Selection of Discriminative Genes in Microarray Experiments Using Mathematical Programming

Regina Berretta, Alexandre Mendes and Pablo Moscato

Centre for Bioinformatics, Biomarker Discovery & Information-Based Medicine
School of Electrical Engineering and Computer Science
Faculty of Engineering and Built Environment
University of Newcastle
Callaghan, NSW, 2308, Australia
E-mail: {Regina.Berretta,Alexandre.Mendes,Pablo.Moscato}@newcastle.edu.au

Microarray technologies allow the measurement of thousands of gene expression levels simultaneously. With them biologists can have a powerful new tool to analyse the complex dynamical process of living organisms. These technologies are challenging traditional scientific disciplines, including Computer Science and Statistics. The reason of this challenge is based on the novel type of large-scale data mining applications. A typical microarray experiment is very costly and, due to budget limitations, the ratio of experiments to genes is generally on the order of 1/100. As a consequence, we rely on combinatorial optimization formalisms to develop robust feature selection methods. In this paper we demonstrate their usefulness in selecting genes that allow a molecular classification of cancer samples when we are given as labels their assumed origin (Colon, Melanoma, etc.). We present some results on five types of cancer presented on a public domain dataset which will allow for the reproducibility of our results.

ACM Classification: G.1.6 (Optimization - Integer programming); H.2.8 (Database Applications - Data mining); I.5. (Pattern Recognition - Feature evaluation and selection); J.3 (Life and Medical Sciences - Biology and genetics)

1. INTRODUCTION

This paper focuses on the extraction of relevant information from a set of microarray experiments by selecting specific genes with the particular purpose to help understand the genetic mechanisms behind five common types of cancer. The models and algorithms presented may have important uses in other applications in the wider field of the Feature Selection problem (Dash and Liu, 1997, 2003; Xing, 2003; Akutsu and Miyano, 2002; Moscato *et al*, 2005c).

Microarray technologies were introduced in the past decade and their use is now permeating scientific disciplines boundaries. More recent advances, in the beginning on this decade, allowed researchers to perform *whole-genome experiments*, in which they simultaneously measure the expression level of thousands of genes allowing their collective changes under controlled conditions to be measured. An unprecedented number of possibilities arise, but this technology is also challenging the area of knowledge extraction from databases (Tamayo and Ramaswamy, 2003; Quackenbush, 2001). Analysis of these data can be done using unsupervised learning techniques as

Manuscript received: 27 February 2006
Communicating Editor: John Cooper

clustering/ordering algorithms (Moscato *et al*, 2005a; Eisen *et al*, 1998) and supervised techniques. In the latter case, we know some additional information that defines a class of experiments as a separate group (Brown *et al*, 2000; Furey *et al*, 2000).

Several combinatorial optimization problems associated to the interpretation of this wealth of information are NP-hard, thus it is unlikely that polynomial-time algorithms exist for them. As a consequence, one of the practical difficulties faced is the amount of data coming from microarray experiments can be very large. In addition, it is generally the case that the number of genes in a microarray dataset is two orders of magnitude larger than the samples. In such scenario, it becomes easy to find low cardinality explanations for the process of interest but many of these genes may be totally unrelated to the research question. These "false positives" can just be explained due to the low samples-to-features ratio. This means that mathematical models for clustering and/or classification methods should take this into account.

In this paper, we used variations of the (α,β)-*k-Feature Set Problem* introduced by Cotta *et al* (2004), to perform the selection of genes. We present two integer programming models for these variants. These models have been also applied to an Alzheimer dataset (Moscato *et al*, 2005b) and different types of cancer (Berretta *et al*, 2005). Our group has also presented a pedagogical and illustrative example of their application to political science (Moscato *et al*, 2005c).

We have used CPLEX (a mathematical programming software package) to guarantee the optimality of our solutions. Our results show that our method has been able to select relevant groups of discriminatory genes, each one containing between 6% and 17% of the total number of genes for which a measure has been provided. The final number depends on the type of cancer. In addition, the solutions present in this paper guarantee a *maximal* number of same-class similarities as well as an *optimal (maximum)* dissimilarity for samples in different classes (to be explained in Section 4).

In this paper, we also describe the use of reductions rules (also known as pre-processing rules) that can considerably reduce the size of the instances. The application of those pre-processing rules creates an instance which in many circumstances turns to be much smaller than the original instance. These reductions are safe since at least one optimal solution exists in the reduced instance.

We applied our approach using a dataset known as NCI60, which is available online and helps for comparison purposes (Ross *et al*, 2000). This dataset contains the gene expression profile of almost seven thousands genes on 60 different types of cancer cell lines and helps to illustrate the advantage of using our mathematical programming approach.

The same instance (NCI60) was used in Berretta *et al* (2005). However, in this contribution the discretization of the expression values was done in a different way. In this paper we use an entropy based method proposed by Fayyad and Irani (1993), which makes an independent analysis for each gene. We note that Cotta *et al* (2004) proved that an associated thresholding problem is NP-hard and in the same paper addressed it with an Evolutionary Algorithm. We have found that the application of Fayyad and Irani's method also provides good results and is a useful alternative. We report our results using it in the identification of differentially expressed genes in Leukaemia, Melanoma, as well as CNS, Colon and Renal cancer.

