Application of Simulated Annealing to the Biclustering of Gene Expression Data

Kenneth Bryan, Pádraig Cunningham, Nadia Bolshakova

Abstract— In a gene expression data matrix a bicluster is a sub-matrix of genes and conditions that exhibits a high correlation of expression activity across both rows and columns. The problem of locating the most significant bicluster has been shown to be NP-complete. Heuristic approaches such as Cheng and Church’s greedy node deletion algorithm have been previously employed. It is to be expected that stochastic search techniques such as Evolutionary Algorithms or Simulated Annealing might improve upon such greedy techniques. In this paper we show that an approach based on Simulated Annealing is well suited to this problem and we present a comparative evaluation of Simulated Annealing and node deletion on a variety of datasets. We show that Simulated Annealing discovers more significant biclusters in many cases. Furthermore, we also test the ability of our technique to locate biologically verifiable biclusters within an annotated set of genes.

I. INTRODUCTION

In recent years the advent of DNA microarray technologies has revolutionised gene expression analysis. It is now possible to monitor the expression of thousands of genes in parallel over many experimental conditions (e.g. different patients, tissue types and growth environments), all within a single experiment (see Lander [16]). The results from these experiments are usually presented in the form of a data matrix in which rows represent genes and columns represent conditions. Each entry in the matrix is a measure of the expression level of a particular gene under a specific condition. Thorough analysis of these datasets aids in the annotation of genes of unknown function and the discovery of functional relationships between genes. This ultimately contributes to the elucidation of biological systems at a molecular level [3].

Gene expression datasets typically contain thousands of genes and hundreds of conditions and mining functional and class information from such large volumes of data presents a far from trivial task. One of the main methods used thus far to investigate the underlying structure of gene expression datasets has been cluster analysis [10], [25], [7]. In this approach genes showing similar expression activity over the set of conditions are grouped together into clusters. The premise behind this is that similarly behaving genes may be co-regulated and share a related function i.e. belong to a common pathway or a cellular structure. Conditions too may be clustered enabling disease types such as cancers to be defined in terms of their unique expression profiles [21]. Gene expression datasets are continually growing in size as more experiments are carried out and as experimental capacity improves. As datasets increase size it becomes less likely that objects (genes) will retain similarity across all attributes (conditions) making clustering problematic. Furthermore it is not uncommon for the expression of genes to be highly similar under one set of conditions and yet independent under another set [2]. Clustering genes over a subset similar of conditions would be more beneficial in such cases. This approach has been termed biclustering and was first introduced to gene expression analysis by Cheng and Church [6]. Cheng and Church identified the problem of finding significant biclusters as being NP-Hard and employed a greedy node deletion algorithm in their search. The review of biclustering algorithms for biological data analysis presented by Madeira and Oliveira [18] also identifies greedy search algorithms as a promising approach. Greedy search algorithms start with an initial solution and find a locally optimal solution by successive transformations that improve some fitness function. Stochastic methods such as Simulated Annealing (SA) [14] improve on greedy search due to their having the potential to escape local optima (see section III). In this paper we present a biclustering technique based on SA that improves on results produced by Cheng and Church’s node deletion algorithm (see section II). We carry out a comparative evaluation using both synthetic and real gene expression datasets and show that our SA based approach finds more significant biclusters in each dataset (see section V). We then use our SA algorithm to analyse an annotated set of genes with a view to discovering biologically verifiable biclusters.

II. BICLUSTERING

In general biclustering refers to the ‘simultaneous clustering’ of both rows and columns of a data matrix [20]. Hartigan pioneered this type of analysis in the seventies using two-way analysis of variance to locate constant valued sub-matrices within datasets. Biclustering may be viewed as a more specific type of sub-space clustering that enforces correlation within a subset of features (conditions) as well as a subset of objects (genes). This approach suits the gene expression context as related genes are thought to be regulated in a synchronised fashion and over certain conditions [2]. Therefore discovering the dominant biclusters within a gene expression dataset may aid the discovery of these co-regulated groups. More recently, inspired by Hartigan’s so called ‘direct clustering’ approaches [12] the concept was introduced to the area of gene expression analysis by Cheng and Church [6]. Since then several alternative biclustering approaches have been taken within gene expression analysis. One approach taken by Tanay et al. [24] likens biclustering to the search for
developed in other studies [24]. Cheng and Church designed a graph which has been proven to be NP-Hard [13]. Biclustering maximum bicluster search to that of locating a maximum of a few hundred elements. Cheng and Church likened the number of possible sub-matrices increases exponentially relationships between the largest number of objects. However δ-biclusters, would be of most interest as they capture the structure. For a review of the above approaches the reader is directed to [18]. These approaches are less intuitive and theoretically quite different from that of Cheng and Church.

