A novel approach for avoiding overlapping among biclusters in expression data

Beatriz Pontes
Department of Computer Science
University of Seville
Avda. Reina Mercedes s/n, 41012, Seville, Spain
bepontes@lsi.us.es

Federico Divina
School of Engineering
Pablo de Olavide University
Ctra. Utrera, Km 1, 41013, Seville, Spain
fdivina@upo.es

Raúl Giráldez
School of Engineering
Pablo de Olavide University
Ctra. Utrera, Km 1, 41013, Seville, Spain
giraldez@upo.es

Jesús S. Aguilar–Ruiz
School of Engineering
Pablo de Olavide University
Ctra. Utrera, Km 1, 41013, Seville, Spain
aguilar@upo.es

Abstract

Biclustering is a technique used in analysis of microarray data. It aims at discovering subsets of genes that presents the same tendency under a subsest of experimental conditions. Various techniques have been introduced for discovering significant biclusters. One of the most popular heuristic was introduced by Cheng and Church [6]. In the same work, a measure, called mean squared residue, for estimating the quality of biclusters was proposed. Even if this heuristic is successful in finding interesting biclusters, it presents a number of drawbacks. In this paper we expose these drawbacks and propose some solutions in order to overcome them. Experiments show that the proposed solutions are effective in order to improve the heuristic.

Keywords: *Gene Expression Data, Biclustering, Mean Squared Residue*

1 Introduction

By measuring the expression level of a large number of genes (of the same organisms or of different ones), under different experimental conditions (different environments, individuals, time series, different cells etc.), it is possible to analyze the behavior of the genes. This allows to discover or justify certain biological phenomena.

Microarray techniques allow to quantify and store these expression data in a matrix, whose columns represent genes and rows represent conditions [3, 9]. Such a matrix is called an expression matrix. Therefore, each entry of the matrix denotes the numerical expression level of a gene under a certain experimental condition. With the development of microarray technique, the interest in extracting useful knowledge from gene expression data has experimented an enormous increase. Machine learning techniques have been applied successfully to this context.

Among these techniques, clustering is often used to group genes that behave similarly under all the conditions [4]. However, some genes may be relevant only for a subset of conditions. Traditional clustering cannot be addressed in two dimensions simultaneously. This motivated the development of biclustering algorithms, that was introduced to microarrays analysis by Cheng and Church [6]. Biclustering aims at grouping genes presenting similar trends under a subset of experimental conditions. So a bicluster represent a submatrix of the expression matrix. From biological point of view, biclustering is a very interesting technique, as it is possible to discriminate groups of conditions by using different groups of genes. Biclustering has been proven to be even much more complex than clustering [8].

Two critical factors have influence on the biclusters searching problem: the definition of a measure that assigns a value of quality to the potential biclusters, and the development of a suitable heuristic. The *Mean Squared Residue* [6] (henceforth MSR) is an example of a quality measure for biclusters. MSR has turned into one of the most popular measures to quantify the quality of a bicluster and it has been used by many researches who have proposed different heuristics for biclustering biological data [8, 5, 1, 10]. Cheng and Church also proposed a heuristic for discovering biclusters using MSR. This heuristic is described in the the following sections.

In this work, a particular emphasis is placed on the heuristic proposed in applied by the biclustering algorithm.

978-0-7695-3326-1/08 $25.00 © 2008 IEEE
DOI 10.1109/HIS.2008.81

The exhaustive search of all the biclusters in a microarray is of exponential order regarding the number genes and conditions. Therefore, it is necessary to develop an heuristic which can find good solutions, although these are not the optimal ones. Cheng and Church developed a sequential covering algorithm. Although this approach is one of the main references for many researchers, it has several shortcomings. Such drawbacks are analyzed in the next section and represent the main motivations for this work. Experiments show that the original algorithm proposed in [6] returns biclusters whose MSR is not the one calculated by the algorithm, due to the presence of random values in the expression matrix. Our proposal do not present such a drawback.

