BIOINFORMATICS

Vol. 00 no. 00 x
Pages 1–5

A Systematic Comparison and Evaluation of Biclustering Methods for Gene Expression Data

Amela Prelić,[a] Stefan Bleuler[a], Philip Zimmermann[b], Anja Wille[c,d], Peter Bühlmann[d], Wilhelm Gruissem[b], Lars Hennig[b], Lothar Thiele[a], and Eckart Zitzler[a*]

Reverse Engineering Group: [a]Computer Engineering and Networks Laboratory, [b]Institute for Plant Sciences and Functional Genomics Center Zurich, [c]Colab, [d]Seminar for Statistics, Swiss Federal Institute of Technology Zurich, ETH Zentrum, 8092 Zurich, Switzerland.

ABSTRACT

Motivation: In recent years, there have been various efforts to overcome the limitations of standard clustering approaches for the analysis of gene expression data by grouping genes and samples simultaneously. The underlying concept, which is often referred to as biclustering, allows to identify sets of genes sharing compatible expression patterns across subsets of samples, and its usefulness has been demonstrated for different organisms and data sets. Several biclustering methods have been proposed in the literature; however, it is not clear how the different techniques compare to each other with respect to the biological relevance of the clusters as well as to other characteristics such as robustness and sensitivity to noise. Accordingly, no guidelines concerning the choice of the biclustering method are currently available.

Results: First, this paper provides a methodology for comparing and validating biclustering methods that includes a simple binary reference model. Although this model captures the essential features of most biclustering approaches, it is still simple enough to exactly determine all optimal groupings; to this end, we propose a fast divide-and-conquer algorithm (Bimax). Second, we evaluate the performance of five salient biclustering algorithms together with the reference model and a hierarchical clustering method on various synthetic and real data sets for *Saccharomyces cerevisiae* and *Arabidopsis thaliana*. The comparison reveals that (i) biclustering in general has advantages over a conventional hierarchical clustering approach, that (ii) there are considerable performance differences between the tested methods, and that (iii) already the simple reference model delivers relevant patterns within all considered settings.

Availability: The data sets used, the outcomes of the biclustering algorithms under consideration, and the Bimax implementation for the reference model are available at `http://www.tik.ee.ethz.ch/sop/bimax`

Contact: bleuler@tik.ee.ethz.ch, zitzler@tik.ee.ethz.ch

INTRODUCTION

In recent years, several biclustering methods have been suggested to identify local patterns in gene expression data. In contrast to classical clustering techniques such as hierarchical clustering (Sokal and Michener, 1958), and k-means clustering (Hartigan and Wong, 1979), biclustering does not require genes in the same cluster to behave similarly over *all* experimental conditions. Instead, a *bicluster* is defined as a subset of genes that exhibit compatible expression patterns over a subset of conditions. This modified clustering concept can be useful to uncover processes that are active only over some but not all samples as has been demonstrated in several studies (Cheng and Church, 2000; Ihmels *et al*., 2002; Ben-Dor *et al*., 2002; Tanay *et al*., 2002; Murali and Kasif, 2003), see (Madeira and Oliveira, 2004) for a survey.

Comparing clustering methods in general is difficult as the formalization in terms of an optimization problem strongly depends on the scenario under consideration and accordingly varies for different approaches. In the end, biological merit is the main criterion for validation, though it can be intricate to quantify this objective. In the literature, there are several comparative studies on traditional clustering techniques (Yeung *et al*., 2001; Azuaje, 2002; Datta and Datta, 2003); however, for biclustering no such extensive comparisons exist as pointed out by Madeira and Oliveira, 2004. Although first steps in this directions have been made (Tanay *et al*., 2002; Yang *et al*., 2003; Ihmels *et al*., 2004), the corresponding studies focus on validating a new algorithm with regard to one or two existing biclustering methods, and therefore general guidelines are difficult to derive.

The main goal of this paper is to provide a systematic comparison and evaluation of prominent biclustering methods. In particular, we address the following questions:

- What comparison and validation methodology is adequate for the biclustering context?
- How meaningful are the biclusters selected by existing methods?
- How do different methods compare to each other: do some techniques have advantages over others or are there common properties that all approaches share?

In order to answer these questions, we have selected a number of salient biclustering methods, implemented them, and tested them on both synthetic and real gene expression data sets. An *in silico* scenario has been chosen to (i) investigate the capability of the algorithms to recover implanted *transcription modules* (Ihmels *et al*., 2002), i.e., sets of co-regulated genes together with their regulating conditions, and to (ii) study the influence of regulatory complexity and noise on the performance of the algorithms. To

*to whom correspondence should be addressed

assess the biological relevance of biclusters on gene expression data for *Saccharomyces cerevisiae* and *Arabidopsis thaliana*, multiple quantitative measures are introduced that relate the biclustering outcomes to annotations by The Gene Ontology Consortium, 2000, metabolic pathway maps, and protein-interaction data.