Cheng and Church defined a bicluster to be a subset of genes and a subset conditions with a high similarity score, where similarity is a measure of the coherence of genes and conditions in the subset. A group of genes are said to be coherent if their level of expression reacts in parallel or correlates across a set of conditions. Similarly, a set of conditions may also have coherent levels of expression across a set of genes. Cheng and Church developed a measure, called the mean squared residue score, which takes into account both row and column correlations and therefore makes it possible to simultaneously evaluate the coherence of rows and columns within a sub-matrix. They thus defined a bicluster to be a sub-matrix composed of subsets of genes and conditions with a low mean squared residue score (the lower the score the better the correlation of the rows and columns). The residue score of an entry \( a_{ij} \) in a bicluster \( B(IJ) \) (where \( I \) is the subset of rows and \( J \) the subset of columns making up the bicluster) is a measure of how well the entry fits into that bicluster. It is defined to be:

\[
R(a_{ij}) = a_{ij} - a_{i\cdot} - a_{\cdot j} + a_{\cdot \cdot}
\]

where \( a_{ij} \) is the mean of the \( i \)th row in the bicluster, \( a_{i\cdot} \) is the mean of the \( i \)th column and \( a_{\cdot j} \) mean of the whole bicluster. The overall mean squared residue score is:

\[
H(I,J) = \frac{1}{|I||J|} \sum_{i \in I, j \in J} R(a_{ij})^2
\]

The next problem to be tackled is how locate these low scoring biclusters within a parent data matrix. The deterministic approach is to sequentially run through all the possible combinations of rows and columns of the data matrix and find the sub-matrices which satisfy a predefined low score, \( \delta \) (the set of \( \delta \)-biclusters). The most significant biclusters, the largest \( \delta \)-biclusters, would be of most interest as they capture the relationships between the largest number of objects. However the number of possible sub-matrices increases exponentially with the size of the parent matrix making this task practically impossible when the matrix exceeds the fairly modest size of a few hundred elements. Cheng and Church likened the maximum bicluster search to that of locating a maximum biclique (largest complete sub-graph) within a parent bipartite graph which has been proven to be NP-Hard [13]. Biclustering based upon this graph theoretic paradigm was more fully developed in other studies [24]. Cheng and Church designed a set of heuristic algorithms to locate these \( \delta \)-biclusters sequentially in a top-down manner by deleting the row and column nodes from the parent matrix which most improve the mean squared residue score. Upon reaching the \( \delta \) threshold a node addition phase is then carried out to add rows/columns which may have been missed. Inversely correlated rows, which may represent negatively regulated genes, are also added at this stage. A subsequent study [27] noted that as with other greedy searches there is a possibility that the system may become trapped at a locally good solution. It is thus unlikely that the global maximum or maximal \( \delta \)-bicluster will be found. Applying a stochastic search technique to locate this global maximum seems to be the next logical step in the bicluster search problem.

III. SIMULATED ANNEALING

Stochastic techniques which allow acceptance of reversals in fitness have been shown to improve on greedy approaches by performing more in-depth searches of solution space. Recently evolutionary optimization schemes employing the mean squared residue function have been used to tackle the bicluster search problem [5], [1]. These attempts failed to find more significant solutions than the Cheng and Church technique in terms of bicluster size and instead focused on returning sets of smaller biclusters with high row variability.

Simulated Annealing is a well established stochastic technique originally developed to model the natural process of crystallisation [19] and later adopted to solve optimisation problems [14]. As with a greedy search it accepts all changes that lead to improvements in the fitness of a solution. However it differs in its ability to allow the probabilistic acceptance of changes which lead to worse solutions i.e. reversals in fitness. The probability of accepting a reversal is inversely proportional to the size of the reversal with the acceptance of smaller reversals being more probable. This probability also decreases as the search continues or as the system cools allowing eventual convergence on a solution. It is defined by Boltzman’s equation:

\[
P(\Delta E) \propto e^{-\frac{\Delta E}{T}}
\]