This paper is organized as follows. In section 2 we describe the algorithm proposed in [6] and analyze its shortcoming. Section 3 describes our proposal. Experiments and conclusions are described in sections 4 and 5, respectively.

2 Cheng & Church Approach

As it was mentioned before, the original algorithm of Cheng and Church [6] (henceforth Ch&Ch) adopts a sequential covering algorithm in order to return a list of n biclusters from an expression data matrix. The scoring metric to measure the biclusters' quality is MSR, that tries to evaluate the coherence of the genes and conditions of a bicluster B consisting of I rows and J columns. MSR is defined as:

$$\texttt{MSR}(B) = \frac{1}{I \cdot J} \sum_{i=1}^{i=I} \sum_{j=1}^{j=J} (e_{ij} - e_{iJ} - e_{Ij} + e_{IJ})^2 \qquad (1)$$

where e_{ij}, e_{iJ}, e_{Ij} and e_{IJ} represent the element in the i^{th} row (condition) and j^{th} column (gene), the row and column means, and the mean of the whole B, respectively. The smaller the value of MSR, the better the bicluster is considered.

Regarding to the algorithm, a simplified scheme of Ch&Ch is given in Figure 1, where the inputs EM and δ are respectively the expression data matrix and the threshold for the MSR, and L is the list of biclusters that is returned.

```
Input:  Expression Matrix $EM$; Thresholds $\delta$
Output:  List of Biclusters $L$
1  preprocess the missing values of $EM$
2  list $L$ = $\emptyset$
3  Bicluster $B$
4  repeat $n$ times
5     $B$ = $EM$
6     $B_\delta$ = multiple node deletion phase($B$,$\delta$)
7     $B'_\delta$ = simple node deletion phase($B_\delta$,$\delta$)
8     $B''_\delta$ = addition phase($B'_\delta$)
9     $L$ = $L \oplus B''_\delta$
10    substitution phase($B''_\delta$, $EM$)
11 end_repeat
12 return $L$
```

Figure 1. Cheng and Church's original algorithm.

After preprocessing the missing values of EM by replacing with random numbers (line 1) and initializing the list of bicluster to empty (line 2), the bicluster discovering process is repeated n times (lines 5-11). First, the bicluster B is initialized to the whole matrix EM. Next, the multiple node deletion phase (line 6) produces a δ-bicluster B_δ, that is with a MSR value no larger than the preset limit δ. This phase is based on the elimination of those rows or columns whose residue is greater than a certain value, depending on the MSR of the current matrix. Later, the single node deletion phase (line 7) removes the row or column from B_δ with the greater residue and returns B'_δ. Next, the node addition phase (line 8) tries to enlarge the current bicluster B'_δ, adding those columns and rows that do not increase the residue of the matrix above the limit δ and the obtained bicluster B''_δ is stored in the list L (line 9). Finally, the substitution phase (line 10) replaces the elements in the EM that are also in B''_δ with random numbers. The substitution of elements contained by the found biclusters with random values is done in order to prevent overlapping among biclusters, so that, as Cheng and Church states, it very unlikely that elements covered by existing biclusters would contribute to any future bicluster discovery [6].

This strategy succeeds in avoiding the overlapping, however it presents two main drawbacks:

1. As biclusters are discovered, more and more elements of the original expression matrix are lost, since they are substituted with random values. It follows that the expression matrix the algorithm is working on contains more and more random values as biclusters are discovered. As a consequence, the algorithm may return biclusters that are not real, since they contain random values. Moreover, in this way some biclusters might not be found. For instance, if gene j and condition i are contained in a bicluster B, the element e_{ij} is substituted by a random value in the expression matrix. This may prevent gene i to be included in other biclusters under the same condition j, even if it could have improved the quality of the bicluster, since some of its original expression values have been substituted by random values. In general it is desirable to avoid overlapping among biclusters, but not at the cost of loosing possible important interactions among genes.