Moreover, we propose a simple biclustering model, which retains common features of most biclustering methods, in combination with a fast and exact algorithm (Bimax)—in contrast, existing biclustering algorithms usually do not guarantee to find global optima. Although restricted from a biological point of view, this model allows to study the validity of the biclustering idea independent of the interfering effects due to approximate algorithms. As such, Bimax has been considered as a reference method in our study. As will be shown in the remainder of this paper, even such a simple approach delivers biologically relevant results and compares well with more sophisticated biclustering methods.

BICLUSTERING METHODS

Selected Algorithms

Five prominent biclustering methods have been chosen for this comparative study according to three criteria: (i) to what extent the methods have been used or referenced in the community, (ii) whether their algorithmic strategies are similar and therefore better comparable, and (iii) whether an implementation was available or could be easily reconstructed based on the original publications. The selected algorithms, which all are based on greedy search strategies, are: Cheng and Church's algorithm, *CC*, (Cheng and Church, 2000); *Samba*, (Tanay *et al.*, 2002); Order Preserving Submatrix Algorithm, *OPSM*, (Ben-Dor *et al.*, 2002); Iterative Signature Algorithm, *ISA*, (Ihmels *et al.*, 2002, 2004); *xMotif*, (Murali and Kasif, 2003). A brief description of the corresponding approaches can be found in the supplementary material.

Reference Method (Bimax)

The above methods use different models which are all too complex to be solved exactly; most of the corresponding optimization problems have shown to be NP-hard. Therefore, advantages of one method over another can be due to a more appropriate optimization criterion or a better algorithm.

To decouple these two aspects, we propose the following reference model that reflects the fundamental idea of biclustering, while allowing to determine all optimal biclusters in reasonable time. This model has the benefit of providing a basis to investigate

1. The usefulness of the biclustering concept in general, independently of interferring effects caused by approximate algorithms; and
2. The effectiveness of more complex scoring schemes and biclustering methods in comparison to a plain approach.

Model The model assumes two possible expression levels per gene: no change and change with respect to a control experiment.[1] Accordingly, a set of m microarray experiments for n genes can be represented by a binary matrix $E^{n\times m}$, where a cell e_{ij} is 1 whenever gene i responds in the condition j and otherwise it is 0. A *bicluster* (G, C) corresponds to a subset of genes $G \subseteq \{1, .., n\}$ that jointly respond across a subset of samples $C \subseteq \{1, .., m\}$. In other words, the pair (G, C) defines a submatrix of E for which all elements equal 1. Note that, by definition, every cell e_{ij} having value 1 represents a bicluster by itself. However, such a pattern is not interesting per se; instead, we would like to find all biclusters that are *inclusion-maximal*, i.e., that are not entirely contained in any other bicluster.

[1] To this end, a preprocessing step normalizes log expression values and then transforms matrix cells into discrete values. To obtain binary values, a commonly applied discretization procedure based on a cutoff threshold representing a twofold change is employed.

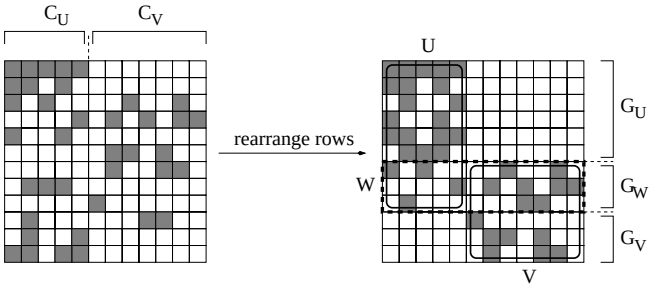

Fig. 1: Illustration of the Bimax algorithm. To partition the input matrix into four non-overlapping parts, first the set of columns is divided into two subsets C_U and C_V, here by taking the first row as a template. Afterwards, the rows of E are resorted: first come all genes that respond only in conditions given by C_U, then those genes that respond to conditions in C_U and in C_V, and finally the genes that respond to conditions in C_V only. The corresponding sets of genes G_U, G_W, and G_V then define in combination with C_U and C_V the resulting submatrices U, V, and W which are decomposed recursively.