where \( \Delta E \) is the difference in energy (fitness) between the old and new states and \( T \) is the temperature of the system. In the virtual environment the temperature of the system is lowered after certain predefined number of accepted changes, successes, or total changes, attempts, depending on which is reached first. The rate at which temperature decreases depends on the cooling schedule. In the natural process the system cools logarithmically however this is so time consuming that many simplified cooling schedules have been introduced for practical problem solving; the following simple cooling model is popular:

\[
T(k) = \frac{T(k-1)}{1 + \sigma}
\]

where \( T(k) \) is the current temperature, \( T(k-1) \) is the previous temperature and \( \sigma \) dictates the cooling rate. Simulated Annealing has been applied to such problems as the well known travelling salesman problem [4] and optimisation of wiring
on computer chips [14]. Its application to biclustering gene expression data is a logical step given the drawbacks of current approaches.

IV. EXPERIMENTAL METHODS

A. Biclustering using Simulated Annealing

Several parameters are common to every Simulated Annealing implementation. The most crucial parameter is the fitness function or how to quantitatively define whether the solution improves or not after a perturbation. The mean squared residue score was used as a measure of bicluster fitness in this study. The annealing schedule used was of the type in equation 4 with \( \sigma = 0.1 \). Consequently each subsequent temperature is approximately 0.9 times that of the previous temperature. In Simulated Annealing it is also important to ensure that an adequate search is performed at each temperature. This is dictated by the number of attempts or successes that occur before each reduction in system temperature. The selection of the number of successes and attempts depends on the depth and size of the search space as determined by the size and dimensionality of the dataset. In this study the number of successes needed to be achieved before cooling occurs was set to be equal to 10 times the number of genes. The number of attempts at each temperature is 10 times this again (this ensures a good search even if there are not the required number of successes). So for a dataset of 1000 genes the system would only lower the temperature after 10,000 successes or 100,000 attempts. Another important parameter is the initial temperature of the system, \( T_0 \). If this parameter is set too high the system will take too long to converge and if it is set too low the proportion of the search space covered will be much reduced. It has been found by experiment that in general an optimal starting temperature is one which allows 80 percent of reversals to be accepted [22].

Our Simulated Annealing Biclustering (SAB) algorithm begins the search in a top-down manner with the initial solution containing all rows and columns. The solution is then iteratively perturbed by the deletion or addition of rows or columns with the mean squared residue being recalculated each time. In Figure 1 the method for generating a new solution is detailed. This method takes into account the number of rows and columns in the current solution and adjusts the probability of a row or column flip accordingly. Note that a minimum solution size of 10×10 was chosen. This was deemed to represent the minimum significant size of a solution in this study. So for example, if genes correlate over 10 conditions it is likely that they may be related. This minimum solution size also prevents the search from ending on a trivial bicluster of one row or one column and score 0. To allow the comparison of SAB with the node deletion algorithm, some way needed to be found to return biclusters of a chosen \( \delta \) value. Upon reaching a \( \delta \)-bicluster the minimum solution size is then reset to that of the \( \delta \)-bicluster. The Simulated Annealing then continues but with the added proviso of accepting solutions less than or equal to the \( \delta \)-score that are larger in size. This gradually increases the size of the \( \delta \)-bicluster. Also, to align SAB with the Cheng and Church node deletion approach the node addition is performed after the search. This adds any missed rows or columns and, importantly for SAB alignment, adds inversely correlated rows which may represent negatively regulated genes. Upon the discovery of a bicluster Cheng and Church masked the solution with randomly imputed numbers from the same range as the dataset. This prevents the bicluster from being rediscovered by the deterministic node deletion algorithm. We use the same method of masking discovered solutions. Typically, using the parameters given above and for a dataset of 3000 genes and 20 conditions the search takes about 60 minutes to converge on a bicluster solution. As a result of the masking of solutions this convergence time is reduced for subsequent bicluster searches.

B. Datasets Used

Cheng and Church chose a yeast cell cycle dataset\(^1\) in their study. This dataset contains 2,884 genes and 17 conditions.

