Biclustering of Expression Data Using Simulated Annealing*

Kenneth Bryan, Pádraig Cunningham, Nadia Bolshakova

Trinity College Dublin, College Green, Dublin 2, Ireland Kenneth.Bryan@cs.tcd.ie

Abstract. In gene expression data a bicluster is a subset of genes and a subset of conditions which show correlating levels of expression. However, the problem of finding significant biclusters in gene expression data grows exponentially with the size of the dataset. This means that exhaustive search for good biclusters is not feasible in real datasets so greedy search techniques such as Cheng and Church's node deletion algorithm have been used. It is to be expected that stochastic search techniques such as Genetic Algorithms or Simulated Annealing might produce better solutions than greedy search. In this paper we show that a Simulated Annealing approach is well suited to this problem and we present a comparative evaluation of Simulated Annealing and node deletion on a variety of datasets and show that Simulated Annealing discovers more significant biclusters in many cases.

1 Introduction

In recent years the advent of DNA microarray technologies has revolutionised the study of gene expression. It is now possible to monitor the expressions of thousands of genes in parallel over various experimental conditions (e.g. different patients, tissue types and growth environments), all within a single experiment (see Lander [1]). Results from multiple experiments may be combined potentially producing datasets containing thousands of genes monitored over hundreds of conditions. These results are usually presented in the form of a table or data matrix where rows represent genes and columns represent conditions. The value in each cell in the matrix represents the expression level of a particular gene under a specific condition. Thorough analysis of these datasets aids in the annotation of gene function and the discovery of related genes and conditions which ultimately contributes to the elucidation of biological systems at a molecular level [2]. However mining this valuable information from such large volumes of data presents a far from trivial task.

Clustering is important in gene expression analysis because it is often useful to group genes according to their expression levels under multiple conditions. It may also be useful to cluster conditions based on the expression of different genes. These correspond to clustering the rows or the columns of the data array.

* This research was sponsored by Science Foundation Ireland under grant number SFI-02/IN1/I111

It is a characteristic of this gene expression data that it is not necessary to include *all* the rows or columns in these clusters. In fact it may be useful to identify a subset of the conditions where a subset of the genes act in a coherent manner - this is termed *biclustering* [3; 4].

The review of biclustering algorithms for biological data analysis presented by Madeira and Oliveira [4] 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 on that solution that improve some fitness function. Simulated Annealing (SA) [5] is an improvement on greedy search in that it has the potential to escape local minima (see section 3). In this paper we present a biclustering technique based on SA that improves on results produced by Cheng and Church's node deletion algorithms that employ greedy search [3] in most of the cases examined. We perform this evaluation on three datasets derived from human and yeast expression studies and show that our SA based solution finds more significant biclusters in each dataset. The measure of significance we use is the *mean-square residue score* proposed by Cheng and Church [3].

Before presenting this evaluation we provide an overview of cluster analysis and biclustering in section 2 and a brief introduction to Simulated Annealing in 3. The experimental methods are presented in section 4 and the evaluation and results are given in section 5. The paper concludes with some conclusions and an outline of avenues for future work in section 6.

2 Cluster Analysis and Biclustering

One of the main methods used to discover patterns within gene expression data has been cluster analysis (or simply clustering). Well established in other fields [6], clustering has been integrated with some success into the gene expression analysis fold [7], [8], [9]. Clustering is an unsupervised learning technique that attempts to model the trends within data in order to uncover previously unknown relationships and classifications. Therefore it is an ideal technique to apply to gene expression data which in many cases may represent a plethora of biological systems and gene relationships not as of yet discovered or fully described. Cluster analysis groups objects on the basis of their similarity as determined by a chosen metric which may vary according to context. In gene expression analysis, similarity of expression is best computed using a correlation score (e.g. Pearson's Correlation Coefficient) of the row vectors within the data matrix [7]. Grouping genes in this way aids annotation of uncharacterised genes and the elucidation gene regulatory networks. The clustering of conditions may also be performed by grouping on the basis of column vector correlation. In this way conditions, such as cancer types, which may be difficult to clinically classify, may be related on a molecular level [10].