2. MATHEMATICAL MODELS

2.1. Min FEATURE SET Problem

The Min FEATURE SET problem we consider in this paper is the following. Consider an integer matrix $G = g_{ij}$, $1 \leq i \leq e$, $1 \leq j \leq n$, where e is the number of experiments/samples, n is the number of features (genes) and g_{ij} represents the measured level of activity of gene j in experiment i. Consider also a vector $T = t_i$, where $1 \leq i \leq e$ and t_i represents the class label that corresponds to experiment i. The objective is to find the minimum cardinality set of features (genes in our application), denoted as S, such that for all pairs of experiments that belong to different classes, there exists at least one feature (gene) that belongs to S, and such that the feature is in different states. In other words, for all pairs of experiments (p,q), with $t_p \neq t_q$, $\exists j \in S$, such that $g_{pj} \neq g_{qj}$. It is then clear that we need that each feature can only be in a few repertoire of possible states, otherwise any feature can easily become discriminative by the definition above (as it is in the case of multi-digit integers accepted as input, which would render the model not very useful in practice).

We will first discuss the Boolean case, assume that the instance of our interest is the following Boolean matrix G and the Boolean vector T below. In this case, the minimum feature set for this instance is $S = \{F4, F5\}$.

G_{5x5}

	F1	*F2*	*F3*	*F4*	*F5*
E1	*1*	*0*	*0*	*0*	*1*
E2	*0*	*1*	*1*	*0*	*1*
E3	*1*	*0*	*0*	*0*	*0*
E4	*1*	*1*	*1*	*1*	*1*
E5	*0*	*1*	*0*	*1*	*1*

T_5

Class
0
0
1
1
1

The *k-FEATURE SET* problem is NP-complete (Davies and Russel, 1994). In addition, Cotta and Moscato (2003) showed that the parameterized version of the *k-FEATURE SET* problem (with parameter k) is W[2]-Complete. In the optimization version, we are interested in finding a feature set of minimum cardinality.

With the purpose of writing an integer programming model, first we define a matrix $A = a_{ij}$, $1 \leq i \leq m$, $1 \leq j \leq n$, where m is the number of pairs of examples that belong to different classes, n is the number of features, and a_{ij} is 1 if $g_{pj} \neq g_{qj}$ or 0 if $g_{pj} = g_{qj}$, where $t_p \neq t_q$. In other words, a_{ij} represents whether the features types in the pair of examples that belong to different classes (p,q) are different or not. Using the previous illustrative instance of the problem, the matrix A would be:

	F1	*F2*	*F3*	*F4*	*F5*
E1,E3	*0*	*0*	*0*	*0*	*1*
E1,E4	*0*	*1*	*1*	*1*	*0*
E1,E5	*1*	*1*	*0*	*1*	*0*
E2,E3	*1*	*1*	*1*	*0*	*1*
E2,E4	*1*	*0*	*0*	*1*	*0*
E2,E5	*0*	*0*	*1*	*1*	*0*

The objective is to choose a minimum subset S of features (columns) which have at least one '*1*' value in each line. That is, a minimum set of features corresponds to the minimum subset of columns having ones that cover all pair of examples. Notice that the minimum feature set in the example is $S = \{F4,F5\}$. An integer programming model for the Min FEATURE SET can be as shown below, where the variable $x_j = 1$ if the feature j is chosen; and 0, otherwise.

$$Min \sum_{j=1}^{n} x_j \tag{1}$$

$$\sum_{j=1}^{n} A_{ij} x_j \geq 0 \quad i=1,\ldots, m \tag{2}$$

$$x_j = 0 \text{ or } 1.$$

Note that the model (1-2) represents also the Min SET COVERING, a classical problem in combinatorial optimization for which many techniques have been developed (Caprara, Toth and Fischetti, 2000).

2.2. Reductions for Min FEATURE SET Problem

Reductions for Min FEATURE SET are rules that can be applied to an instance to try to eliminate, *a priori*, some rows and columns from matrix *A* and, consequently, reduce the instance size. We describe four reduction rules for the Min FEATURE SET problem below. These reductions rules are the same for the Set Covering Problem and it is possible to find them in references about classical Integer Programming texts such as Garfinkel and Nemhauser (1972).

2.2.1. Reduction R0

If $a_{ij} = 0$ for all j, then, the instance is infeasible, since the constraint (2) cannot be satisfied. In other words, if no feature can distinguish a pair of examples that belong to different classes, then the instance is infeasible.

2.2.2. Reduction R1

If $a_{ij} = 0$ for all $j \neq k$ and $a_{ik} = 1$, then $x_k = 1$. In other words, if just one feature distinguishes a pair of examples that belong to different classes, then this feature must be in any feasible cardinality solution. In addition, all rows i such that $a_{ik} = 1$, can be deleted, since the feature k will cover these rows. Finally, column k can be deleted.

In the example given, the feature *F5* should be in the solution, since it is the only one that covers the pair of examples *(E1,E3)*. We can delete rows *1* and *4*, since the pair of examples *(E1,E3)* and *(E2,E3)* are covered by the inclusion of feature *F5* in our solution.

2.2.3. Reduction R2

A feature j covers a subset W if $a_{ij} = 1$ for all $i \in W$. If a feature j_1 covers a subset W_1 and j_2 covers a subset W_2 and $W_2 \subseteq W_1$, then feature j_2 is dominated by feature j_1 and consequently, can be deleted, i.e., $x_{j2} = 0$.

In the example above, after being updated with the result of reduction R1, the feature *F4* covers the set $W_4 = \{(E1,E4), (E1,E5), (E2,E4), (E2,E5)\}$. The feature *F3* covers the set $W_3 = \{(E1,E4), (E2,E5)\}$. Since, $W_3 \subseteq W_4$, *F3* is redundant and can be deleted. Notice that, with the same rule we can delete *F1* and *F2*. Now, using the reduction rule R1, feature *F4* is chosen and the instance is solved to optimality (as the reduction rules are safe procedures that do not miss at least one optimal solution of the original instance after they reduce it).

