motivation behind our proposed method.

4.2 Validating the quality of the selected features using simulated data

We use the simulated dataset in Section 3.3 to validate the power of our feature selection method. We have generated the data so that 30 genes were differentially expressed between the two classes. The hope is that these genes will be picked by our method out of the large number of redundant genes.

Figure 3 shows the Receiver Operating Characteristic (ROC) curves for both

the likelihood-based and MDL-based RSN (Lik-RSN and MDL-RSN, respectively).

*****FIGURE 3 ABOUT HERE ******

We build the ROC curve by considering the proportion of times a gene was selected as the probability of it being classed as one of the 30 pre-defined most predictive genes. Hence, sensitivity is the proportion of these 30 genes which have been correctly identified. We also compare the likelihood-based and MDL-based RSN with a stepwise version of the likelihood-based RSN (SW-RSN) and with the well known stepwise linear regression method (SW-LinReg). For the stepwise approaches, a greedy search algorithm based upon the forward selection method is used to detect the k features that optimize the corresponding score. The latter is given by the likelihood score (1) for the SW-RSN method and by the residual sum of squares error of the regression model built on the subset of features for the SW-LinReg method (implemented in the S+ function *stepwise*). The value of k was set to 5 based on experiments as shown in Figure 2. The SW-RSN method is similar to the selective naïve Bayes of [26] except that we do not use an accuracy-based wrapper method.

The plot shows that the global simulated annealing approaches perform better than the stepwise approaches, by finding more of the correct features whilst minimizing the detection of incorrect ones. This is likely to be due to the greedy search getting stuck in local optima which a simulated annealing optimization search manages to circumnavigate. The likelihood-based and MDL-based approaches both show a high accuracy in detecting the true differentially expressed genes, with the likelihood score slightly outperforming the MDL approach for this particular choice of k.

4.3 Classification accuracy of RSN

The results in the previous two subsections show that the RSN method manages to handle the extreme situation of small sample size due to its simplicity and is also able to single out the most predictive features in the dataset from the redundant ones due to its robustness. In this subsection, we apply the RSN method to two real gene expression datasets, namely the prostate cancer and the B-cell datasets described in Section 3. Together with these two datasets, we also generate a third dataset, by sampling 25% of the prostate cancer data (consisting of only 28 samples). The remaining 75% of the data were used as a test set to estimate the accuracy of the classifier. This was possible due to the large number of samples available for this dataset and was done with the aim of further testing the accuracy of the method on extremely small sample-size situations. For the other two datasets, we used leave-one-out cross validation on the B-cell dataset (i.e. $m = 26$ in Algorithm 2) and 10-fold cross-validation on the prostate dataset (i.e. $m = 10$ in Algorithm 2).

Table 1 shows the classification accuracy achieved by RSN using both the likelihood and MDL scores, along with a number of other well documented feature selection methods. These include the forward stepwise selective naïve Bayes (SW-RSN) and forward stepwise linear regression (SW-LinReg), previously described, and C5, an updated commercial version of the decision tree generator C4.5 [30].

*****TABLE 1 ABOUT HERE ******

From Table 1 it can be seen that RSN was better than C5 and SW-LinReg on all three datasets and comparable to the SW-RSN method overall. The relative accuracy of RSN (when using the MDL or likelihood score) is particularly good on the 25% sample of the Prostate dataset. Note that the ratio between the number of variables and the number of observations is particularly extreme in

this dataset, with only 28 samples and 1410 predictors. The likelihood-based and MDL-based RSN methods are comparable under the particular choice of $k = 5$ used for the likelihood score. This can be seen as an advantage to the MDL score, as this does not require the value of k to be chosen a priori.

In addition, we have explored the use of a Tree Augmented Network (TAN) classifier [31] rather than the naïve Bayes classifier within the same framework of the MDL-RSN method. Here the naïve Bayes structure is augmented by allowing also dependencies between the gene variables. This was done with the aim of testing whether relaxing the assumption of independence of the naïve Bayes classifier improves the classification accuracy of the method. The method achieved an accuracy of 73% and 97% on the B-cell and Prostate datasets, respectively. This shows no significant improvement with the RSN method under the constraint of simplicity that our framework imposes. Furthermore, a closer inspection of the network of the TAN classifier reveals no consistent links between the gene variables, as we explain more in the next section.

5 DISCUSSION

An integral part of the RSN method is that it will output the set of the most predictive genes found with high confidence during the estimation procedure. These can then be further analysed to gain new insight into the underlying biology. This means that the method must strike a balance between discovering a reasonable number of novel genes and not overfitting the data by learning classifiers with too many features. In this section, we explore the quality of the selected features on the prostate and B-cell datasets.

Figure 4 plots the log-likelihood and MDL scores for individual links in the naïve Bayes network, measured on the full B-cell data.

*****FIGURE 4 ABOUT HERE ******

The genes are ordered based on their MDL score. The plot shows how, despite the small number of observations, the two scores manage to differentiate amongst genes. A few genes on the right hand side of the plot are found with a high score and as such they are identified as the best candidate for future investigation. This is promising as it implies that our results should only generate a small number of high confidence genes for classifying the data.