DEFINITION 1. *The pair* $(G, C) \in 2^{\{1,..,n\}} \times 2^{\{1,..,m\}}$ *is called an* inclusion-maximal bicluster *if and only if (1)* $\forall\, i \in G, j \in C : e_{ij} = 1$ *and (2)* $\nexists (G', C') \in 2^{\{1,..,n\}} \times 2^{\{1,..,m\}}$ *with (i)* $\forall\, i' \in G', j' \in C' : e_{i'j'} = 1$ *and (ii)* $G \subseteq G' \wedge C \subseteq C' \wedge (G', C') \neq (G, C)$.

This model is similar to the one presented by Tanay *et al.*, 2002 who consider a more realistic definition of optimality where a bicluster can also contain 0-cells.

Algorithm Since the size of the search space is exponential in n and m, an enumerative approach is infeasible in order to determine the set of inclusion-maximal biclusters. Alexe *et al.*, 2002 proposed an algorithm in a graph-theoretic framework that can be employed in this context, if the matrix E is regarded as an adjacency matrix of a graph. By exploiting the fact that the graph induced by E is bipartite, their incremental algorithm can be tailored to this application which reduces the running-time complexity from $\Theta(n^2\, m^2 \beta)$ to $\Theta(nm^2\beta)$, where β is the number of all inclusion-maximal biclusters in $E^{n\times m}$ with $m \leq n$ (see supplementary material).

In this paper, though, we propose and use a faster divide-and-conquer approach, the binary inclusion-maximal biclustering algorithm (Bimax) that provides a running-time complexity of $O(nm\beta)$—the complete algorithm and the proof of the upper bound are given in the appendix. It tries to identify areas of E that contain only 0s and therefore can be excluded from further inspection. This strategy is especially beneficial for our purposes as E is, depending on the cutoff threshold, sparse; in all data sets used in this study, the proportion of 1-cells over 0-cells never exceeded 6% when considering a twofold change cutoff.

More specifically, the idea behind the Bimax algorithm, which is illustrated in Fig. 1, is to partition E into four non-overlapping submatrices, two of which contain only 0-cells and therefore can be disregarded in the following. The algorithm is then recursively applied to the remaining two submatrices U and V; the recursion ends if the current matrix represents a bicluster, i.e., contains only 1s. If U and V do not share any rows and columns of E, the output is the union of the corresponding inclusion-maximal biclusters. However, if U and V have a set G_W of rows in common as shown in Fig. 1, there are biclusters that extend over U and V. To determine these, we also run the algorithm for the submatrix W that is given by G_W; special care is necessary, though, to only generate those biclusters in W that share at least one common column with U as well as one with V.

Limitations Theoretically, the number of inclusion-maximal biclusters can be expontential in n and m and therefore generating the entire set can become infeasible. For real data, though, the actual number lies within reasonable bounds as the number of 1-cells is small. For instance, for a 6000×50-matrix with a density of 5%, around 6500 biclusters are

returned by the algorithm, while the theoretical bound is $1.13e^{+15}$. The running time for such a matrix is around 3 seconds on a 3 GHz Intel Xeon machine, and about 35 minutes for corresponding 6000×450-matrices. The supplementary material provides more details about these issues.

Furthermore, a secondary filtering procedure, similarly to other biclustering approaches such as (Tanay *et al.*, 2002; Ihmels *et al.*, 2004), can be applied to reduce the number of biclusters to the desired size; this issue will be discussed in the next section. Another possibility is to constrain the size of the biclusters during the search process. The advantage of the Bimax algorithm over the incremental procedure is that such size constraints can be naturally integrated—thereby, considerable speed-ups are achievable.

COMPARISON METHODOLOGY

There exist several studies that address the issue of comparing and validating one-dimensional clustering methods (Datta and Datta, 2003; Gat-Viks *et al.*, 2003; Kerr and Churchill, 2001; Yeung *et al.*, 2001; Azuaje, 2002). All of them make use of different quantitative measure or *validity indices*, which can be divided into three categories (Halkidi *et al.*, 2001): internal, external, and relative indices. Internal indices solely rely on the input data—examples are the two measures of *homogeneity* and *separation* (Gat-Viks *et al.*, 2003). However, such indices are difficult to devise in the biclustering context as existing approaches adopt different definitions of what an optimal bicluster is: constantly up-regulated, just constant or showing coherent or complementary trends, etc. In contrast, external criteria are based on additional data in order to validate the obtained results. In the context of gene expression data, these would correspond to prior biological knowledge of the systems being studied; alternatively, a validation can be done by referring to other types of genomic data representing similar aspects of the regulation mechanisms being investigated. The third category of relative indices measures the influence of the input parameter settings on the clustering outcome.