2.2.4. Reduction R3

Let $Q_1 = \{ j \,/\, a_{i_1 j} = 1 \}$ and $Q_2 = \{ j \,/\, a_{i_2 j} = 1\}$. If $Q_1 \subseteq Q_2$ then row i_2 can be deleted. In other words, if a pair of examples i_1 is covered by the set of features Q_1 and a pair of examples i_2 is covered by the set of features Q_2 and $Q_1 \subseteq Q_2$, we can delete the pair i_2, since it will be covered by any of the features chosen to cover the pair i_1.

In the example, the pair of examples *(E1,E3)* is covered by $Q_1 = \{F5\}$ and the pair of examples *(E2,E3)* is covered by $Q_2 = \{F1,F2,F3,F5\}$. Since $Q_1 \subseteq Q_2$, the pair of examples *(E2,E3)* can be deleted from matrix *A*. Notice that when we choose a feature to cover the pair *(E1,E3)* we inevitably will cover the pair *(E2,E3)*.

Although the Min FEATURE SET is an NP-hard optimization problem, the reduction rules can be very useful in practice to reduce the instance size before we apply a method (either a polynomial-time heuristic or an exact exponential time algorithm) to find one of the optimal solutions.

2.3. Min α-β FEATURE SET Problem

A generalization of the Min FEATURE SET is the Min α–β FEATURE SET problem introduced by Cotta *et al* (2004). This generalization could be very useful when the dataset is noisy and a larger number of different discriminative features need to be considered for a more reliable classification.

The problem is defined as follows. We have the same input as for Min Feature Set, i.e., matrix $G = g_{ij}$, $1 \le i \le e$, $1 \le j \le n$, where e is the number of samples (experiments) and n is the number of features (genes) and a vector $T = t_i$, where $1 \le i \le e$ and t_i represents the class (outcome, type of cancer) of the sample (experiment) i. In addition, the input also includes two integer values $\alpha \ge 1$ and $\boldsymbol{\beta} \ge 0$. The objective is again to find the minimum set of genes (features) S, but the two conditions below also need to be satisfied.

Condition 1: For all pairs of samples that belong to different classes, at least α features that belong to S have different feature types. In other words,

For all pairs (p,q) with $t_p \ne t_q$,

define $S_1(p,q) = \{j \in S \mid g_{pj} \ne g_{qj}\}$

So, $|S_1| \ge \alpha$.

Condition 2: For all pairs of samples that belong to the same class, at least β features that belong to S have identical feature types. In other words,

For all pairs (p,q) with $t_p = t_q$,

define $S_2(p,q) = \{j \in S / g_{pj} = g_{qj}\}$

So, $|S_2| \ge \beta$.

To illustrate, consider the same matrix *G* defined previously. Observe that, if we have as input the values $\alpha=1$ and $\beta=1$, the Min α–β Feature Set cannot be $S = \{F4,F5\}$, since the examples *E3* and *E4*, which belong to the same class are completely different for the features *F4* and *F5*. For $\alpha=1$ and $\beta=1$ the minimum cardinality ($\alpha=1$, $\beta=1$)-feature set is $S = \{F1,F3,F5\}$.

For an integer programming formulation for this problem, we will define two matrices, *A* and *B*. Matrix *A* will be the same defined before, that is, $A = a_{ij}$, $1 \le i \le m$, $1 \le j \le n$, where n is the number of features, m is the number of pairs of examples that belong to different classes and a_{ij} is 1 if $g_{pj} \ne g_{qj}$ or 0 otherwise, where $t_p \ne t_q$. Matrix *B* will be $B = b_{ij}$, $1 \le i \le m'$, $1 \le j \le n$, where n is the number of features, m' is the number of pairs of examples that belong to the same classes and b_{ij} is 1 if $g_{pj} = g_{qj}$ or 0 otherwise, where $t_p = t_q$. Using the previous example, the matrix *B* would be:

	F1	F2	F3	F4	F5
E1,E2	0	0	0	1	1
E3,E4	1	0	0	0	0
E3,E5	0	0	1	0	0
E4,E5	0	1	0	1	1

The mathematical model can be written as:

$$\text{Min} \sum_{j=1}^{n} x_j \tag{3}$$

$$\sum_{j=1}^{n} A_{ij} x_j \geq \alpha \quad i=1,\ldots, m \tag{4}$$

$$\sum_{j=1}^{n} B_{ij} x_j \geq \beta \quad i=1,\ldots, m' \tag{5}$$

$$x_j = 0 \text{ or } 1$$

2.4. Reductions for Min α–β FEATURE SET

We define below reduction rules for Min α–β FEATURE as described before for Min FEATURE SET Problem. Consider the following definitions:

$$Q_a^i = \{ j / a_{ij} = 1\} \text{ and } Q_b^l = \{ j / b_{lj} = 1\}.$$

The sets Q_a^i and Q_b^l represent the features that can cover a pair of samples i and l, respectively, from matrix A and B. Let r_a^i be an integer that represents the number of features that remain to cover the pair the samples i from matrix A by α. Equivalently, r_b^l represents the number of features that remain to cover the pair the samples l from matrix B by β. At the beginning of the application of the reduction rules $r_a^i = \alpha$ and $r_b^l = \beta$.