In the field of gene expression analysis, datasets are continually growing in size as the experimental capacity improves and as more microarray experiments are performed. As the size of datasets increase it becomes increasingly unlikely

that objects will retain similarity across all features - thus making clustering problematic. The biological context exacerbates this problem as it is not uncommon for the expression of genes to be highly similar under one set of conditions and yet independent under another set [11]. Therefore in these large expression datasets better, more comprehensive clusterings may be achieved by examining similarity over a subset of genes or conditions only.

A variation on the clustering approach that supports the discovery of these local signals within a dataset is that of biclustering. In general biclustering refers to the 'simultaneous clustering' of both rows and columns of a data matrix [12]. 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 thought of 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) - clustering both like objects and like features. This approach suits the gene expression context as related genes are thought to be regulated in a synchronised fashion and over certain conditions [13] 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 [14] the concept was introduced to the area of gene expression analysis by Cheng and Church [3]. Since then several similar approaches have been taken within gene expression analysis [15], [16], [17].

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 score called the *mean squared residue score* and 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 a subset of genes and subset of 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 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:

$$R(a_{ij}) = a_{ij} - a_{Ij} - a_{iJ} + a_{IJ} \tag{1}$$

where a_{iJ} is the mean of the ith row in the bicluster, a_{Ij} is the mean of the jth column and a_{IJ} mean of the whole bicluster. The overall *mean squared residue score* is:

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

The problem then is how to best locate these biclusters with low scores within a parent data matrix. The obvious 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, δ (the set of δ-biclusters). However the number of possible sub-matrices increases exponentially with the size of the parent matrix and this task becomes 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 [18]. Biclustering based upon this graph theoretic paradigm was more fully developed in other studies [16]. Cheng and Church designed a set of heuristic algorithms to locate these δ-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.

In a subsequent study [19] a weakness was noted in this approach in that there was the possibility that the system would become trapped at a local minimum so the maximal or best bicluster is unlikely to be found. Thus significant relationships within the data may be overlooked.

3 Simulated Annealing

Many optimisation techniques, such as the node deletion method outlined above, are prone to the problem of becoming trapped at local minima. A local minimum represents a good solution but not the best or global solution. This occurs when a greedy search finds a good solution which cannot be improved upon given the locality of the search space. In such a scenario one way to expand the search space is to accept changes that do not immediately improve the fitness but which allow the search to explore other areas outside the locality for better solutions.

Simulated Annealing is a well established stochastic approach used to overcome this feature of greedy optimisation. This process was originally developed by Metropolis [20] to model the cooling and crystallisation of materials such as glass and metals. Subsequently Kirkpatrick [5] noted that this model could be adapted to solve optimisation problems and provide a way for a system to escape local minima. Classic optimization techniques accept only improvements in the system as measured by a chosen fitness function. Simulated annealing differs in that it allows the probabilistic acceptance of changes in the system that lead to a temporary disimprovement of the fitness score.

In a cooling crystalline solid the probability of accepting change which leads to a higher energy state $P(\Delta E)$ is defined by Boltzman's equation:

$$P(\Delta E) \propto e^{-\frac{\Delta E}{T}} \tag{3}$$

Where ΔE is the difference in energy between the old and new states and T is the amount of random energy available i.e. the temperature of the system. The crystalline structure which results represents the lowest possible energy state or global solution for the system. Simulated Annealing models this natural process in a bid to discover a solution closer to this global optimum. In the virtual environment a variable corresponding to temperature is gradually decremented to

ensure a convergence at some point. In the natural process the system cools logarithmically, this type of annealing schedule is referred to as boltzman's annealing. It is given by the equation:

$$T(k) = \frac{T_0}{\log k} \tag{4}$$

Boltzmann annealing is so time consuming that many alternative simplified cooling schedules have been introduced for practical problem solving; the following linear cooling model is popular.

$$T(k) = \frac{T(k-1)}{1+\sigma} \tag{5}$$

Simulated Annealing has been applied to such problems as the well known travelling salesman problem [21] and optimisation of wiring on computer chips [5]. Its application to biclustering within the area of gene expression is a logical step given the drawbacks of current approaches.