2. During the execution of the algorithm, the MSR of biclusters considered has to be computed. If a bicluster contains random values its computed MSR is not real, since it is influenced by the presence of random values. This has a negative influence of the overall search process, since the algorithm cannot compute the real values of MSR for some biclusters.

After having performed a number of experiments, we have found that after that a number biclusters has been discovered, the percentage of random values in the expression data matrix is very high.

Another point that has to be considered is that the Ch&Ch make use of a threshold δ in order to reject biclusters: bicluster with MSR higher than δ are rejected. However if some elements of the biclusters are random, the MSR of this biclusters might be higher that δ, and thus rejected. But again the MSR is influenced by the presence of random values. The MSR of the same bicluster with the original elements is different, and could therefore be lower than δ. Also the opposite case may arise, i.e., a bicluster with estimated MSR lower than δ is accepted, but when the original values are used instead of the random ones, the MSR might increase to a level higher than δ. In this last case the bicluster should have been rejected.

$$B = \begin{pmatrix} 53 & 8 & 65 & 84 \\ 122 & 77 & 134 & \mathbf{60} \\ 55 & 10 & 67 & 86 \\ 73 & 28 & 85 & 104 \\ 140 & 95 & 152 & 171 \end{pmatrix}$$

Figure 2. Example of a bicluster containing a random element (showed in bold). The original value of the element was, e.g., 153.

Figure 2 shows an example of a bicluster of such a situation. The bicluster represented in the table has a MSR equal to 259.47. If the element shown in bold were a random value and the original value were 153 the MSR of the bicluster would drop to 0 when the MSR is computed with the original value. If a delta of, e.g., 100 were used, the bicluster depicted in the table would be rejected, even if with the original values it represents a perfect bicluster.

The above considerations clearly show that the replacement strategy adopted by Cheng and Church may prevent the discovering of interesting biclusters, or, on the other hand, yields the algorithm towards the discovering of biclusters considered to be interesting only because of random values they contain. This clearly illustrate the limitations of the replacement policy adopted by Cheng and Church. These considerations represents our main motivations for the work presented in this paper.

3 Our Approach

In this section we describe the variations we incorporated to Ch&Ch, giving rise to a variant algorithm we call Ch&Ch-R. The main variation is represented by the removal of the substitution phase used in the original algorithm (see line 10 in Figure 1). We have also introduced other variations in order to render Ch&Ch not deterministic. This was necessary since in our version of Ch&Ch elements contained in already found biclusters are not replaced by random values. It follows that a deterministic version of Ch&Ch would always find the same bicluster.

3.1 Overlapping Control Mechanism

In our context, two biclusters are overlapped if they contain the same gene (or group of genes) under the same experimental conditions. By controlling the level of overlapping among biclusters, we can decide whether a bicluster may be considered as a significative one, with respect to its overlapping percentage with the previous ones. In our approach, we control the overlapping by means of a matrix of weights W, as in [8]. This matrix has the same dimensions as the original expression data matrix, so that each element $w(e_{ij})$ of W represents a weight associated with e_{ij}. Initially, $w(e_{ij}) = 0, \forall i, j$. Each time a bicluster B is stored in the list L, $w(e_{ij})$ is increased by one if $e_{ij} \in B$.

The matrix of weights W will allow to know how many times each element e_{ij} is contained in the found biclusters. So it can be used to measure the overlapping of a new bicluster as $P(B) = \frac{\sum_{e_{ij} \in B} w(e_{ij})}{V(B)}$, where $V(B)$ is the volume of a bicluster B and $w(e_{ij})$ is the weight for the element $e_{ij} \in B$. $P(B)$ will be high for a bicluster whose elements are already contained in the previous biclusters.

Since we aim at avoiding overlapping as much as possible, $P(B)$ can be used, in combination with MSR, in order to reject biclusters. In order to do this, we need to define a criterion for establishing if $P(B)$ is to be considered high. Bottom and top limits for $P(B)$ are 0 and nb, respectively, where nb is the number of biclusters found.