\(^1\)http://arep.med.harvard.edu/biclustering/yeast.matrix
We used this dataset and two additional real datasets to compare our SAB algorithm with node deletion. The first additional dataset contains 27 conditions and 2,774 genes. It is derived from a study on scleroderma, a potentially serious skin disorder which affects epithelial cells [26]. This dataset contains gene expression data from both normal and affected patients. The second additional dataset of 3051 genes and 38 conditions representing different classes of lymphoma was distilled from a larger dataset [11] using techniques described in [9] to enrich the dataset with genes with the highest variance across conditions. A synthetic dataset was also used to compare the algorithms. The attraction of a synthetic dataset is that all the major biclusters can be defined and embedded in the data. The success of bicluster discovery can then be more quantitively measured. We have developed a synthetic dataset construction technique and believe it to be a more faithful rendering of reality than previous approaches[23]. A dataset of size 100x100 was constructed. Biclusters were generated using real gene profiles from the yeast dataset as templates. Firstly an expression level shift was added to the template gene profile vector. The amount of shift is chosen randomly for each additional artificial profile and maintained within a user defined range. This resulting spread mirrors the expression level variations which occur in vivo and also makes individual expression profiles more discernable within the bicluster. As it stands the bicluster has a perfect score of 0. Some error needs to be introduced to reflect the in vivo model. Each expression value in each artificial profile is then augmented by a correlation error ($E_s$): 

$$E_s = \sigma(x) \cdot \epsilon \cdot r_s$$  

(5)

where $\sigma(x)$ is the standard deviation of the template gene profile (this scales the error for the particular template), $\epsilon$ is a user defined constant in a range [0,1] (this variable dictates the level of error and the quality of the biclusters) and $r_s$ is a random variable in a range [-1,1] (this enables the expression level of the generated profile to be greater or less than the template). Given that the original gene profile template is defined as:

$$X = \{x_1, ..., x_n\}$$  

(6)

the newly constructed correlating profile will be given as:

$$Y = \{x_1 + (S + E_1), ..., x_n + (S + E_n)\}$$  

(7)

where $x$ represents a particular expression value, $S$ is the shift applied to the vector and $E$ is the correlation error applied to each correlating expression value. An illustration of the template profile and a generated profile is shown in Figure 2.

In all five biclusters were constructed of sizes 10x10, 20x10, 10x20, and two 10x10 overlapping biclusters and embedded in a background randomly generated within the same range as the biclusters (0-600).

V. Evaluation of Biclustering Using Simulated Annealing

There are three questions dealt with in the evaluation section. Firstly we investigate whether SAB can retrieve solutions closer to the global maximum than Cheng and Church’s node deletion (ND) approach i.e. larger $\delta$-biclusters. We then investigate using a synthetic dataset the ability of SAB to discover all the bicluster signals within a dataset. Lastly we use an annotated dataset to investigate whether biclusters discovered by SAB reflect in vivo functional modules i.e. whether SAB can discover biologically verifiable biclusters.

A. Comparative Evaluation with Node Deletion

Cheng and Church carried out node deletion on the yeast dataset mentioned above and used a mean squared residue threshold ($\delta$) of 300 (as determined by equation (2)). The SAB algorithm was applied to the same yeast dataset. In this study $\delta$ thresholds of 300, 200 and 100 were set and the size of the discovered biclusters compared in each case. SAB produces biclusters of at least 10 columns (conditions) in width. To ensure that the column size of the resultant biclusters does not bias the results an adjusted node deletion algorithm (ND2) is also run in which the column size of resultant biclusters is set to 10. This is achieved in ND2 by prioritising column deletion until the minimum threshold of 10 has been reached whereupon row deletion alone is carried out. Figure 3(a) shows the size of the first bicluster found by ND, ND2 and SAB over the various $\delta$ thresholds for the yeast dataset. Figure 3(b) shows the second bicluster discovered when the first was masked with random numbers as described in section IV. SAB performed better than ND and ND2 on the yeast data for all $\delta$-scores locating larger $\delta$-biclusters in all cases. The results for all three data sets are shown in Table I, numbers in bold mark the best biclusters. Biclusters in italics are taken from a significantly larger dataset (after masking a smaller bicluster) and cannot be compared with neighbouring values in the second bicluster column. SAB performed better than ND discovering larger $\delta$-biclusters in all cases. The ND2 algorithm performed better than the original ND but even so SAB still performed better in most cases. It can be seen that SAB performs better than ND2 discovering a larger first bicluster in 4/9 cases and draws in a further 3 cases. SAB performed best in discovering the second biclusters in 6/9 cases over the three datasets.
Comparison of Biclustering Algorithms (Yeast Data)

(a) Comparison of Biclustering Algorithms (Yeast Data)

(b) Comparison 2nd Bicluster (Yeast Data)

Fig. 3. (a) Comparisons of Cheng and Church’s node deletion algorithm (ND), our adjusted node deletion algorithm (ND2) and Simulated Annealing Biclustering (SAB) using the yeast dataset over δ-scores of 300, 200 and 100. (b) The second biclusters found by CCND2 and SAB.