4 Experimental Methods

4.1 Biclustering using Node Deletion

Church and Cheng's biclustering implementation is composed of three algorithms. The first is a multi-node deletion algorithm, designed to speed up the processing of large datasets. However this also diminishes the accuracy of the search result, potentially reducing the size of discovered biclusters, and therefore is precluded from the evaluation. The second algorithm is a single node deletion algorithm which is the main greedy search part of the approach. Also included in the comparisons is a node addition algorithm, designed to search the remaining matrix subsequent to the discovery biclusters for missed rows or columns. This addition algorithm also adds in anti-correlated or inverted rows which may represent negatively regulated genes. This addition algorithm contributes significantly to the size of the biclusters produced by addition of these anti-correlated rows.

The node deletion algorithm begins with the parent matrix and iteratively removes the rows and columns which most improve the fitness function score. This score is based on the concept of an entry's residue score and mean square residue scores introduced in equations 1 and 2). Every row and column in the matrix is assigned a row and column score, which is the mean squared residue for the entries in the row/column. This row/column score represents how well each row/column fits in with the rest of the data in the matrix. The row or column with the highest (worst) mean squared residue score is deleted. The algorithm doesn't allow any rows/columns to be dynamically added back in during this greedy search and it is only after the greedy search has finished that node addition algorithm is executed. A dynamic addition of rows/columns to improve the fitness score is one way in which the search could be augmented.

4.2 Biclustering using Simulated Annealing

Several parameters are common to every Simulated Annealing implementation. The first most obvious is the fitness function or how to quantitatively define whether the system improves or not after a perturbation. Cheng and Church's mean squared residue score was used as the fitness function in this study. The annealing schedule used was of the type in equation 5 with $\sigma = 0.1$. The means that each subsequent temperature is approximately 0.9 times that of the previous temperature. Also important in Simulated Annealing is how long the search spends at each temperature to ensure the search space a has been adequately explored. This is measured in terms of how often the system is perturbed(*attempts*) and how many times this perturbance is accepted (*successes*). These values are linked to the size of the search space and are generally a multiple of the number of objects in the dataset, in this case the number of genes. In this study the number of Successes was set to be equal to 10 times that of the number of genes with the number of Attempts 10 times that again. So for a dataset of 1000 genes between 10,000 and 100,000 changes would be made to the system at each temperature before it is lowered. The system begins with all rows and columns included and the search space is traversed by both deletion and re-addition of rows and columns.

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 potential search space 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]. The minimum size of a result was set to 10x10 otherwise the search would continue deleting rows and columns until a trivial bicluster of size one row or one column and score 0 remained. It is thought that a correlation over a minimum of 10 rows by 10 columns represents a significant relationship between genes and conditions. Because of the large inequality in the number of columns and rows in the datasets used the resultant biclusters tend to contain the minimum 10 columns.

Furthermore, in order to align the Simulated Annealing with the Cheng and Church node deletion algorithm and allow comparison of solutions, some way needed to be found to allow the Simulated Annealing to locate biclusters of a chosen δ value. Firstly the search began in a top-down way similar to node deletion with all rows and columns being included. Upon reaching a δ-bicluster solution the size of the solution is constrained to that minimum. Then a bottom up search ensues from this point in which the size of the bicluster is increased but the score maintained around δ. Also to align the Simulated Annealing with the Cheng and Church node deletion approach the node addition is performed after the search. This mainly adds anti-correlated genes to the bicluster.

4.3 Datasets Used

Cheng and Church chose a yeast cell cycle dataset on which to perform biclustering. This dataset contains 2,884 genes and 17 conditions and is available

at http://arep.med.harvard.edu/biclustering/yeast.matrix. Simulated annealing was used to perform biclustering on this dataset in order to compare and evaluate this new approach. Many of the genes in this dataset vary little in expression across the 17 conditions so the largest biclusters are inclined to be quite flat. Also 17 conditions are not many in which to find significant sub-signals of 10 columns or more. Two additional datasets, both larger in terms of the number of conditions and more variable in terms gene expression across the conditions, are also used to compare the two biclustering methods. The first dataset containing 27 conditions and 2,774 genes is derived from a study on scleroderma, a potentially serious skin disorder which affects epithelial cells [23]. This dataset contains genes expression data from both normal and affected patients and can be found at http://genome-www.stanford.edu/scleroderma/data.shtml. The last dataset of 3051 genes and 38 conditions representing different classes of lymphoma was also used to compare the two biclustering techniques. This dataset was distilled form a larger dataset [24] using techniques described in [25] to enrich the dataset with genes with the highest variance across conditions.