In order to use $P(B)$ for rejecting biclusters in different iterations, it is convienient that the range of values $P(B)$ can assume is always the same in all iterations. For this purpose in Equation 2 we define the *overlapping factor* of a bicluster for the iteration nb.

$$P_{nb}(B) = \frac{\sum_{i,j \in B} w(e_{ij})}{V(B) \times nb} \qquad (2)$$

Notice that $P_{nb}(B) \in [0, 1], \forall nb$. In this way, we can use it to reject B if $P_{nb}(B)$ is greater than a certain threshold ω. Moreover, notice that, the latest biclusters are allowed to have more elements in common with the already found ones, because $P_{nb}(B)$ tends to be smaller as nb increases. In other words, the biclusters with high overlapping are penalized more in the first iterations.

By setting the overlapping threshold ω, the user can decide the level of overlapping among the solutions. In our experiments, we have used threshold $\omega = 0.5$.

3.2 Re-adaptation of Cheng & Church Algorithm

The main adaptation we propose consists of eliminating the substitution phase of the original algorithm and including the overlapping control mechanism described in the previous section. Notice that since Ch&Ch is deterministic, by removing the substitution phase, the algorithm will produce always the same bicluster. Therefore, the adaptation of the algorithm requires some modifications in order to make the algorithm non-deterministic. Thus, we propose the next variations:

- The substitution phase has been replaced by the overlapping control mechanism, that produces biclusters with overlapping factor smaller than ω and updates the matrix of weights W.
- Multiple node deletion phase has been redefined in terms of the selection of the rows or columns to be deleted from the bicluster. Removing first those rows or columns that produce more overlapping with previous biclusters speeds up the convergence of the algorithm. The selection of the rows and columns is made using the matrix of weights W.
- The selection mechanism for the columns or rows to be added in the node addition phase has been also redefined. Those columns or rows that are less overlapped with previous biclusters are selected first, provided that its addition does not increase the matrix residue above δ. Likewise, this selection is based on the matrix of weights W. The redefinition of the node addition phase aims at finding biclusters with a low overlapping degree.
- Finally, the initial bicluster is randomly determined from the original microarray, with the exception the first iteration where the initial bicluster is the whole matrix as in Ch&Ch.

Adding aforementioned modifications, the pseudocode of our re-adaptation of Ch&Ch, called Ch&Ch−R, is shown in Figure 3.

After preprocessing the missing values of *EM*, the variables *L* (list of biclusters), *nb* (counter for the loop or number of biclusters found), *W* (matrix of weight) and B (initial bicluster) are initialized. Notice that in the first iteration, the bicluster B is initialized to the the whole matrix EM (line 5) in order to take into account the whole set of genes and experimental conditions. Next, the while-loop is executed, where the three first phases (lines 7-9) are the re-adapted multiple node deletion phase, simple node deletion phase and re-adapted addition phase, respectively. These steps always produce δ-biclusters, that is MSR(B_δ), MSR(B'_δ) and MSR(B''_δ) are smaller that δ. Notice that single node deletion (line 8) phase is always deterministic, since the selection of the row or column to be removed depends on their residues. Therefore, there is no adaptation of this phase in our approach.

```
Input:  Expression Matrix $EM$; Thresholds $\delta$ and $\omega$
Output:  List of Biclusters $L$
 1 preprocess the missing values of $EM$
 2 list $L = \emptyset$
 3 $nb$ = 1
 4 Matrix of Weight $W$ = 0
 5 Bicluster $B = EM$
 6 while $nb <= n$ (number of biclusters)
 7   $B_\delta$ = multiple node deletion phase($B,\delta,W$) [re-adapted]
 8   $B'_\delta$ = simple node deletion phase($B_\delta,\delta$)
 9   $B''_\delta$ = addition phase($B'_\delta,W$) [re-adapted]
10   if $P(B''_\delta) <= \omega$
11     $nb = nb + 1$
12     $L = L \oplus B''_\delta$
13     update $W$
14   end_if
15   $B$ = random selection ($EM$)
16 end_while
17 return $L$
```