Here, external indices are used to assess the biclustering methods under consideration as the focus is on the biological relevance of the biclusters found. On the one hand side, we investigate the performance of the algorithms on data sets that have been generated on the basis of an artificial model. Although such a controlled setting inherently only reflects certain aspects of biological reality, it has two advantages over real data: (i) the optimal solutions are known beforehand, and (ii) it can be arbitrarily scaled to different levels of complexity. On the other hand side, we assess the relevance of biclusters by using further biological data, namely GO annotations, metabolic pathways maps, and protein-protein interactions.

Note that, similarly to other studies (Tanay *et al.*, 2002; Ihmels *et al.*, 2002), biclusters are evaluated only with respect to the gene dimension. One reason is the lack of data for the condition dimension, i.e., annotations for samples are usually not available. Secondly, an important question that the present study addresses is how the biclustering methods compare to classical clustering approaches. Since the latter only operate on genes, conditions cannot be regarded in such a comparison.

Validation Using Synthetic Data

Data Sets The artificial model used to generate synthetic gene expression data is similar to an approach proposed by Ihmels *et al.*, 2002. In this setting, biclusters represent *transcription modules*; these modules are defined by (i) a set G of genes regulated by a set of common transcription factors, and (ii) a set C of conditions in which these transcription factors are active. More specifically, we consider

- A set of t transcription factors;
- A binary activation matrix $A^{t \times m}$ where $a_{ij} = 1$ iff transcription factor i is active in condition j;
- A binary regulation matrix $R^{t \times n}$ where $r_{ij} = 1$ iff transcription factor i regulates gene j;

and compute the corresponding gene expression matrix E by setting the expression value e_{ij} of gene i at condition j to $e_{ij} = \max_{1 \le k \le t} r_{ki} \cdot a_{kj}$.

Based on this model, two scenarios have been considered. In the first scenario, 10 non-overlapping transcription modules, extending over 10 genes and 5 conditions, emerge. Each gene is regulated by exactly one transcription factor and in each condition only one transcription factor is active. The corresponding data set has been used to study the effects of different types of noise on the performance of the biclustering methods. For the second scenario, the regulatory complexity has been systematically varied: here, each gene can be regulated by d transcription factors and in each condition up to d transcription factors can be active. As a consequence, some of the resulting biclusters overlap and the number of transcription modules increases to up to 28. The parameter d is an indicator for the overlap degree, and nine different data sets have been generated for $d = 0, 1, \ldots, 8$. [2]

Match Scores In order to assess the performance of the selected biclustering approaches, we will use a score that describes the degree of similarity between the computed biclusters and the transcription modules implanted in the synthetic data sets.

The following score is designed to compare two biclusters.

DEFINITION 2. *Let $G_1, G_2 \subseteq \{1, \ldots, n\}$ be two sets of genes. The* match score *of G_1 and G_2 is given by the function*

$$S_G(G_1, G_2) = \frac{|G_1 \cap G_2|}{|G_1 \cup G_2|}$$

which characterizes the correspondence between the two gene sets.

This match score, which resembles the *Jaccard coefficient*, cf. (Halkidi *et al.*, 2001), is symmetric, i.e., $S_G(G_1, G_2) = S_G(G_2, G_1)$, and its value ranges from 0 (the two sets are disjoint) to 1 (the two sets are identical). A match score S_C for sample sets can defined by analogy.

On this basis, a score for comparing two sets of biclusters can be introduced as follows.

DEFINITION 3. *Let M_1, M_2 be two sets of biclusters. The* gene match score *of M_1 with respect to M_2 is given by the function*

$$S^*_G(M_1, M_2) = \frac{\sum_{(G_1, C_1) \in M_1} \max_{(G_2, C_2) \in M_2} S_G(G_1, G_2)}{|M_1|}$$

which reflects the average of the maximum match scores for all biclusters in M_1 with respect to the biclusters in M_2.

The gene match score is not symmetric and usually yields different values when M_1 and M_2 are exchanged; accordingly, both $S^*_G(M_1, M_2)$ and $S^*_G(M_2, M_1)$ need to be considered. Although, this comparative study takes only the gene dimension into account, an overall match score can be defined as $S^*(M_1, M_2) = \sqrt{S^*_G(M_1, M_2) \cdot S^*_C(M_1, M_2)}$ where S^*_C is the corresponding *condition match score*.