5 Evaluation of Biclusters

A further difficulty with biclustering is the lack of established evaluation methods. Parameters often used for cluster evaluation such as inter and intra cluster distances have reduced meaning when one considers biclusters which may be composed of groupings in different dimensions and have potentially overlapping relationships i.e. biclusters could share a number of features (conditions) or objects (genes). There are two questions dealt with in the evaluation section. The first question is whether Simulated Annealing can retrieve solutions closer to the global optimum. In the bicluster search a solution closer to the global optimum would be represented by discovering larger δ-biclusters than the Cheng and Church algorithm. The second question is related to one which is continually being debated in biology - whether genes which show similar expression can be classed into functional modules which bear out under practical scrutiny. In this paper we use an annotated gene set to investigate whether Simulated Annealing discovers such verifiable biclusters.

5.1 Comparisons with Node Deletion

Cheng and Church chose biclusters with a correlation threshold as measured by the mean squared residue score of 300 (as determined by equation (2)). A Simulated Annealing algorithm capable of locating solutions closer to the global maximum should therefore discover larger biclusters of the selected quality. It should find larger δ-biclusters. Also consequentially in a sequential search where discovered biclusters are masked biclusters should be discovered in order from largest to smallest.

The Simulated Annealing algorithm was applied to the same yeast dataset as used by Cheng and Church [3]. Cheng and Church used a mean squared residue

threshold of 300 at which to terminate the search. In this study thresholds of 300, 200 and 100 were set and the size of the discovered biclusters compared in each case.

The original Cheng and Church algorithm produces δ-biclusters with varying column and row number. As described in the methods section 4. Simulated Annealing produces significant δ-biclusters no less than 10 columns in size. To align the algorithms and ensure that the column size of the resultant biclusters does not bias the results an Adjusted Node Deletion Algorithm is also run in which the column size is set to 10.

The Simulated Annealing algorithm performed better than the Church and Cheng node deletion algorithm in most cases over several datasets locating larger δ-biclusters of genes and conditions. The results were best for the yeast data shown in figure 1. It can also be seen that the Adjusted Node Deletion algorithm performs better also, demonstrating that larger δ-biclusters can be found over a reduced set of 10 features/columns. If the first bicluster is masked using random values, as in [3]. Simulating Annealing performs similarly in finding a second δ-bicluster 2. Two further datasets Scleroderma and Lymphoma are also analysed, all results are shown in Table 1. Figure 2 compares the results from the Adjusted

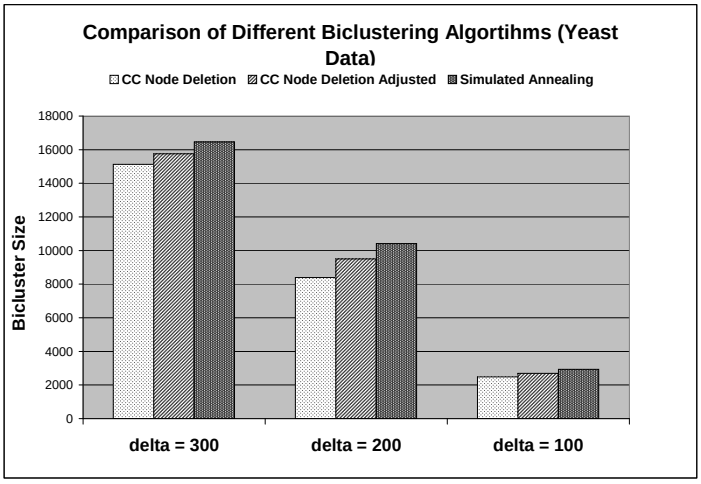

Fig. 1. δ-bicluster comparisons of Cheng and Church's Node Deletion algorithm, Adjusted Node Deletion Algorithm and Simulated Annealing using the yeast dataset over three different correlation scores, X-axis. The numbers on the Y-axis represent the size of a bicluster i.e. its columns multiplied by its rows.