Figure 3. Ch&Ch−R **Algorithm.**

Once B''_δ is returned by the re-adapted addition phase, the overlapping control method is run. If the overlapping factor of the bicluster *P*(B''_δ) does not exceed the threshold ω, then nb is increased, the bicluster is included within the list and W is updated according to B''_δ as described in section 3.1. Otherwise, if the overlapping factor is above ω, the bicluster B''_δ is rejected, because it has too many common elements with the biclusters previously found.

Finally, bicluster B is randomly generated from the original dataset EM to be used in the next iteration. The dimension of B is also randomly chosen, as well as the specific genes and conditions that make up the bicluster. In the original algorithm, each iteration starts from the whole matrix EM, modified from the last iteration by the substitution phase, the one we have eliminated. However, different experiments showed that starting from a random bicluster produced better results and the convergence is faster.

4 Experiments

In order to test our proposal we conducted experiments with Ch&Ch and Ch&Ch−R on three well known datasets: Yeast *Saccharomyces cerevisiae* cell cycle expression dataset originated from [7] (2884 genes and 17 conditions); Human B–cells expression data originated from [2](4026 genes and 96 conditions); and *Colon Cancer* dataset originated from [7] (2000 genes and 62 conditions).

For the yeast dataset, δ was set to 300, for the human dataset to 1200 and for the colon dataset δ was set to 500. The values of δ used for the yeast and the human dataset are taken from [6], while for the colon dataset the value of δ was set following a procedure suggested in [6].

Table 1. Ch&Ch average results for each dataset. Standard deviation is given between brackets.

Dataset	MSR	MSR(real)	GeneVarMean	Overlap	GenesMean	CondsMean
Yeast	124.80(72.86)	498.46(306.08)	836.36(456.27)	42.94%(36.30)	219.47(309.99)	7.25(3.42)
Human	857.01(107.99)	9940.90(8381.73)	10985.36(8780.13)	49.53%(23.62)	271.52(234.25)	14.70(12.26)
Colon	389.02(76.99)	2159.61(13343.43)	5929.48(16066.57)	9.26%(18.40)	21.89(22.12)	8.81(7.24)

Table 2. Ch&Ch−R average results for each dataset. Standard deviation is given between brackets.

Dataset	MSR	GeneVarMean	Overlap	GenesMean	CondsMean	Overlap2Bics
Yeast	225.138(24.85)	334.02(84.33)	94.55%(12.21)	758.18(212.89)	8.59(2.47)	30.1%(17.58)
Human	1109.94(21.09)	1432.11(101.06)	91.21%(13.45)	134.53(17.34)	45.66(7.41)	25.23%(0.39)
Colon	435.31(13.84)	742.64(13.99)	94.81%(12.15)	134.48(18.15)	24.80(4.44)	33.94%(0.65)

All these datasets were preprocessed as in [6]. The most important preprocessing operation regards missing values: missing values are replaced with random values, although it is known the existing risk that these random numbers can affect the discovery of biclusters [11]. The expectation was that these random values would not form recognizable patterns. For each dataset, we have obtained 100 biclusters, using both the Ch&Ch and Ch&Ch−R.

Tables 1 and 2 show the average results (and their deviations in brackets) obtained on each dataset by the two algorithms. The first column gives the average MSR, the second column in Table 1 shows the mean of the real MSR, i.e., when the MSR is calculated using the original values, and not the random values introduced in the substitution phase. The column *GeneVarMean* shows the gene variance. Next column (*Overlap*) represents the mean of the percentages of overlapping for each biclusters with all the previous ones. Note that in Table 1 this value also represents the mean of the percentage of random values that have been used in the algorithm. This is because values that are contained in more than a bicluster have been substituted with random values. In Table 2, this value only represents the average overlapping with previously found biclusters. This is not equivalent to the amount of overlapping between two given biclusters. Finally, columns *GenesMean* and *CondsMean* show the mean of the number of genes and conditions.