2.4.1. Reduction R0

If $|Q_a^i| < r_a^i$, for at least one row i from matrix A, then the instance is infeasible, since the constraint (4) cannot be satisfied. Analogously, if $|Q_b^l| < r_b^l$, for at least one row l from matrix B, the instance is infeasible, since constraint (5) cannot be satisfied. In other words, if at least there is one pair of examples that belong to different classes that does not have at least r_a^i features that have different values for them, then the instance is infeasible (analogously for the within class similarity constraint).

2.4.2. Reduction R1

If $|Q_a^i| = r_a^i$ for any pair of examples i, then $x_j = 1$ for all $j \in Q_a^i$. In other words, if a pair of examples i is covered by exactly r_a^i features, then all these features should be in any optimal solution. Next, for all $j \in Q_a^i$, it is necessary to update all $r_a^i / a_{ij}=1$ and $r_b^l / b_{ij}=1$. Finally, we can delete all rows i from A such that $r_a^i \leq 0$, all rows l from B such that $r_b^l \leq 0$ and all columns $j \in Q_a^i$.

Analogously, if $|Q_b^l| = r_b^l$, for any l, then $x_j = 1$ for all $j \in Q_b^l$. In other words, if a pair of examples is covered by exactly r_b^l features, then all these features should be in the solution. Next, for all $j \in Q_b^l$, it is necessary to update all $r_a^i\, / a_{ij}=1$ and $r_b^l\, / b_{ij}=1$. Finally, again, we can delete all rows i from A such that $r_a^i \leq 0$, all rows l from B such that $r_b^l \leq 0$ and all columns $j \in Q_b^l$.

Consider $\alpha=\beta=1$ in the example. The feature *F5* should be in the solution, since it is the only one that covers the pair of examples *(E1,E3)* when we examine the matrix *A*. Also we can delete rows *1* and *4* from matrix *A*, since the pair of examples *(E1,E3)* and *(E2,E3)* are covered by feature *F5* with $\alpha=1$.

In the matrix *B* we can delete the rows *1* and *4*, since the feature *F5* will cover the pair of examples *(E1,E2)* and *(E4,E5)*. We conclude that features *F1* and *F3* should be in the solution, since only *F1* covers the pair of examples *(E3,E4)* and only *F3* covers the pair of examples *(E3,E5)*. We also can delete the rows *2*, *3* and *6* from matrix *A*, since features *F1* and *F3* cover all pair of examples that remain in the matrix A. Notice that we could reduce the entire instance and finish with the solution *{F1,F3,F5}*.

2.4.3. Reduction R2

A feature j covers a subset W_α if $a_{ij} = 1$ for all $i \in W_\alpha$. Respectively, a feature j covers a subset W_β if $b_{ij} = 1$ for all $i \in W_\beta$. If a feature j_1 covers a subset W_α^1 and W_β^1; j_2 covers a subset W_α^2 and W_β^2 ; $W_\alpha^2 \subseteq W_\alpha^1$ and $W_\beta^2 \subseteq W_\beta^1$; and for all $i \in W_\alpha^2$ we have $| Q_a^i | > r_a^i$ and for all $i \in W_\beta^2$ we have $| Q_b^i | > r_b^l$; then j_2 is redundant and can be deleted.

2.4.4. Reduction R3

If $Q_a^{i_1} \subseteq Q_a^{i_2}$ and $r_a^{i_1} \geq r_a^{i_2}$ or $Q_b^{l_1} \subseteq Q_a^{i_2}$ and $r_b^{l_1} \geq r_a^{i_2}$, then row i_2 from matrix A can be deleted. In other words, if a pair of examples i_1 is covered by the set of features $Q_a^{i_1}$; a pair of examples i_2 is covered by the set of features $Q_a^{i_2}$ and $Q_a^{i_1} \subseteq Q_a^{i_2}$; then the pair of samples i_2 can be deleted, if the number of features that remain to cover the pair the samples i_1 is greater than the number of features that remain to cover the pair the samples i_2 ($r_a^{i_1} \geq r_a^{i_2}$). Equivalent interpretation can be done if $Q_b^{l_1} \subseteq Q_a^{i_2}$ and $r_b^{l_1} \geq r_a^{i_2}$. Analogously, if $Q_b^{l_1} \subseteq Q_b^{l_2}$ and $r_b^{l_1} \geq r_b^{l_2}$ or $Q_a^{i_1} \subseteq Q_b^{l_2}$ and $r_a^{i_1} \geq r_b^{l_2}$ then row l_2 from matrix B can be deleted.

2.5. Max Cover α–β k-FEATURE SET

We will introduce another mathematical model by fixing the number of features in a value k and the objective is to find a set of k features that maximize the *coverage*. The coverage represents the number of pair of examples that belong to different classes (matrix A) plus the number of pair of examples that belong to the same class (matrix B) that the set of features cover, including repetitions. The coverage of a feature j is:

$$c_j = \sum_{i=1}^{m} A_{ij} + \sum_{i=1}^{m'} B_{ij}$$

The mathematical model is described below.

$$\text{Max } \sum_{j=1}^{n} c_j x_j \tag{6}$$

$$\sum_{j=1}^{n} A_{ij} x_j \geq \alpha \quad i=1,\ldots,\ m \tag{7}$$

$$\sum_{j=1}^{n} B_{ij} x_j \geq \beta \quad i=1,\ldots, m' \tag{8}$$

$$\sum_{j=1}^{n} x_j = k \tag{9}$$

$$x_i = 0 \text{ or } 1$$

This model can be useful, for example, when an instance has more than one optimal solution for the model (3-5). In this case we may be interested to select, among all such solutions, the one that covers the largest number of pairs of examples.