By repeating the network learning 10 times for each cross-validation experiment, we are able to get a confidence level on each gene. Figure 5 plots the proportion of times a gene from the B-cell and prostate cancer dataset, respectively, was selected out of the total runs (260 for the B-cell and 100 for the Prostate). For the B-cell dataset all the genes with a proportion greater than zero (in total 21) are plotted in the figure, sorted in descending order. For the Prostate dataset, we only show those with a proportion greater than 0.5, out of the 103 genes that were found with a non-zero proportion.

*****FIGURE 5 ABOUT HERE ******

As one can see, only a few genes were consistently picked out in a high proportion of classifiers, with a large fraction of genes never selected by the method (i.e. with zero frequency). This means that the method does manage to home in on a small number of interesting genes, that can then be further investigated from a biological perspective. Note that the different shape of the two histograms for the two datasets is mainly due to the fact that not all the genes with proportion greater than zero are plotted on the prostate dataset. Figure 6 compares the results of the full prostate dataset with the list of genes found by the 25% Prostate dataset. It plots the average percentage of overlap between the top 20 genes found on the full dataset and the genes found on subsamples of the full data. The average is computed over four 25% samples of the full dataset. For a fixed frequency f, the overlap is defined by $\frac{n_{25\%}(f)}{20}$, where $n_{25\%}(f)$ is

the number of the top 20 genes on the full dataset which are found amongst the genes on the 25% data that have frequency greater than f. The plot shows how the degree of overlap increases from about 20% as the frequency decreases. About 80% of the genes found on the small dataset with a non-zero frequency include the top 20 genes of the full dataset. Note that the total number of genes found with non-zero frequency was on average 47 on the four 25% subsamples. These results show also the importance of genes selected with a low frequency as potentially interesting for further biological investigation and strenghten the case for data perturbation as a mean for the identification of these features.

*****FIGURE 6 ABOUT HERE ******

The RSN method makes the assumptions that the genes are independent given the class by using a naïve Bayes classifier and that simulated annealing finds a good solution to the optimization problem. We have tested the effect of both assumptions on the feature selection method. As for the first assumption, we have compared the features selected by RSN with the features selected by the same method but based on a TAN classifier. This classifier allows dependencies between the genes as already explained in Section 4.3. The comparison showed that there is an 81% and 96% overlap between the genes in Figure 5 and the same number of top genes found by the TAN classifier on the B-cell and Prostate datasets, respectively. Furthermore, the links between the gene variables were found with a very low frequency, less than 0.02 on the B-cell dataset and one single link with frequency 0.01 on the Prostate dataset. An inspection of the correlation between gene pairs in the two datasets revealed a relatively low correlation. The absolute correlation between gene pairs is on average 0.23 and 0.2 on the B-cell and Prostate datasets with only 7% and 4% of genes with correlation greater than 0.5, respectively. Overall, these results suggest that relaxing the assumption of independence between the genes brings no significant

change to the identification of the predictive features, when the aim is to learn a simple classifier from a very small number of samples and when the correlation between the genes is relatively low.

As for the second assumption, we have measured the consistency of simulated annealing within the different cross-validation folds in order to measure how often the same set of genes was selected in the optimization search. Figure 7 plots the percentage of times a gene is selected within each separate CV fold.

*****FIGURE 7 ABOUT HERE ******

We measure this on the four genes with the highest frequency count and three genes with low frequency found on the B-cell dataset. The plots show a high consistency of simulated annealing, both for the top genes, where the consistency is across folds, and maybe more interestingly, this is also the case for the bottom genes, where each gene is selected only on a very small number of folds. We have also looked at how often the same set of genes was selected together by the simulated annealing algorithm across the different folds. As expected, the consistency was lower: 16% of all networks generated were found to contain all of the top four links in Figure 7, whilst 38% contained three of them, 35% contained just two and 11% contained only one.

The genes identified with high confidence by our method are expected to be the ones that separate the two classes best. To test for this, we have measured the separation of the class means in terms of the class dispersion, using a standard Fisher's method [32], and compared the score achieved by the features selected by our MDL-RSN method with the one obtained from a set of randomly drawn features. The latter was measured as an average across 100 random samples. Figure 8 plots this measure of separability on the two datasets, as genes with decreasing confidence level are considered. The plot shows how overall the selected genes differentiate the two classes better than random genes. As

expected, the separability measure between the two classes increases with the dimension of the problem given the limited data available.

*****FIGURE 8 ABOUT HERE ******

The genes identified by our method as positively or negatively associated with the presence of latent herpesvirus infection of B-cell tumours and of prostate cancer (x-axes of Figure 5) were further investigated for biological significance. On the B-cell dataset, the selected genes have not been systematically identified before, therefore the presence or absence of a gene in the classifier will require further experimental validation. Nevertheless, a number of the genes have been previously associated with viral infections (marked in Figure 5 with asterisks), suggesting that the classifier has indeed identified biologically relevant genes. Some of the genes encode proteins known to interact directly with virus proteins and genes. For example, BCL2/adenovirus E1B 19kDa interacting protein 2 (U15173), which was found in 64% of the classifiers, is a known pro-apoptotic protein that interacts with the adenovirus E1B protein [33], whereas the nuclease sensitive element binding protein 1 (M85234) is known to activate Simian virus 40 DNA replication [34] and JC virus replication [35]. Other genes have known roles in the interferon response to virus infection. For example, the proteasome subunit beta type 8 (Z14982) is part of the interferon inducible immunoproteosome [36] and the damage-specific DNA binding protein 1 (U32986) is a cellular protein essential for the targeted degradation of STAT1 by several paramyxoviruses and is also targeted by hepatitis B virus X protein (HBx) to interfere with its cell growth functions [37]. Finally several genes have known roles in the cell mediated immune response to pathogens. CD48 (BC016182) is a co-receptor for strong natural killer cell adhesion [38] and beta-2-microglobulin (AK026463) forms a complex with MHC Class I molecules for presentation of viral peptides to the immune system [39]. Together these data demonstrate that