Now, let M_{opt} denote the set of implanted biclusters and M the output of a biclustering method. The *average bicluster relevance* is defined as $S^*_G(M, M_{\text{opt}})$ and reflects to what extent the generated biclusters represent true biclusters in the gene dimension. In contrast, the *average module recovery*, given by $S^*_G(M_{\text{opt}}, M)$, quantifies how well each of the true biclusters is recovered by the biclustering algorithm under consideration. Both scores take the maximum value of 1, if $M_{\text{opt}} = M$.

[2] In detail, activation and regulation matrices were created as follows:

$$r_{ij} = \begin{cases} 1 & \text{if } (i-1)n'/t + 1 \le j \le in'/t + d \\ 0 & \text{else} \end{cases}$$

for $1 \le i \le t$, $1 \le j \le n' + d$, and

$$a_{ij} = \begin{cases} 1 & \text{if } (i-1)m'/t + 1 \le j \le im'/t + d \\ 0 & \text{else} \end{cases}$$

for $1 \le i \le t$, $1 \le j \le m' + d$. For scenario 1, the parameters were $n' = 100, m' = 50, t = 10$, and $d = 0$. For scenario 2, the parameter setting was $n' = 100, m' = 100, t = 10$ in combination with different overlap degrees $d \in \{0, \ldots, 8\}$.

Validation Using Prior Knowledge

Prior biological knowledge in the form of natural language descriptions of functions and processes that genes are related to has become widely available. One of the largest organized collection of gene annotations is currently provided by The Gene Ontology Consortium, 2000. Similarly to the idea pursued in (Tanay *et al*., 2002), we here investigate whether the groups of genes delivered by the different algorithms show significant enrichment with respect to a specific Gene Ontology (GO) annotation. In detail, biclusters are evaluated by computing the hypergeometric functional enrichment score, cf. (Berriz *et al*., 2003), based on Molecular Function and Biological Process annotations; the resulting scores are adjusted for multiple testing by using the Westfall and Young procedure (Westfall and Young, 1993; Berriz *et al*., 2003). This analysis is performed for the model organism *Saccharomyces cerevisiae*, since the yeast GO annotations are more extensive compared to other organisms. The gene expression data set used is the one provided by Gasch *et al*., 2000, which contains a collection of 173 different stress conditions and a selection of 2993 genes.

In addition to GO annotations, we consider specific biological networks, namely metabolic and protein-protein interaction networks, that have been derived from other types of data than gene expression data. Although each type of data reveals other aspects of the underlying biological system, one can expect to a certain degree that genes that participate in the same pathway respectively form a protein complex also show similar expression patterns as discussed in (Zien *et al*., 2000; Ideker *et al*., 2002). The question here is whether the computed biclusters reflect this correspondence.

To this end, we model both pathway information as well as protein interactions in terms of an undirected graph where a node stands for a protein and an edge represents a common reaction in that the two connected proteins participate respectively a measured interaction between the two connected proteins. In order to verify whether a given bicluster (G, C) is plausible with respect to the metabolic respectively protein interaction graph, we consider two scores: (i) the proportion of pairs of genes in G for which there exists no connecting path in the graph, and (ii) the average path length of pairs of genes in G for which such a path exists. One may expect that both the number of disconnected gene pairs and the average distance between two connected genes is significantly smaller for genes in G than for randomly chosen genes. For both scores, a resampling method is applied where 1000 random gene groups of the same size as G are considered; a Z-test is used to check whether the scores for the bicluster (G, C) are significantly smaller or larger than the average score for the random gene groups.

As to the metabolic level, we use a pathway map that describes the main bio-synthetic pathways at the level of enzymatic reactions for the model organism *Arabidopsis thaliana* (Wille *et al*., 2004). As this map has been manually assembled at the Institute for Plant Science at ETH Zurich by an extensive literature search, the resulting graph represents a high level of confidence. The gene expression data set used in this context are publicly available at `http://nasc.nott.ac.uk/` and comprise 69 experimental conditions and a selection of 734 genes.

To investigate the correspondence of biclusters and protein-protein interaction networks, again *Saccharomyces cerevisiae* is considered because the amount of interaction data available is substantially larger than for *Arabidopsis thaliana*. Here, we combine the aforementioned gene expression data set for yeast (Gasch *et al*., 2000) with corresponding protein interactions stored in the DIP database (Salwinski *et al*., 2004), resulting in 11498 interactions for 3665 genes overall.