3. COMPUTATIONAL EXPERIMENTS AND RESULTS

We now discuss the application of the models presented before in a well known instance described in Section 3.1. Sections 3.2 and 3.3 describe the experiments and results obtained, respectively.

3.1. The instance NCI60

Ross *et al* (2000) introduced an important dataset for the molecular classification of different types of cancer (available on the authors' website supplement) containing the expression level in 64 cell lines of 6,831 of 60 different types of cancer. Using a hierarchical clustering algorithm, Ross *et al* (2000) have identified several groups of genes that correspond to the labels of the cell lines. Two groups, namely *"Leukaemia Cluster"* and *"Melanoma Cluster"*, for example, have been visually identified as a highly-expressed group of genes in the leukaemia-derived and in most of the melanoma-derived cell lines. However, it is very difficult to identify from their results, an analogous group of genes that is highly under-expressed within the same type of cancer samples and also discriminates from the other types of cell lines.

Waddell and Kishino (2000) argued that such a dataset, even if excellent in technical terms, may be of low information content. They noted that Ross *et al* (2000) did not emphasise on the impact of mutation on cell lines upon their analysis; it is then possible that the expression profiles, conditioned to the mutation status of a group of *"key player"* genes, would be an important missing part of the study. It is then necessary to identify which are the subsets of genes that can "explain" commonalities and differences between any given group and the rest. In addition, there are two other reasons motivating our experiments: first, to allow a direct comparison with the results of Ross *et al* (2000) (which were biased to obtain over-expressed genes), and to provide genetic signatures for Colon, Renal and Central Nervous Systems tumours.

3.2. Computational Experiments

First, we completed all missing values for the NCI60 dataset using the *LSImpute_EMarray* algorithm introduced by Bø, Dysvik and Jonassen (2004). Our choice was based on its relatively low running time and good performance on the NCI60 dataset as independently verified by the original authors. For the estimation of the missing values we have used the set of 64 cell lines and 6,831 genes. After the missing values have been completed we worked with a reduced instance containing just the 34 samples from the five main types of cancer, namely CNS (5 samples), Renal (7 samples), Leukaemia (8 samples), Colon (7 samples) and Melanoma (7 samples), totalling 34 cell lines. The Leukaemia, Renal and Colon cell lines chosen are the same evidenced in Ross *et al*

(2000). For the Melanoma group, they included two breast cancer cell lines (MDA-MB435 and MDA-N). We removed them from this group for our study, and kept the original seven melanoma cell lines. Finally, we also selected five CNS cell lines. In Ross *et al* (2000) the authors did not consider the CNS cell lines as a separate group due to their results using an unsupervised hierarchical clustering heuristic method. The five groups and their associated cell lines are described in Figure 1 (*top-left*). This figure is also available in colour, along with other supplementary material at *http://www.cs.newcastle.edu.au/~nbi/JRPIT/JRPIT_nci60sup.html*.

Next, the gene expression values, which extend over a large interval (between -5.6 and 8.0) need to be quantized in a few states, to use the mathematical model described in Section 2.3. In Berretta *et al* (2005), we use information on the *average* and the *standard deviation* of the gene expression values over all genes and samples to create the discretization. As gene expression values can vary over different intervals for each gene, genes for which the expression values lie within a sufficiently small interval may be considered not relevant as discriminators, whereas they would if the discretization thresholds were gene-specific.

We applied an entropy-based heuristic introduced by Fayyad and Irani (1993) to create binary states for each gene (indicating over or under expression over a given value). The procedure used to discretize the gene expression values is described next.

In our application, assigning a threshold T for the gene expression values of gene g, induces a partition of the set of samples S in two subsets S_1 and S_2. If we have been also given as input for each sample one of two possible class labels, either C_1 or C_2, we can compute $P(C_i,S_j)$ the proportion of samples from class C_i in S_j. The *class information entropy* of the partition induced by T for gene g, denoted $E(g,T;,S)$, is given by:

$$E(g,T;S) = \frac{|S_1|}{|S|} \cdot Ent(S_1) + \frac{|S_2|}{|S|} \cdot Ent(S_2) \tag{10}$$

where $Ent(S_j)$ is the *class entropy* of a subset S_j defined as:

$$Ent(S_j) = -\sum_{i=1}^{2} P(C_i,S_j).log(P(C_i,S_j)) \qquad j = 1, 2. \tag{11}$$

A binary discretization for g is determined by selecting the cut point T for which $E(g,T;S)$ is minimal amongst all the possible cut points. The resulting *information gain* is given by:

$$Gain(g,T;S) = Ent(S) - E(g,T;S) \tag{12}$$

The *Minimal Description Length Principle* is used to check whether the information gain suffices to characterize the gene g as being discriminative for the two classes of cell lines. Thus, the gene is considered to be discriminative if:

$$Gain(g,T;S) > \frac{log_2(N-1)}{N} + \frac{\delta(g,T;S)}{N} \tag{13}$$

where N is the number of expression values in the set S (34 in our tests), and:

$$\delta(g,T;S) = log_2(3^c - 2) - [c.Ent(S) - c_1.Ent(S_1) - c_2.Ent(S_2)]\ . \tag{14}$$

where c is the number of class labels, and c_i is the number of class labels represented in the set S_i. In this paper, *we will only provide genetic signatures with genes for which the algorithm above has* *assigned a unique threshold*. We have chosen to do this for illustrative purposes only, as in these cases an individual gene will tend to be in two clearly identifiable different states.