From these tables, it is evident that the random values introduced in the expression matrix during the substitution phase negatively affect Ch&Ch. In fact, the MSR computed considering the original values is, on average, higher than the specified δ. This means that many of the biclusters returned by the algorithm are not δ-biclusters, which is in contradiction with the specification of the algorithm. This fact is particularly evident for the human and the colon datasets, where the average real MSR is about eight and four time higher, respectively, than the δ used for these datasets. Moreover, it can be noticed the huge difference between the MSR computed by Ch&Ch and the real one.

On the other hand, all the biclusters obtained by Ch&Ch−R are δ-biclusters, and the average MSR is much lower than the MSR of the biclusters found by Ch&Ch. This results in itself show the limitations of the substitution phase adopted in Ch&Ch. This substitution phase is effective for avoiding overlapping among biclusters, as it can be noticed by the overlapping percentages shown in Table 1. However, the cost of this substitution phase is producing biclusters that are not δ-biclusters.

As far as the gene variance is concerned, it can be noticed that, in general, Ch&Ch obtained better results. However this result is influenced by the fact that MSR is much higher for the biclusters discovered by Ch&Ch. In general biclusters with lower MSR have also a lower gene variance, and this explain the lower average gene variance for the biclusters obtained by Ch&Ch−R.

Biclusters found by Ch&Ch−R are characterized by a higher volume, even if the average MSR of the biclusters is lower than the MSR of biclusters found by Ch&Ch. This is due to the overlapping policy adopted by Ch&Ch. In fact, the random values introduced causes biclusters found in later iterations of the algorithm to have a very low volume. This is because random values are in general not included in biclusters, since they introduce noise that would cause the genes not be be coherent under some conditions.

Table 2 contains an additional column, named *Overlap2Bics*, with the average percentages of overlapped values between two given biclusters for each dataset. Note that these amounts are considerably lower than the overlapping percentage of the whole set of biclusters (column Overlap).

Finally, for complete our experiments, Figure 4 shows an example of one bicluster for each dataset found by Ch&Ch−R. From a visual inspection of the biclusters, we can notice that the genes contained in the biclusters found on the human and colon dataset fluctuate in unison under the same subset of experimental conditions. The bicluster relative to the yeast dataset contains many genes that present an almost flat behavior. This is reflected by the lower gene variance for this dataset.

One consideration that is worth mentioning is that many of the biclusters found by Ch&Ch−R would not have been discovered by Ch&Ch, due to the overlapping policy it adopts. By introducing random values, there is the risk of masking interesting biclusters, which, in this way, will not be discovered.

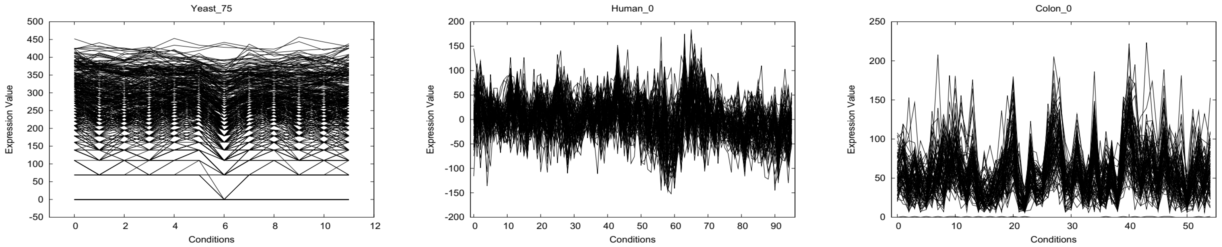

Figure 4. Examples of three biclusters. The leftmost bicluster is relative to the yeast dataset, the bicluster in the center is relative to the human dataset, and the rightmost one to the colon dataset.