TABLE I

Comparison of biclusters discovered in each real dataset.

<table>
<thead>
<tr>
<th></th>
<th>ND</th>
<th>ND2</th>
<th>SAB</th>
<th>ND</th>
<th>ND2</th>
<th>SAB</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Bicluster 1</td>
<td>Bicluster 2</td>
<td>Bicluster 1</td>
<td>Bicluster 2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Yeast</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>15165</td>
<td>15750</td>
<td>16460</td>
<td>9012</td>
<td>3930</td>
<td>8320</td>
</tr>
<tr>
<td>200</td>
<td>8465</td>
<td>9540</td>
<td>10360</td>
<td>4972</td>
<td>2630</td>
<td>3860</td>
</tr>
<tr>
<td>100</td>
<td>2520</td>
<td>2700</td>
<td>2940</td>
<td>1260</td>
<td>830</td>
<td>1390</td>
</tr>
<tr>
<td>Scleroderma</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>15190</td>
<td>18260</td>
<td>18230</td>
<td>4120</td>
<td>6780</td>
<td>6310</td>
</tr>
<tr>
<td>200</td>
<td>7296</td>
<td>12920</td>
<td>13210</td>
<td>7876</td>
<td>3290</td>
<td>4030</td>
</tr>
<tr>
<td>100</td>
<td>2730</td>
<td>5170</td>
<td>5140</td>
<td>1570</td>
<td>830</td>
<td>850</td>
</tr>
<tr>
<td>Lymphoma</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>300</td>
<td>1344</td>
<td>3320</td>
<td>3220</td>
<td>518</td>
<td>1740</td>
<td>1810</td>
</tr>
<tr>
<td>200</td>
<td>1032</td>
<td>2510</td>
<td>2460</td>
<td>300</td>
<td>1370</td>
<td>1200</td>
</tr>
<tr>
<td>100</td>
<td>851</td>
<td>1780</td>
<td>1790</td>
<td>136</td>
<td>1050</td>
<td>810</td>
</tr>
</tbody>
</table>

B. Bicluster Retrieval in Synthetic Data

It has been shown in the previous section that SAB has the ability to retrieve more significant biclusters than node deletion. However it is difficult to know how successful SAB is in recovering all the significant signals in this real data. To test this aspect of SAB we decided to use a synthetic data set constructed in the manner described in IV-B. We then carried out a comparative evaluation of SAB and node deletion using this dataset and measured their ability to recover the five embedded biclusters. From Table II it can be seen that both node deletion and SAB discover five bicluster signals within the synthetic dataset but SAB retrieves substantially more of the embedded biclusters.

TABLE II

Biclusters Recovered from Synthetic Dataset

<table>
<thead>
<tr>
<th>Bic.</th>
<th># Genes</th>
<th>Dominant FM</th>
<th>Genes in FM</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>81</td>
<td>Ribosomal Proteins(96)</td>
<td>61</td>
<td>2.15×10⁻³⁸</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Glycolysis/Gluconeogenesis(26)</td>
<td>5</td>
<td>0.16</td>
</tr>
<tr>
<td>2</td>
<td>59</td>
<td>Basal Transcription Factors(10)</td>
<td>6</td>
<td>1.6×10⁻⁴</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nucleotide Metabolism(81)</td>
<td>16</td>
<td>4.02×10⁻³</td>
</tr>
</tbody>
</table>

C. Biological Interpretation

A further way to evaluate SAB is to use a fully annotated dataset. To our knowledge this precise approach to bicluster evaluation has not been used before in the literature. Of the 2884 genes in the yeast dataset 550 can be annotated from the online database called the Kyoto Encyclopaedia of Genes and Genomes, KEGG3. Ideally the biclusters in such a dataset would then reflect in vivo groups of genes known to be functionally related. Because of the smaller size of the annotated dataset a δ-score of 100 was chosen as the mean squared residue threshold. In Table III it can be seen that the first bicluster discovered from this annotated dataset is rich in genes from the ribosomal functional category. The second bicluster contains transcription factors and genes involved in nucleotide metabolism. These genes are the main regulators of protein production and gene expression in the cell. The statistical significance of discovering each functional category is given in terms of p-values as formulated in [8]. Further biclusters contained correlating genes but no dominating known functional categories, so are not listed. This may be partly due to the incomplete nature of the dataset as only the annotated genes were used from the yeast data.