the classifier has identified genes involved with unlinked, diverse stages of viral infection suggesting their identification by chance is unlikely. The roles of the remaining genes in infection are unknown.

On the prostate cancer dataset, the top scoring EST (expressed sequence tags), with GENEBANK ID AA055368 and consistently identified in 79% of the experiments, is from caveolin-1, a prominent prostate cancer marker [40]. Interestingly, the third EST, with GENEBANK ID AA487560 and a score of 0.69, comes from a different part of the same gene. It is positive that our method has managed to identify both parts of the same gene. Amongst the high scoring ESTs there is also the SLUG gene (N64741) with a known repressor role in breast cancer [41]. It would be interesting to investigate this further and see whether this gene is related also to prostate cancer in males. Furthermore TCF7L1 (AA180237) is a well characterised cancer inducing transcription factor [42]. Finally, there are at least two ESTs from uncharacterised genes that our analysis shows to have strong connections to prostate.

The computational runtime of the method depends on the number of folds for cross-validation as well as the number of repeats of the stochastic method. Approximately, the cross-validation part requires 5 minutes per fold. In our experiments, we had 26 folds on the B-cell dataset and 10 on the Prostate dataset. In addition, we have run simulated annealing 10 times on each fold. Therefore, the overall runtime was near to 20 hours for the B-cell dataset and 8 hours for the Prostate dataset. However, we have only repeated the simulated annealing 10 times for each fold to illustrate its consistency. Given the high consistency shown by the results, it is not necessary to perform this step for further analysis, though it might be worth having reassurance of its consistency. We would like to point out that if the type of analysis discussed in this paper can be used to home in on interesting genes early on in the experimental phase,

then a huge amount of time will be saved.

6 CONCLUSION

In this paper, we have proposed a method capable of identifying features that are highly predictive in classifying high-dimensional data when only very few samples are available. In particular, we have focussed on the recent application of microarray technology. Given the high costs of a single microarray experiment, gene expression data are characterised by many variables, involving up to thousands of genes, but only a handful of observations, the samples. As such, these data are prime examples of extremely small sample-size but high-dimensional data.

We have used two gene expression datasets and a simulated dataset to validate and assess the performance of our method. The method was developed following certain criteria that we believe should be considered when analysing extremely small and high-dimensional data. First of all, we have chosen a simple classification method, involving only a small number of parameters to estimate, with the aim of reducing the risk of overfitting the data. Secondly, we have tailored the feature selection method to penalise the selection of too many features, either by putting a constraint on the maximum number of features or by using a score that incorporates a penalization term for overly complex models. This second approach is particularly appealing as it avoids the choice of the maximum number of features allowed for selection, which is often a difficult and expensive task. Thirdly, we have used simulated annealing to select the optimal set of features in the dataset. We have generally found that a global optimization technique such as this performs better than a greedy search, as the latter can easily get stuck in local optima. Finally, we have repeated the learning many times as well as perturbing the data, in order to obtain a more

robust set of predictive genes.

The results show an improvement of our method as compared to other more standard techniques. When using this method, only few genes were consistently identified with high confidence. These were further validated from a biological perspective. A number of the prostate cancer and B-cell genes identified have been previously associated with prostate cancer and viral infections, respectively. Other less well-known genes from the B-cell dataset, found in a high proportion of the classifiers, are being investigated in a further study.

The small number of interesting genes identified by the method are the basis for the design of the next set of biological experiments. The identified features and the more substantial new data could then be used to build and test a final classifier. Future work will also involve incorporating existing biological knowledge into the prior of the Bayesian classifier as well as extending the model to handle temporal information.

ACKNOWLEDGEMENTS

The work of Veronica Vinciotti on this paper was funded by the BBSRC (Grant: 100/EGM17735). We would like to thank Eleftherios Panteris for his help with the biological validation of the prostate data. We would also like to thank the two anonymous reviewers for their constructive comments.

References

[1] Duggan JD, Bittner M, Chen Y, Meltzer P, Trent JM. Expression profiling using cDNA microarrays. Nature Genetics Supplement 1999;21:10–14.

[2] Ambroise C, McLachlan GJ. Selection bias in gene extraction on the basis of microarray gene-expression data. PNAS 2002;99:6562–6566.

[3] Dougherty ER. Small sample issues of microarray-based classification. Comparative and Functional Genomics 2001;2:28–34.

[4] Hand DJ, Yu K. Idiot's Bayes - not so stupid after all? International Statistical Review 2001;69:385–398.