Node Deletion algorithm and the Simulated Annealing algorithm in discovery of the second bicluster. Here we can see that the Simulated Annealing performs significantly better in the yeast dataset. Table 1 demonstrates how the three algorithms performed on the three datasets. In all cases Simulated Annealing out-performs the original Cheng and Church Node Deletion Algorithm. However

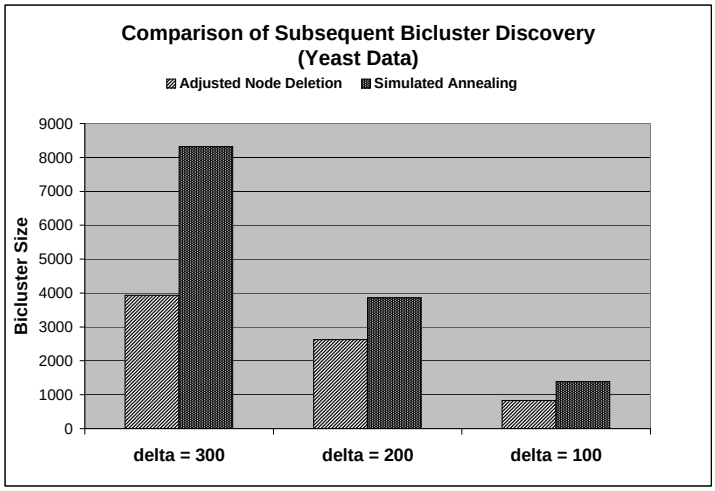

Fig. 2. δ-bicluster comparisons of Cheng and Church's Node Deletion algorithm, Adjusted Node Deltion Algorithm and Simulated Annealing using the yeast dataset over three different correlation scores, X-axis. The numbers on the Y-axis represent the size of a bicluster i.e. its columns multiplied by its rows.

when adjusted to yield biclusters with 10 columns/features the Node Deletion Algorithm performs better. The Simulated Annealing performs better than this adjusted Node Deletion algorithm in 4/9 cases and draws in a further 3. Table 1 also shows that Simulated Annealing performed best in discovering the second biclusters in 6/9 cases over the three datasets.

	Node Deletion	Adjusted Node Deletion	Simulated Annealing	Node Deletion	Adjusted Node Deletion	Simulated Annealing
Yeast	First Bicluster			Second Bicluster		
$\delta = 300$	15165	15750	**16460**	*9012*	3930	**8320**
200	8463	9540	**10360**	*4972*	2630	**3860**
100	2520	2700	**2940**	*1260*	830	**1390**
Scleroderma						
$\delta = 300$	13590	**18260**	**18230**	*4320*	**6780**	6310
200	7296	12920	**13210**	*7876*	3290	**4030**
100	2730	**5170**	**5140**	*1570*	830	**850**
Lymphoma						
$\delta = 300$	1344	**3320**	3220	*518*	1740	**1810**
200	1032	**2510**	2460	*300*	**1370**	1200
100	851	**1780**	**1790**	*136*	**1050**	810

Table 1. First and Second Biclusters of three different /*delta*-scores from each datasetIn the table the best biclusters in each column are marked in bold. Where there are two marked this is considered a draw because of the very small percentage difference in size(less than 0.005 percent). In the Second Bicluster search values in italics are not readily comparable because the first bicluster discovered by Node Deletion is often significantly smaller than that from the other two techniques - leaving more data to work with.

5.2 Biological Interpretation

A further way to evaluate the quality of biclusters is to search for an increased biological significance of the discovered biclusters. Ideally biclusters would represent groups of related genes, some of which may overlap. Therefore in order to evaluate biclusters it would make sense to apply the algorithm to an already annotated and classed set of genes. Of the 2884 genes in the original Cheng and Church yeast dataset 550 have been annotated, that is functions have been assigned based on functional analyses or sequence homology with other organisms. These annotations can be found in an online database called Kyoto Encyclopaedia of Genes and Genomes (http://www.genome.jp/kegg/genes.html). Analyses was carried out on this dataset using the Simulated Annealing Algorithm to assess whether any of the discovered biclusters represented biological functional modules. A δ-score of 100 was chosen as a quality threshold.