TABLE III

Known functional modules (FM) found by SAB in the annotated gene dataset.

<table>
<thead>
<tr>
<th>Bic.</th>
<th># Genes</th>
<th>Dominant FM</th>
<th>Genes in FM</th>
<th>P-value</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>81</td>
<td>Ribosomal Proteins(96)</td>
<td>61</td>
<td>2.15×10⁻³⁸</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Glycolysis/Gluconeogenesis(26)</td>
<td>5</td>
<td>0.16</td>
</tr>
<tr>
<td>2</td>
<td>59</td>
<td>Basal Transcription Factors(10)</td>
<td>6</td>
<td>1.6×10⁻⁴</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Nucleotide Metabolism(81)</td>
<td>16</td>
<td>4.02×10⁻³</td>
</tr>
</tbody>
</table>

VI. CONCLUSIONS & FUTURE WORK

Using SAB we have shown that stochastic methods have the potential to give improved results for the bicluster search problem. SAB discovers more significant biclusters than Cheng and Church’s original node deletion approach. The biclusters discovered by SAB are considered more significant as they have the same level of row/column correlation as those discovered by Cheng and Church (as measured by δ) and yet are larger in size. One could have also taken the approach to search for biclusters of the same size and lower δ-score. However this approach would yield information on fewer genes/conditions and direct comparison with Cheng and Church’s results would not

3http://www.genome.jp/kegg/genes.html
be possible. SAB also performs better when compared to our improved version of the node deletion algorithm. Furthermore we have shown that SAB discovers more complete biclusters than node deletion when using the synthetic dataset. When applied to the annotated yeast dataset SAB discovers biclusters which represent recognisable classes of genes. In the annotated dataset SAB discovered just two biclusters which had an over-representation of known functional groups. Apart from the incompleteness of this annotated dataset, as mentioned in V-C, this may be due to the manner in which SAB searches for biclusters. SAB works in top-down manner with the mean squared residue function promoting the deletion of rows/columns which do not fit in with the trends in the dataset. As a result, biclusters may be biased towards core regulatory genes which govern the general state of gene expression the cell. Outlying biclusters would tend to have their ill-fitting rows/columns deleted early on in the search. Evidence of this can be seen in the nature of the classes of genes in biclusters 1 and 2 from the annotated set. Perhaps this bias could be harnessed to discover regulatory genes within gene expression data. Although a bottom-up search approach using the mean squared residue as a fitness function would probably not find such large biclusters it would perhaps promote more variability in the classes of genes it discovers. In future research we intend to use Simulated Annealing in a bottom-up search in a bid to discover smaller more natural biclusters which may better reflect the natural state or organisation in an organism.

References


Kenneth Bryan received a B.A. Mod (Hons) degree in Microbiology from Trinity College Dublin in 2001. In 2002 he received a Graduate Diploma in Information Technology from Dublin City University. He is working towards a PhD degree in the computational analysis of gene expression data. His current work involves developing heuristic methods for efficient biclustering of gene expression data.

Pádraig Cunningham is an Associate Professor in the Department of Computer Science in Trinity College Dublin. His current research focus is on machine learning, specifically case-based reasoning, clustering and ensemble methods. He has a B.E. and M.Eng.Sci. from the National University of Ireland and a Ph.D. from Dublin University which he received in 1989. He became a Fellow of TCD in 1998. After completing his PhD he worked with Digital Equipment Corporation as a Software Engineer and with Hitachi Europe Ltd. as a Research Scientist. In 2004 he was elected to Fellowship of ECCI, the European Coordinating Committee for Artificial Intelligence Research.

Nadia Bolshakova is a Research Fellow at the Department of Computer Science in Trinity College Dublin. She is developing research in the areas at the intersection of computer science and life sciences, such as Bioinformatics and Intelligent Systems. She holds a Ph.D. from the State Technical University, Russia, which she received in 2000. She also took part in research projects in the Institute of Cytology of Russian Academy of Sciences. Dr. Bolshakova has published number of articles in journals, conference proceedings and books related to bioinformatics, biomodelling and cell biology. She has been a referee for several scientific journals. Her current research project deals with developing of methods, tools and systems to support research on microarray data, protein secondary-structure prediction and other related application domains.