[5] Ben-Dor A, Bruhn L, Friedman N, Nachman I, Schummer M, Yakhini Z. Tissue classification with gene expression profiles. Journal of Computational Biology 2000;7:559–584.

[6] Dudoit S, Fridlyand J, Speed TP. Comparison of discrimination methods for the classification of tumors using gene expression data. JASA 2002;97:77–87.

[7] Xiong M, Li W, Zhao J, Jin L, Boerwinkle E. Feature (gene) selection in gene expression-based tumor classification. Molecular Genetics and Metabolism 2001;73:239–247.

[8] Li J, Liu H, Downing JR, Yeoh AE, Wong L. Simple rules underlying gene expression profiles of more than six subtypes of acute lymphoblastic leukemia (ALL) patients. Bioinformatics 2003;19(1):71–78.

[9] Tobler JB, Molla MN, Nuwaysir EF, Green RD, Shavlik JW. Evaluating machine learning approaches for align probe selection for gene-expression arrays. Bioinformatics 2002;18(1):S164–S171.

[10] Blum A, Langley P. Selection of relevant features and examples in machine learning. Artificial Intelligence 1997;97(1-2):245–271.

[11] Furlanello C, Serafini M, Merler S, Jurman G. Entropy-based gene ranking without selection bias for the predictive classification of microarray data. BMC Bioinformatics 2003;4:54.

[12] Inza I, Sierra B, Blanco R. Gene selection by sequential search wrapper approaches in microarray cancer class prediction. Journal of Intelligence and Fuzzy Systems 2002;12:25–33.

[13] Wang J, Bo TB, Jonassen I, Myklebost O, Hovig E. Tumor classification and marker gene prediction by feature selection and fuzzy c-means clustering using microarray data. BMC Bioinformatics 2003;4:60.

[14] Xing EP, Karp M. CLIFF: clustering of high-dimensional microarray data via iterative feature filtering using normalized cuts. Bioinformatics 2001;17:S306–S315.

[15] Yeung KY, Bumgarner RE. Multiclass classification of microarray data with repeated measurements: application to cancer. Genome Biology 2003;4:R83.

[16] Inza I, Larrañaga P, Blanco R, Cerrolaza AJ. Filter versus wrapper gene selection approaches in DNA microarray domains. Artificial Intelligence in Medicine 2004;31:91–103.

[17] Liu H, Li J, Wong L. A comparative study of feature selection and classification methods using gene expression profiles and proteomic patterns. Genome Informatics 2002;13:51–60.

[18] Li T, Zhang C, Ogihara M. A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. Bioinformatics 2004;20(15):2429–2437.

[19] Lee KE, Sha N, Dougherty ER, Vannucci M, Mallick BK. Gene selection: a Bayesian variable selection approach. Bioinformatics 2003;19:90–97.

[20] Jörnsten R, Yu B. Simulataneous gene clustering and subset selection for sample classification via MDL. Bioinformatics 2003;19:1100–1109.

[21] Cooper GF, Herskovitz E. A Bayesian method for the induction of probabilistic networks from data. Machine Learning 1992;9:309–347.

[22] Lam W, Bacchus F. Learning Bayesian belief networks: an approach based on the MDL principle. Computational Intelligence 1994;10(4):269–293.

[23] Friedman N, Goldszmidt M. Learning Bayesian networks with local structure. In: Proceedings of the 12th conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann; 1996. p. 252–262.

[24] Friedman N, Yakhini Z. On the sample complexity of learning Bayesian networks. In: Proceedings of the 12th conference on Uncertainty in Artificial Intelligence. Morgan Kaufmann; 1996. p. 274–282.

[25] Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. Science 1983;220:671–680.

[26] Langley P, Sage S. Induction of selective Bayesian classifiers. In: Proceedings of the 10th Annual Conference on Uncertainty in AI. Seattle: Morgan Kaufmann; 1994. p. 399–406.

[27] Friedman N, Linial M, Nachman I, Pe'er D. Using Bayesian networks to analyze expression data. Journal of Computational Biology, 2000;7:601–620.

[28] Lapointe J, Li C, Higgins JP, van de Rijn M, Bair E, Montgomery K, et al. Gene expression profiling identifies clinically relevant subtypes of prostate cancer. PNAS 2004;101(3):811–816.

[29] Jenner RJ, Maillard K, Cattini N, Weiss RA, Boshoff C, Wooster R, et al. Kaposi's sarcoma-associated herpesvirus-infected primary effusion lymphoma has a plasma cell gene expression profile. PNAS 2003;100:10399–10404.

[30] Quinlan JR. C4.5 : Programs for machine learning. Morgan Kaufmann; 1993.

[31] Friedman N, Geiger D, Goldszmidt M. Bayesian network classifiers. Machine Learning 1997;29:131–163.

[32] Hand DJ. Construction and assessment of classification rules. Chichester: Wiley; 1997.

[33] Boyd J, Malstrom S, Subramanian T, Venkatesh L, Schaeper U, Elangovan B, et al. Adenovirus E1B 19kDa and Bcl-2 proteins interact with a common set of cellular proteins. Cell 1994;79(2):341–351.

[34] Muller K, Mermod N. The Histone-interacting domain of nuclear factor I activates Simian virus 40 DNA replication in vivo. Journal of Biological Chemistry 2000;275(3):1645–1650.