The table below represents the first two biclusters discovered from the annotated gene set. It can be seen that a large proportion of bicluster 1 are of the same class of functionally similar genes (ribosomal proteins). The statistical significance of discovering this grouping is given by the 'lift' score [26]. This value measures whether a particular grouping is over-represented or not. If this value is equal to 1 then the significance is no more than would be achieved by a random selection. A value above 1 suggests that there is some positive bias in the selection. The use of the mean squared residue as the objective function

	No. of Genes	Dominant Functional Category	Genes in Functional Category	Lift Value
Bicluster 1	81	Ribosomal Proteins(96)	61	4.31
		Glycolysis/Glucogenisis(26)	5	5.7
Bicluster 2	59	Basal Transcription Factors(10)	6	5.51
		Nucleotide Metabolism(81)	16	1.84

Table 2. The first two biclusters found in the annotated gene set using SA. It can be seen that the first bicluster discovered is rich in genes from the ribosomal functional category. The second bicluster contains transcription factors and genes involved nucleotide metabolism. These genes are the main regulators of protein production and gene expression in the cell.

shows some bias in the type of biclusters it discovers. Because it is a measure of how well an entry fits into an overall dataset and the approaches above begin with the whole dataset - the biclusters found are in some ways distilled versions of the parent dataset. The score promotes simultaneous selection of the most representative features and objects. One can see that if a large bicluster existed in a dataset, in which the rows behaved differently from the general trends in the dataset, that its rows would be the first to be removed in a top-down type search. This bias may also be seen in the contextual evaluation results from the

annotated gene set. Biclusters of the most influential rows and columns are selected. The first bicluster found is rich in ribosomal proteins. The ribosome is the main organelle responsible amino acid and protein production in the cell. These proteins include transcription factors which go on to influence gene expression in the whole cell. So the first bicluster represents a group of the most influential rows. The second bicluster found contains some of these transcription factors plus other proteins involves in transcription such as polymerases and nuclear binding proteins. Thus the first two biclusters discovered represent groups of the most influential genes involved in regulating gene expression. This apparent bias may hinder the discovery of maximal biclusters.

6 Conclusions & Future Work

We have shown that Simulated Annealing perfoms better than Cheng and Church's original Node Deletion algorithm. Although the Adjusted Node Deletion algorithm shows improvments on the original Simulated Annealing still performs better in more cases reviewed. The performance of Simulated Annealing seems to depend somewhat on the dataset in question with its best performance in the yeast dataset. Cheng and Church's *mean square residue score* was conceived with their node deletion algorithm in mind. We have shown that its effectiveness depends to some extent on that algorithm. As a measure of bicluster fitness it is biased towards small biclusters. So when used with a search algorithm such as SA that can avoid local minima, it will almost always produce trivial singleton biclusters. This is much less likely to happen with node deletion as it gets trapped in local minima. Furthermore the score seems also to be biased in the type of biclusters it discovers, finding groups of rows which best match the general trends in the dataset. This is an interesting result and may imply that the score could be applied to feature selection procedures. This bias was also seen in the contextual evaluation with regulatory groups of genes being discovered in the first two biclusters. Perhaps this factor could be harnessed for regulatory gene prediction. Because of this slight bias in the score changes in the parameters of SA which usually increase accuracy, such as a slower annealing schedule will simply strengthen this bias. Afterall any optimisation method is only as good as the fitness function used. As part of our future work in this area, it would be very useful to develop a bicluster fitness measure that is not biased toward small biclusters and in the types of biclusters it discovers.