To find a genetic signature associated to one of the groups, for instance the Melanoma cell lines, our two classes will be '*Melanoma*' and '*all-others*', where 'all others' in this case correspond to the other four remaining groups (CNS, Renal, Leukaemia and Colon) bundled as a single group ('*non-Melanoma*'). By repeating the same procedure with the five groups, we could uncover sets of genes that are differentially expressed in each of the groups, in comparison with the others. Using the entropy based method described above, we first remove the genes which are *a priori* not discriminative. In this paper, from the 6,831 genes in the dataset, those that did not receive a unique threshold were discarded from the instance. The genes that have not been filtered out, now discretized in only two states, constitute an instance of the α–β FEATURE SET problem (where the maximum attainable value of α is different in each case). The number of genes that remain after the entropy-based filtering took place were 460 for CNS, 842 for Renal, 1,614 for Leukaemia, 735 for Colon and 990 for Melanoma. Any gene that has an expression level above that threshold value will be considered as "*over-expressed*" (for that sample, analogously we define "*under-expressed*"). We then calculate, for each of the instances, the maximum value of α that could be obtained by a feasible α–β feature set. These values were *283* for CNS, *431* for Renal, *767* for Leukaemia, *338* for Colon, and *605* for Melanoma. This means, for example, that we know that for every pair of cell lines, with one belonging to any of the seven Renal cell lines and the other belonging to anyone of the other "non-Renal'' group, we have at least *431* genes for which in one sample the gene is "*over-expressed*" and in the other sample it is "*under-expressed*".

We then find, for each of the instances, the size of the minimum cardinality α–β feature set, with $\beta=0$ and with α being fixed to the maximum *a priori* value which is possible for that instance. We have solved each of these problems to optimality using CPLEX (a mathematical programming software package). We found that there exists: an $(\alpha,\beta)=(283,0)$ feature set (with an optimal number of $k=402$ genes) for CNS; an $(\alpha,\beta)=(431,0)$ feature set with $k=663$ genes for Renal; $(\alpha,\beta)=(767,0)$ feature set with $k=1175$ genes for Leukaemia; an $(\alpha,\beta)=(338,0)$ feature set with $k=497$ for Colon; and an $(\alpha,\beta)=(605,0)$ feature set with $k=850$ for Melanoma.

Finally, we aim to try to find the maximal β achievable by a Max Cover α–β FEATURE SET (with α fixed to the previously obtained maximum *a priori* values), for each of the optimal cardinalities obtained in the previous step. We have solved each of this Max Cover α–β FEATURE SET problems to optimality so we found that there exists: an optimal Max Cover $(\alpha,\beta)=(283,264)$ feature set (with a number of $k=402$ genes) for CNS; an optimal Max Cover $(\alpha,\beta)=(431,363)$ feature set with $k=663$ genes for Renal; an optimal Max Cover $(\alpha,\beta)=(767,703)$ feature set with $k=1175$ genes for Leukaemia; an optimal Max Cover $(\alpha,\beta)=(338,283)$ feature set with $k=497$ for Colon; and an optimal Max Cover $(\alpha,\beta)=(605,610)$ feature set with $k=850$ for Melanoma.

3.3. Computational Results

The solutions are shown in Figure 1 (*a-e*) generated using the gene ordering algorithm of Moscato *et al* (2005a). It is clear that our method has uncovered not only a significantly larger number of genes that are differentially over-expressed, but more significant number of under-expressed genes in each group. The existence of those under-expressed genes is not cited in Ross *et al* (2000), even though they can certainly contribute to understand the pathways involved in the disease. These images also illustrate another source of useful information that is obtained by good orderings of the genes identified. For instance in Figure 1d, a large number of genes is differentially under-expressed in Leukaemia and Colon (see the lower half of the figure) yet markedly over-expressed in the other