[35] Kim SY, Choi EC, Woo JY, Henson JW, Kim HS. Transcriptional activation of JC virus early promoter by phorbol ester and interleukin-1[beta]: critical role of nuclear factor-1. Virology 2004;327(1):60–69.

[36] Rock KL, Goldberg AL. Degradation of cell proteins and the generation of MHC class I-presented peptides. Annual Review of Immunology 1999 1999;17(1):739–779.

[37] Leupin O, Bontron S, Strubin M. Hepatitis B virus X protein and Simian virus 5 V protein exhibit similar UV-DDB1 binding properties to mediate distinct activities. Journal of Virology 2003;77(11):6274–6283.

[38] Barber DF, Long EO. Coexpression of CD58 or CD48 with intercellular adhesion molecule 1 on target cells enhances adhesion of resting NK cells. Journal of Immunology 2003;170(1):294–299.

[39] DM Hill TKasliwal, Schwarz E, Hebert AM, Chen T, Gubina E, Zhang L, et al. A dominant negative mutant beta 2-microglobulin blocks the extracellular folding of a major histocompatibility complex class I heavy chain. Journal of Biological Chemistry 2003;278(8):5630–5638.

[40] Yang G, Addai J, Ittmann M, Wheeler TM, Thompson TC. Elevated caveolin-1 levels in African-American versus White-American prostate cancer. Clinical Cancer Research 2000;6:3430–3433.

[41] Hajra KM, Chen DY, Fearon ER. The SLUG zinc-finger protein represses E-cadherin in breast cancer. Cancer Research 2002;62:1613–1618.

[42] Sagara N, Katoh M. Mitomycin C resistance induced by TCF-3 overexpression in gastric cancer cell line MKN28 is associated with DT-diaphorase down- regulation. Cancer Research 2000;6:3430–3433.

Tables

	B-CELL	PROSTATE	25% PROSTATE
MDL-RSN	73%	96%	88%
Lik-RSN(5)	68%	96%	91%
SW-RSN(5)	73%	95%	83%
SW-LinReg(5)	62%	93%	74%
C5	38%	84%	87%

Table 1: Comparison of the RSN classifier to other feature selection methods on the three datasets.

Figures

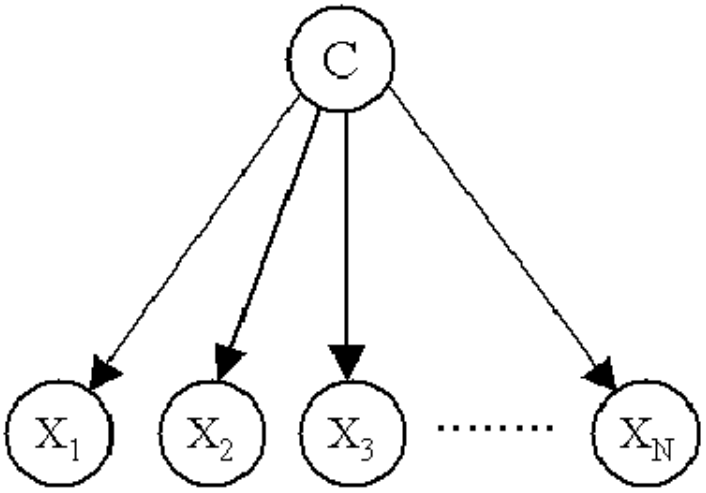

Figure 1: The naïve Bayes classifier.

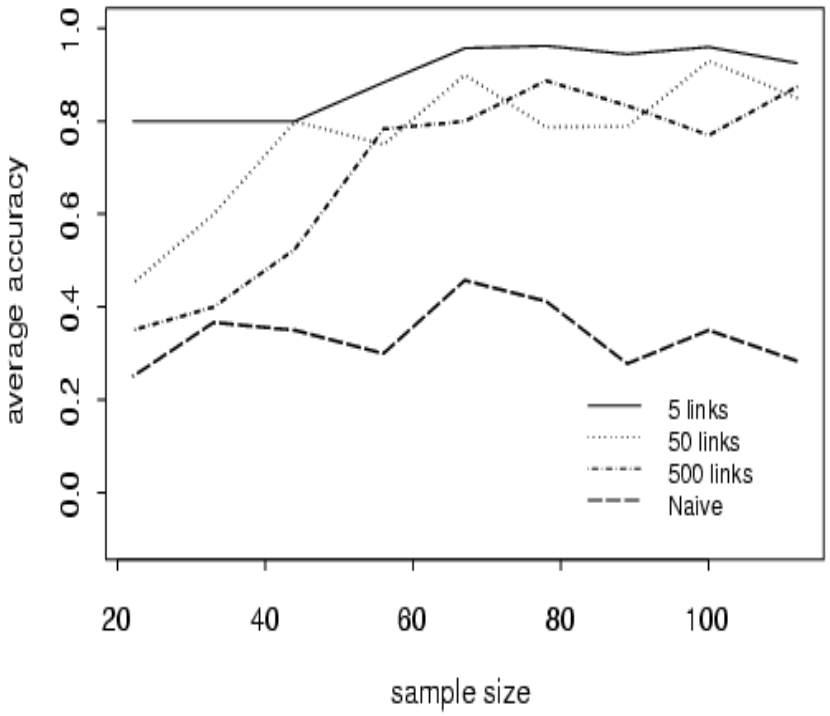

Figure 2: The classification accuracy of RSN models based on the likelihood score and increasing values of k, on datasets with sequentially reduced sample size.