References

[1] Lander, E.S.: Array of hope. Nat. Genet. **21** (1999) 3–4

[2] Berrer, D., Dubitzky, W., Draghici, S.: 1. In: A Practical Approach to Microarray Data Analysis. Kluwer Academic Publishers (2003) 15–19

[3] Cheng, Y., Church, G.M.: Biclustering of expression data. In: Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB). (2000) 93–103

[4] Madeira, S.C., Oliveira, A.L.: Biclustering algorithms for biological data analysis: A survey. IEEE/ACM Transactions on Computational Biology and Bioinformatics **1** (2004) 24–45
[5] Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by simulated annealing. Science **220** (1983) 671–680
[6] Kaufman, L., Rousseeuw, P.: Finding Groups in Data: An Introduction to Cluster Analysis. John Wiley (1990)
[7] Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D.: Cluster analysis and display of genome-wide expression patterns. Proceedings of the National Academy of Sciences, USA **8** (1998) 14863–8
[8] Tavazoie, S., Hughes, J.D., Campbell, M., Cho, R.J., Church, G.M.: Systematic determination of genetic network architecture. Proceedings of the National Academy of Sciences, USA. **22** (1999) 281–5
[9] Dhillon, I.S., Mallela, S., Modha, D.S.: Information theoretic co-clustering. In: Proceedings of the ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (2003)
[10] Pomeroy, S.L., Tamayo, P., Gaasenbeek, M., Sturla, L.M., Angelo, M., McLaughlin, M.E., Kim, J.Y., Goumnerova, L.C., Black, P., Lau, C., Allen, J.C., Zagzag, D., andT Curran, J.M.O., Wetmore, C., Biegel, J.A., Poggio, T., Mukherjee, S., Rifkin, R., Califano, A., Stolovitzky, D., Louis, D., Mesirov, J., Lander, E., Golub, T.:) prediction of central nervous system embryonal tumour outcome based on gene expression. Nature **24** (2002) 436–42
[11] Ben-Dor, A., Chor, B., Karp, R., Yakini, Z.: Discovering local structure in gene expression data: the order-preserving submatrix problem. Journal of Computational. Biology **10** (2003) 373–84
[12] Mirkin, B.: Mathematical Classification and Clustering. Dordrecht: Kluwer (1996)
[13] Cho, H., Dhillon, I.S., Guan, Y., Sra., S.: Minimum sum squared residue co-clustering of gene expression data. In: SIAM international conference on datamining. (2004)
[14] Hartigan, J.A.: Direct clustering of a data matrix. Journal of the American Stastical Association **67** (1972) 123–129
[15] Lazzeroni, L., Owen, A.: Plaid models for gene expression data. Statistica Sinica **12** (2002) 61–86
[16] Tanay, A., Sharan, R., Shamir, R.: Discovering statistically significant biclusters in gene expression data. Bioinformatics. **18** (2002) 36–44
[17] Kluger, Y., Basri, R., Chang, J.T., Gerstein, M.: Spectral biclustering of microarraydata: Coclustering genes and conditions. Genome Research **13** (2003) 703–716
[18] Johnsen, D.S.: The np-completeness column: an ongoing guide. Journal of Algorithms **8** (1987) 438–448
[19] Yang, J., Wang, H., Wang, W., Yu, P.: Enhanced biclustering on expression. In: IEEE Third Symposium on Bioinformatics and Bioengineering. (2003)
[20] Metropolis, N., Rosenbluth, A.W., Rosenbluth, M.N., Teller, A.H., Teller, E.: Equations of state calculations by fast computing machines. Journal of Chemical Physics **21** (1958) 1087–1092

[21] Binder, K., Stauffer, D.: Applications of the Monte Carlo Method in Statistical Physics. In: A simple introduction to Monte Carlo simulations and some specialized topics. Spring-Verlag, Berlin (1985) 1–36
[22] Preiss, B. In: Data Structures and Algorithms with Object-Oriented Design Patterns in Java. John wiley and Sons (1999)
[23] Whitfield, M.L., Finlay, D.R.., Murray, J.I., Troyanskaya, O., Chi, J.T., Pergamenschikov, A., McCalmont, T., Brown, P.O., Botstein, D., Connolly, M.K.: Systemic and cell type-specific gene expression patterns in scleroderma. Proceedings of the National Acadaemy of Sciences **100** (2003) 12319–12324
[24] Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M., Bloomfield, C., Lander, E.: Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. Science **286** (1999) 531–537
[25] Dudoit, S., Fridlyand, J.: Bagging to improve the accuracy of a clustering procedure. Bioinformatics **19** (2003) 1090–1099
[26] Han, J., Kamber, M.: 6. In: Data Mining: Concepts and Techniques. Morgan Kaufmann Publishers (2000)