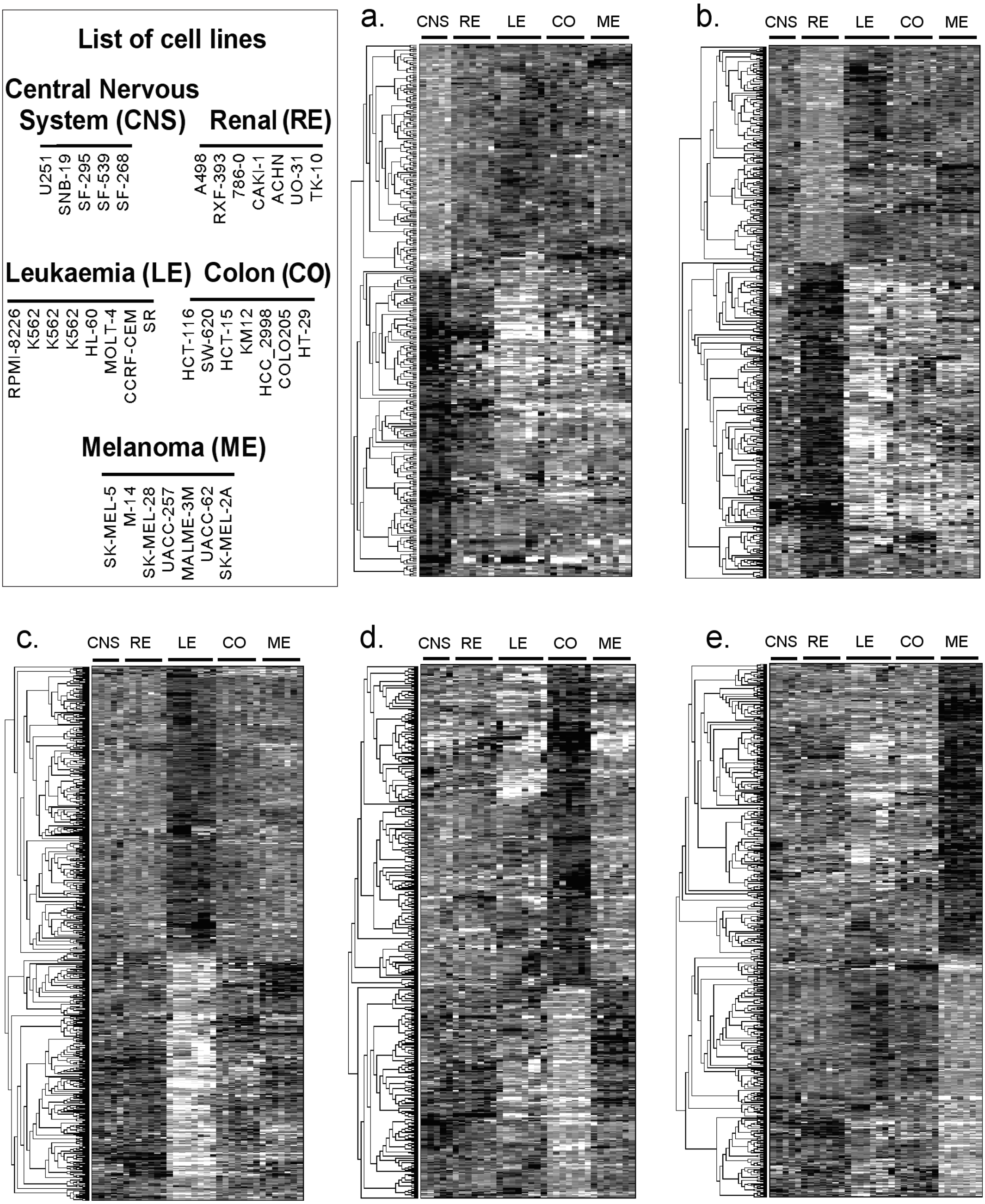


Figure 1: Genes consistently over-expressed/under-expressed in cell lines from five different types of cancer present in the NCI60 instance. They are: (a) Central Nervous System (CNS) – 402 genes; (b) Renal – 663 genes; (c) Leukaemia – 1,175 genes; (d) Colon – 497 genes; and (e) Melanoma – 850 genes. The cell lines that compose each group of samples are seen at the top-left diagram. Please refer to the supplementary material (*http://www.cs.newcastle.edu.au/~nbi/JRPIT/JRPIT_nci60sup.html*) for the colour version of the images and the complete list of genes.

cell lines. These expression similarities between different types of cell lines can also help understand the diseases and how they operate.

4. CONCLUSION

We have presented integer programming models and algorithms that have shown to be very useful to address the molecular classification of cancer from microarray data. The methodology is completely general and can have a much broader application (Moscato *et al*, 2005b). Our contribution also highlights the importance of safe data reduction methods that keep optimal solutions and maintain the relevant information in the dataset. Although the problem is NP-hard, for these instances the total computational time required to run each of the experiments was always below one second, in an Intel Xeon 1.8 GHz computer with 3Gb of RAM.

The results indicate that the method allows a good balance of discrimination between classes while keeping within-class consistency. This allows Life Science researchers to uncover a larger number of genetic pathways that could lead, in turn, to a broader picture of differential genetic regulation mechanisms. That is an indicative of the relevance and flexibility of the method, which would help to uncover yet unknown mechanisms that link genes, their products, and diseases.

REFERENCES

AKUTSU, T. and MIYANO, S. (2002): Selecting informative genes for cancer classification using gene expression data. In *Computational and Statistical Approaches to Genomics*, pp. 79-92. ZHANG, W. and SHMULEVICH, I. (eds). Kluwer Academic Publishers, Boston.

BØ, T.H., DYSVIK B. and JONASSEN, I. (2004): LSimpute: accurate estimation of missing values in microarray data with least squares methods. *Nucleic Acids Research* 32(3):e34.

BERRETTA, R., MENDES, A. and MOSCATO, P. (2005): Integer programming models and algorithms for molecular classification of cancer from microarray data. *Proc. 28th Australasian Computer Science Conference (ACSC2005), Newcastle, Australia. In Conferences in Research and Practice in Information Technology* 38:361-370, ESTIVILL-CASTRO, V. (ed), Australian Computer Society, Sydney.

BROWN, M.P., GRUNDY, W.N., LIN, D., CRISTIANINI, N., SUGNET, C.W., FUREY, T.S., ARES JR., M. and HAUSSLER, D. (2000): Knowledge-based analysis of microarray gene expression data by using support vector machines. *Proceedings of the National Academy of Science of the USA* 97(1):262-267.

CAPRARA, A., TOTH, P. and FISCHETTI, M. (2000): Algorithms for the set covering problem. *Annals of Operations Research* 98:353-371.

COTTA, C. and MOSCATO, P. (2003): The k-Feature Set problem is W[2]-complete. *Journal of Computer and System Sciences* 67:686-690.

COTTA, C., SLOPER, C. and MOSCATO, P. (2004): Evolutionary search of thresholds for robust feature set selection: application to the analysis of microarray data. *Proc. 2nd European Workshop on Evolutionary Bioinformatics (EvoBIO2004), Coimbra, Portugal. In Lecture Notes in Computer Science* 3005:21-30. RAIDL, G. *et al* (eds). Springer-Verlag, Berlin.

DASH, M. and LIU, H. (1997): Feature selection for classification. *Intelligent Data Analysis* 1(3):131-156.