Implementation Issues

All of the selected methods have been re-implemented according to the specifications in the corresponding papers, except of Samba for which a publicly available software tool, Expander (Sharan *et al*., 2003), has been used. The OPSM algorithm has been slightly extended to return not only a single bicluster but the q largest biclusters among those that achieve the optimal score; q has been set to 100. Furthermore, the standard hierarchical clustering method (HCL) in MATLAB has been included in the comparison, which uses single linkage in combination with Euclidean distance. The parameter settings for the various algorithms correspond to the values that the authors have recommended in their publications (supplementary material).

As the number of generated biclusters varies strongly among the considered methods, a filtering procedure, similarly to (Tanay *et al*., 2002; Ihmels *et al*., 2002), has been applied to the output of the algorithms to provide a common basis for the comparison. The filtering procedure adopted here follows a greedy approach: in each step, the largest of the remaining biclusters is chosen that has less than o percent of its cells in common with any previously selected bicluster; the algorithm stops if either q biclusters have been selected or none of the remaining ones fulfills the selection criterion. For the synthetic data sets, q equals the number of optimal biclusters, which is known beforehand, and for the real data sets, q is set to 100; in both cases, a maximum overlap of $o = 0.25$ is considered.

As to the input data, the gene expression matrices have been normalized using mean centering.

RESULTS

Synthetic Data

The data derived from the aforementioned artificial model enables us to investigate the capability of the methods to recover known groupings, while at the same time further aspects like noise and regulatory complexity can be systematically studied. The data sets used in this context are kept small, i.e., $n = 100, m = 50$ for scenario 1 and $n = m = 100$ for scenario 2, in order to allow a large number of numerical experiments to be performed—for a 100×100-matrix, the running-times of the selected algorithms varied between 1 and 120 seconds. The size of the considered data sets, though, does not restrict the generality of the results presented in the following, since the inherent structure of the data matrix, i.e., the overlap degree, is the main focus of our study.

Effects of Noise The first artificial scenario, where all biclusters are non-overlapping, serves as a basis to assess the sensitivity of the methods to noise in the data. It is to be expected that hierarchical clustering works well in such a scenario as the implanted gene groups are clearly separated in the condition dimension.

Two types of noise are considered: (i) measurement noise, and (ii) noise caused by discretization errors. The former type stands for variations related to the underlying experimental technologies and the stochasticity of the biological systems under consideration. Measurement noise is imitated by adding random values drawn from a normal distribution to each cell of the original gene expression matrix. The second type of noise is a result of data preprocessing; some of the considered methods, e.g., Samba, ISA, and Bimax, discretize the data which may lead to quite different matrices depending on the chosen discretization threshold. Discretization errors are simulated by flipping a certain portion of the cells from 0 to 1 and vice versa—according to a change of the discretization threshold. For both types of noise, the noise level, i.e., the standard deviation σ respectively the proportion of flipped cells, is systematically increased. For each noise level, 10 different data matrices have been generated from the original gene expression matrix E, and the performance of each algorithm is averaged over these 10 input matrices.

Fig. 2(a) and 2(b) summarize the performances of the considered methods with respect to measurement noise, while Fig. 2(c) and 2(d) depict the results for noise caused by discretization errors.

In the absence of noise, ISA, Samba, and Bimax are able to completely identify all implanted transcription modules; as