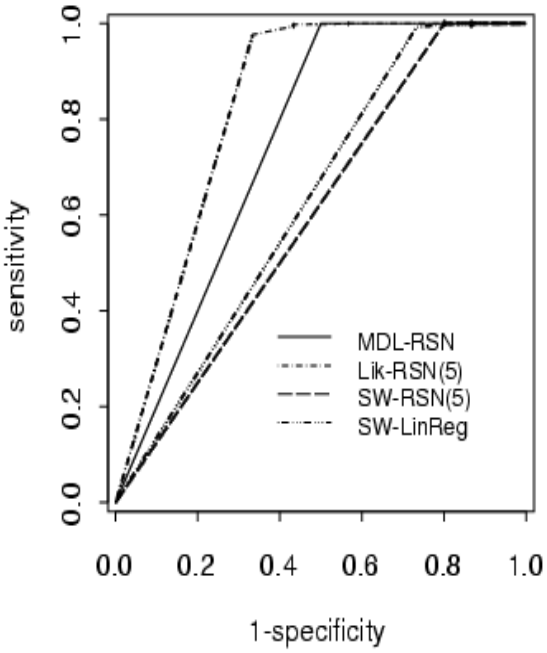

Figure 3: ROC curves on simulated data comparing different versions of the RSN method and the standard stepwise linear regression.

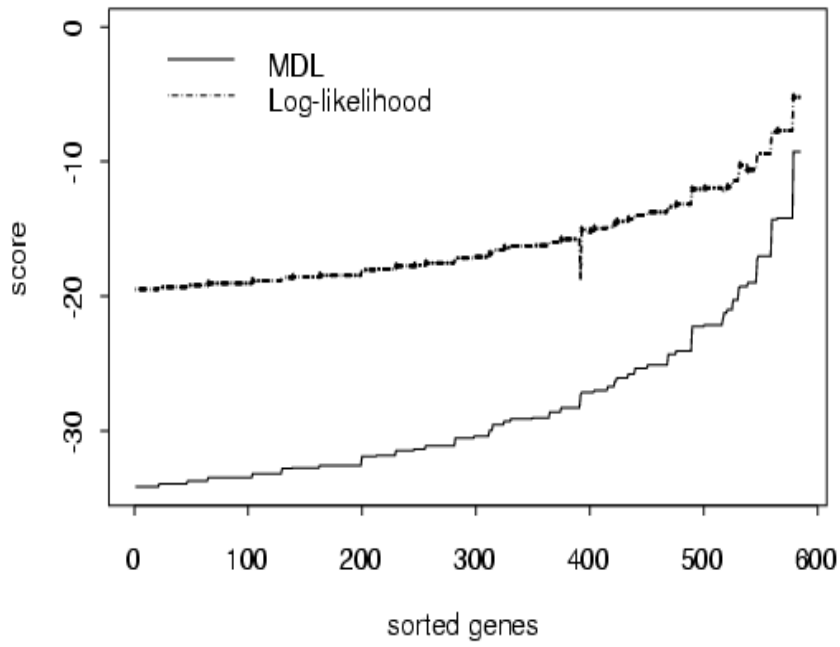

Figure 4: The log-likelihood and MDL scores for individual links in the B-cell dataset.