5 Conclusions

In this paper we have proposed some variations that can be applied to the Ch&Ch algorithm in order to overcome its shortcomings described in section 2. The original algorithm is very good at discovering biclusters, however, after some iterations it starts to work with more and more random values in the expression matrix, due to the substitution phase used. This causes the algorithm to estimate wrongly the MSR of the biclusters.

Our proposal is based on a matrix of weights, that is used to estimate the overlapping of a bicluster with already found ones. We have defined an overlapping factor which is used in order to reject biclusters if their overlapping is above a certain threshold. In this way the algorithm is always working with the original expression data, and so the biclusters it finds only contains original data. Since no random values are introduced in the expression matrix, we have introduced other modifications to the algorithm in order to render it non deterministic. Results show that many biclusters found by Ch&Ch have a MSR that is higher than δ, due to the random values they contain. This is an important shortcoming of Ch&Ch, since in this way the biclusters it discovers are not δ biclusters. It is also important to notice that many biclusters found by Ch&Ch-R would have not been obtained using the original Ch&Ch. This is due to the fact that Ch&Ch does not work with the original expression matrix. This causes that many biclusters are masked by random values.

As future work, we intend to investigate a way to use the overlapping factor for guiding the algorithm towards biclusters that have little overlap with other ones. This could be done, e.g., by modifying the modification phases of the algorithm, and by using the overlapping factor, in combination with MSR in order to decide, for instance, which node to delete from the bicluster.

Acknowledgement

This research was supported by the Spanish Ministry of Science and Technology under Grant TIN2007-68084-C02-00, and the Junta de Andalucía Research Program.

References

[1] J. S. Aguilar-Ruiz, D. S. Rodriguez, and D. A. Simovici. Biclustering of gene expression data based on local nearness. In *1Proceedings of EGC 2006,*, pages 681–692, Lille, France, 2006.

[2] A. A. Alizadeh, M. B. Eisen, R. E. Davis, and et al. Distinct types of diffuse large b-cell lymphoma identified by gene expression profiling. *Nature*, 403:503–511, 2000.

[3] P. Baldi. *DNA Microarrays and Gene Expression : From Experiments to Data Analysis and Modeling*. Cambridge University Press, 2002.

[4] A. Ben-Dor, R. Shamir, and Z. Yakhini. Clustering gene expression patterns. *Journal of Computational Biology*, 6(3-4):281–297, 1999.

[5] K. Bryan and P. Cunningham. Balboa: Extending bicluster analysis to classify orfs using expression data. In *Proceedings of the 7th IEEE International Conference on Bioinformatics and Bioengineering (BIBE 2007)*, pages 995–1002, 2007.

[6] Y. Cheng and G. M. Church. Biclustering of expression data. In *Proceedings of the 8th International Conference on Intellingent Systemns for Molecular Biology*, pages 93–103, La Jolla, CA, 2000.

[7] R. Cho, M. Campbell, E. Winzeler, and et al. A genome-wide transcriptional analysis of the mitotic cell cycle. *Molecular Cell*, 2:65–73, 1998.

[8] F. Divina and J. Aguilar-Ruiz. Biclustering of expression data with evolutionary computation. *IEEE Transactions on Knowledge & Data Engineering*, 18(5):590–602, 2006.

[9] C. Tilstone. Dna microarrays: Vital statistics. *Nature*, 424:610–612, 2003.

[10] J. Yang, H. Wang, W. Wang, and P. S. Yu. An improved biclustering method for analyzing gene expression profiles. *International Journal on Artificial Intelligence Tools*, 14:771–790, 2005.

[11] J. Yang, W. Wang, H. Wang, and P. S. Yu. δ–clusters: Capturing subspace correlation in a large data set. In *Proceedings of the 18th IEEE Conference on Data Engineering*, pages 517–528, 2002.