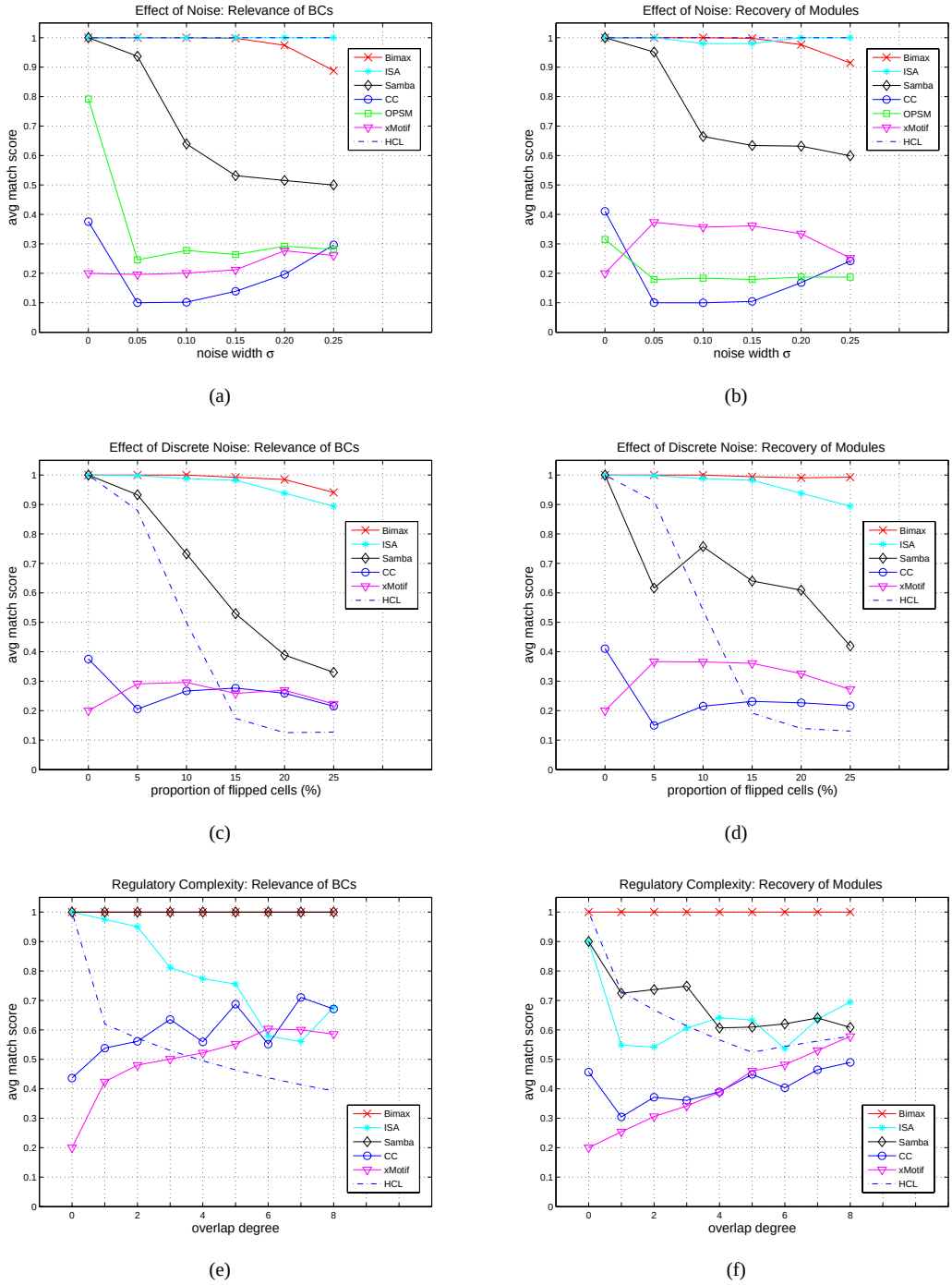

Fig. 2: Results for the artificial scenarios: (a), (b) non-overlapping modules with increasing measurement noise; (c), (d) non-overlapping modules with increasing discrete noise; (e), (f) overlapping modules with increasing overlap degree and no noise.

expected, the same holds for the hierarchical clustering approach, if the number k of clusters to be generated corresponds to the actual number of implanted modules. In contrast, the scores obtained by CC and xMotif are substantially lower. The reason is that the largest biclusters found by these two methods mainly contain 0-cells, i.e., the algorithms do not focus on changes in gene expression, but consider the similarity of the selected cells as the only clustering criterion. This problem has been discussed in detail in (Cheng and Church (2000)). Finally, OPSM represents a special case in this comparison, because it seeks clear trends of up- or down-regulation and to this end transforms expression values into discrete ranks. As a result of this transformation, submatrices with quasi-constant expression values that corresponds to modules in our artificial scenario are hard to find with this approach. The high average bicluster relevance is rather an artefact of the implementation used in this paper which keeps the order of the columns when

identical expression values are present. However, as soon as noise is added, this artificial order is destroyed, which in turn leads to considerably lower gene match scores, cf. Fig. 2(a) and 2(b). For this reason, OPSM is not further considered in the next two *in silico* experiments.

Concerning the influence of noise, both ISA and Bimax are only marginally affected by either type of noise and still recover more than 90% of all implanted modules even for high noise levels. With HCL, measurement noise has no observable effects, while the second type of noise leads to a drastical drop in performance. The latter observation can be explained by the fact that the expression patterns of the genes within a transcription module, considered across the available samples, become widely different, if the input matrix is disturbed with random bit flips. Even if other distance metrics and linkage options are used for HCL, the results looks similar to the ones depicted in Fig. 2(c) and 2(d). Samba seems to be sensitive to both types of noise as the average gene match scores decrease by 40% to 50% for a medium noise level; still, the scores are significantly larger than for CC and xMotif. Remarkably, the performance of CC appears to improve with increasing noise. This phenomenon, though, is again a result of the adopted algorithmic strategy, cf. (Cheng and Church, 2000): the largest biclusters may mainly cover the background, i.e., 0-cells. With noise, the biclusters found in the matrix background tend to be smaller, and this results in an improved gene match score; further evidence is provided in the supplementary material.