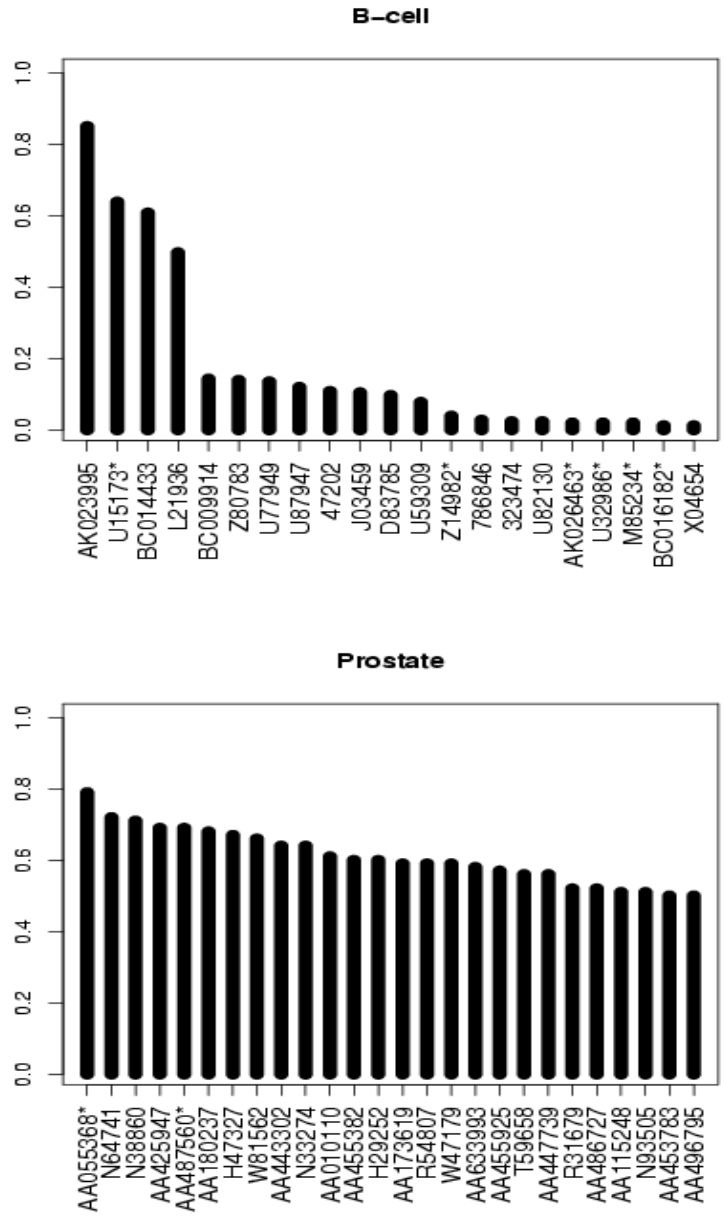

Figure 5: Frequency of links discovered by MDL-RSN on the B-cell and Prostate datasets. All links with frequency greater than zero are shown for the B-cell dataset, whereas only the links with frequency greater than 0.5 are shown for the Prostate dataset. The asterisks refer to those genes that were previously associated with viral infection or prostate cancer.

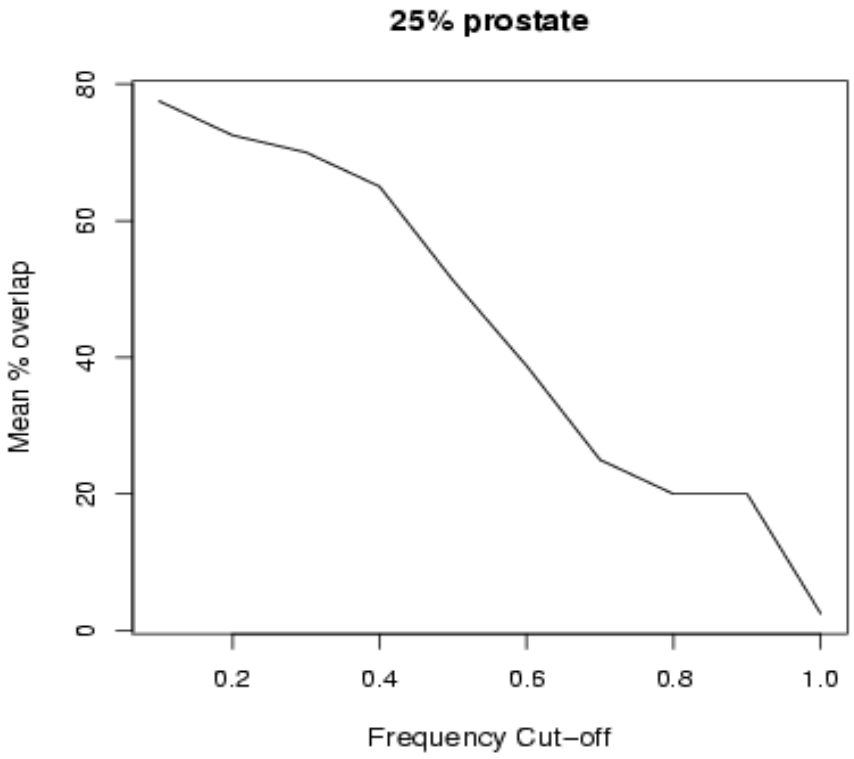

Figure 6: Average percentage of overlap between the top 20 genes found by MDL-RSN on the full prostate dataset and the genes found by the same method on four 25% subsamples of the data with frequency greater than a specified cut-off.