DASH, M. and LIU, H. (2003): Consistency-based search in feature selection. *Artificial Intelligence Journal* 151(1-2):155-176.

DAVIES, S. and RUSSELL, S. (1994): NP-completeness of searches for smallest possible feature sets. *Proc. AAAI Fall Symposium on Relevance*, pp. 37-39. GREINER, R. and SUBRAMANIAN, D. (eds), New Orleans, USA.

EISEN, M., SPELLMAN, P., BROWN, P. and BOTSTEIN, D. (1998): Cluster analysis and display of genome-wide expression patterns. *Proceedings of the National Academy of Sciences of the USA* 95(25):14863-14868.

FAYYAD, U. and IRANI, K. (1993): Multi-interval discretization of continuous-valued attributes for classification learning. *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pp. 1022-1029, Chambéry, France.

FUREY, T.S., CRISTIANINI, N., DUFFY, N., BEDNARSKI, D.W., SCHUMMER, M. and HAUSSLER, D. (2000): Support vector machine classification and validation of cancer tissue samples using microarray expression data. *Bioinformatics* 16(10):906–914.

GARFINKEL, R.S. and NEMHAUSER, G.L. (1972): *Integer Programming*. John Wiley & Sons, New York.

ILOG (2005): ILOG CPLEX 9.0, http://www.ilog.com/products/cplex. Accessed 27-Feb-2006.

MOSCATO, P, BERRETTA, R. and MENDES, A. (2005a): A new memetic algorithm for ordering datasets: applications in microarray analysis. *Proc. MIC2005 – The 6th Metaheuristics International Conference* (CD-ROM), 695-700, Vienna, Austria.

MOSCATO, P., BERRETTA, R., MENDES, A., HOURANI, M. and COTTA, C. (2005b): Genes related with Alzheimer's disease: A comparison of evolutionary search, Statistical and integer programming approaches. *Proc. 3rd European Workshop on Evolutionary Bioinformatics (EvoBIO2005)*, Lausanne, Switzerland, 2005. In *Lecture Notes in Computer Science* 3449:84-94, ROTHLAUF, F. et al. (eds.), Springer-Verlag, Berlin.

MOSCATO, P., MATHIESON, L., MENDES, A. and BERRETTA, R. (2005c): The electronic primaries: Predicting the U.S. presidency using feature selection with safe data reduction. *Proceedings of the 28th Australasian Computer Science Conference (ACSC2005)*, Newcastle, Australia. In *Conferences in Research and Practice in Information Technology* 38:371-380, ESTIVILL-CASTRO, V. (Ed.), Australian Computer Society, Sydney.

QUACKENBUSH, J. (2001): Computational analysis of cDNA microarray data. *Nature Reviews* 2(6):418-428.

ROSS, D.T., SCHERF, U., EISEN, M.B., *et al* (2000): Systematic variation in gene expression patterns in human cancer cell lines. *Nature Genetics* 24(3):227-235.

TAMAYO, P. and RAMASWAMY, S. (2003): Cancer genomics and molecular pattern recognition. In *Expression profiling of human tumors: diagnostic and research applications*, 73-102. LADANYI, M. and GERALD, W. (eds). Humana Press.

WADDELL, P.J. and KISHINO, H. (2000): Cluster inference methods and graphical models evaluated on NCI60 microarray gene expression data. *Genome Informatics* 11:129-140.

XING, E.P. (2003): Feature selection in microarray analysis. In *Understanding and Using Microarray Analysis Techniques: A Practical Guide*, 110-131. BERRAR, D.P., DUBITZKY, W. and GRANZOW, M. (eds). Kluwer Academic Publishers.

BIOGRAPHICAL NOTES

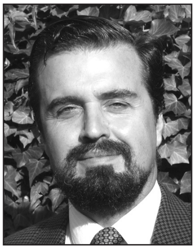
Dr Regina Berretta is a Lecturer at the School of Electrical Engineering and Computer Science, the University of Newcastle, and affiliated with the Newcastle Bioinformatics Initiative since 2003. She has a BSc in Computational and Applied Mathematics, a BSc in Mathematics, MSc and PhD titles in Operations Research from the State University of Campinas – UNICAMP, Brazil. Her research activities and interests are in mathematical modelling and combinatorial optimization problems arising in bioinformatics, with particular emphasis in functional genomics and biomarker discovery.

Regina Berretta

Alexandre Mendes is a Research Academic at the School of Electrical Engineering and Computer Science, at the University of Newcastle, Australia, and is a member of the Centre for Bioinformatics, Biomarker Discovery and Information-Based Medicine. He holds a PhD in Electrical Engineering (Automation) from the State University of Campinas, Brazil, and has been working in optimisation since 1997. Alexandre's current research is focused on mathematical modelling, optimisation techniques and their applications in bioinformatics and is currently being funded by the Australian Research Council.

Alexandre Mendes

Pablo Moscato is the founding director of the Newcastle Bioinformatics Initiative, a partner group of the Australian Research Council Centre in Bioinformatics. He is also the co-director of the Centre for Bioinformatics, Biomarker Discovery and Information-Based Medicine. A former member of the Caltech Concurrent Computation Program, he introduced Memetic Algorithms in 1989 and has successfully applied these hybrid methodologies for many large-scale optimization problems. Although mainly known by his contributions to heuristics, meta-heuristics and optimization in Computer Science, Pablo holds a Physics degree from the University of La Plata, Argentina, and a PhD in Electrical Engineering (Automation) from the State University of Campinas, Brazil.



Pablo Moscato