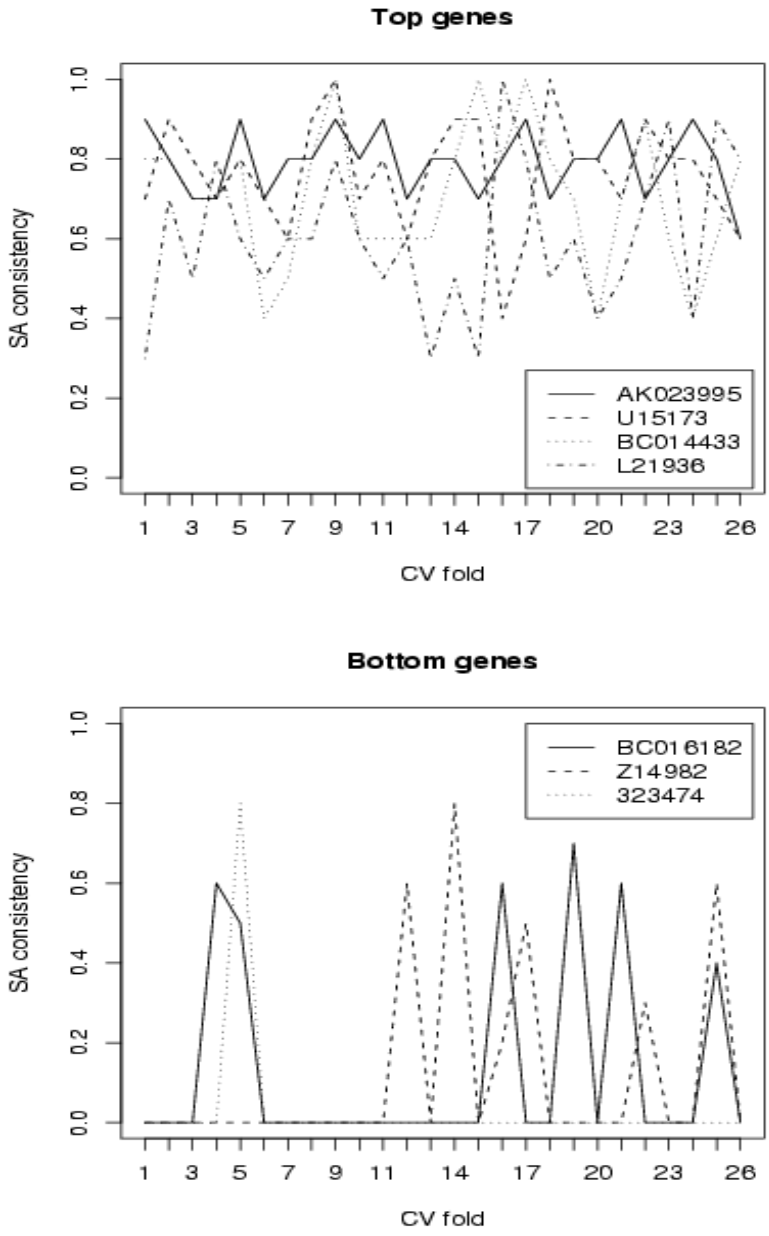

Figure 7: Consistency of simulated annealing (SA) within the different CV-folds for the four top genes (left) and the three bottom genes (right) selected by MDL-RSN on the B-cell dataset.

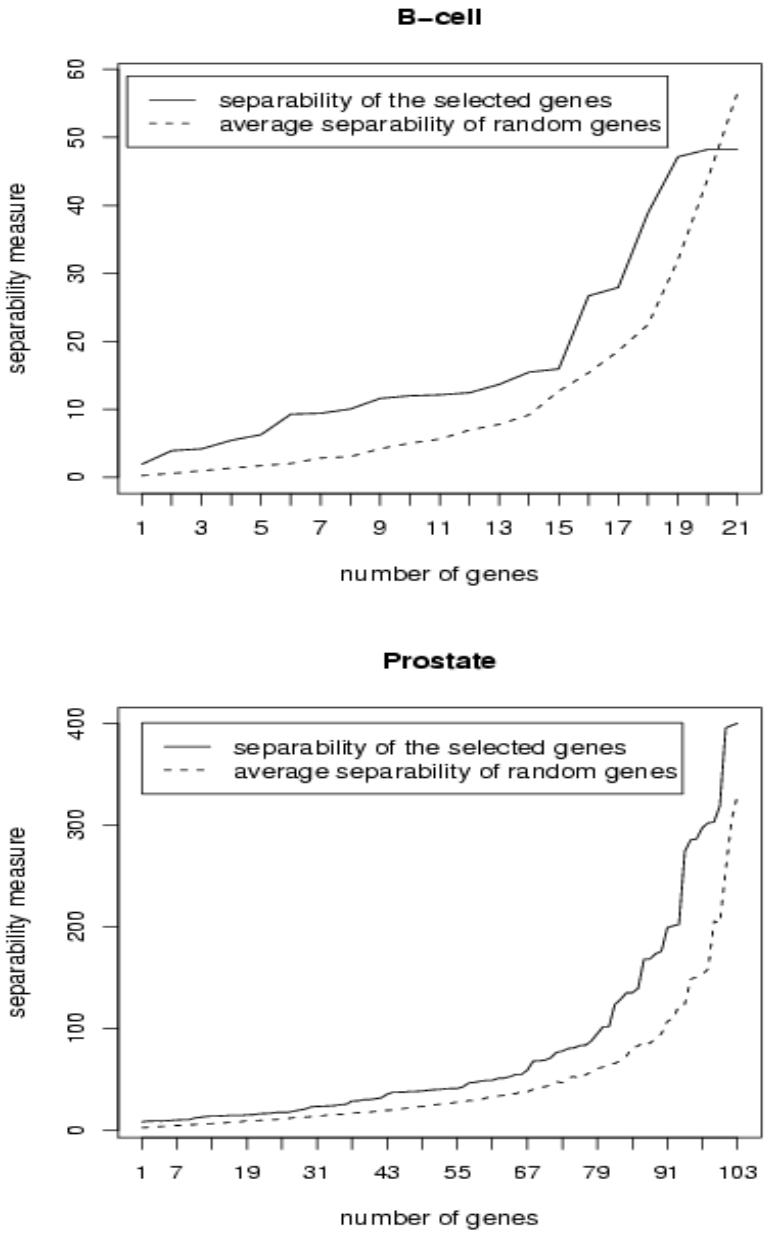

Figure 8: Measure of separation of the class means in terms of the class dispersion for the features selected by MDL-RSN, in decreasing value of confidence (solid line), as compared with a same number of features randomly drawn (dotted line).