Regulatory Complexity The focus of the second artificial scenario is to study the behavior of the chosen algorithms with respect to increased regulatory complexity. Here, a single gene may be activated by a *set* of transcription factors, and accordingly the implanted transcription modules may overlap. This setting is expected to reveal the advantages of the biclustering approach over traditional clustering methods such as hierarchical clustering.

Fig. 2(e) and 2(f) depict the results for different overlap degrees, cf. the description of the data sets on Page 3, and in the absence of noise. The only method that fully recovers all hidden modules in the data matrix is—by design—the reference method, Bimax. Among the remaining methods, Samba provides the best performance: all biclusters found represent hidden modules; however, not all implanted modules are recovered. In comparison, ISA appears to be more sensitive to increased regulatory complexity: the average module recovery score is similar to the one of Samba, but the average bicluster relevance drops to 60% with the largest considered overlap degree of 8. This may be explained by the normalization step in the first preprocessing step of the algorithm. With increasing overlap, the ratio of 0- to 1-cells per row decreases and the expression value range after normalization becomes narrower. As a result, the differences between unchanged and up- or down-regulated expression values blurr and are more difficult to separate based on the gene and chip threshold parameters t_g, t_c. These parameters have a strong impact on the performance as shown in the supplementary material. As to CC and XMotif, both gene match scores increase with larger overlaps degrees, but are still lower than the ones by Bimax, Samba, and ISA; again, this is due to the fact that the number of 0-cells decreases with larger overlaps. Comparing the biclustering methods with HCL, one can observe that already a minimal overlap causes a large decrease in the performance of HCL—even if the optimal number of clusters is used. The reason is that clusters obtained by HCL form a partition of genes, i.e., are non-overlapping, and this implies that not every planted transcription module can be possibly recovered.

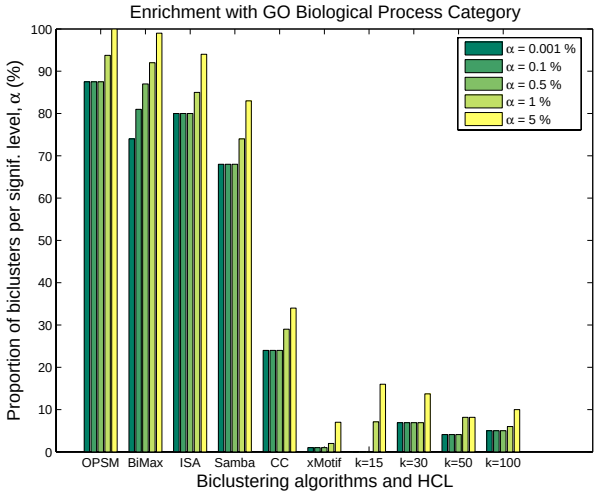

Fig. 3: Proportion of biclusters significantly enriched by any GO Biological Process category (*Saccharomyces cerevisiae*) for the six selected biclustering methods as well as for hierarchical clustering with $k \in \{15, 30, 50, 100\}$. The columns are grouped method-wise, and different bars within a group represent the results obtained for five different significance levels α.

Real Data

Any artificial scenario inevitably is biased regarding the underlying model and only reflects certain aspects of biological reality. For instance, the assumption that a transcription module exhibits a quasi-constant expression level favors some algorithms, and therefore OPSM needed to be excluded from the comparison in the previous section. In the following, the algorithms are tested on real data sets and the biological relevance of the obtained biclusters is evaluated with respect to GO annotations, metabolic pathway maps, and protein-protein interaction data.

Functional Enrichment The histogram in Fig. 3 reflects for each method the proportion of biclusters for which one or several GO categories are overrepresented—at different levels of significance. Best results are obtained by OPSM. Given that this approach only returns a small number of biclusters, here 12 in comparison to 100 with the other methods, it delivers gene groups that are highly enriched with the GO Biological Process category. This result is insofar interesting as the effect of the noise observed in the artificial setting does not seem to be a problem with the considered real data set. Bimax, ISA, and Samba also provide a high portion of funtionally enriched biclusters, with a slight advantage of Bimax and ISA (over 90% at a significance level of 5%) over Samba (over 80% at a significance level of 5%). In contrast, the scores for CC are considerably lower (around 30%) due to the same problem as discussed in the previous section. Cheng and Church, 2000 mention that the first few biclusters should probably be discarded, but the practical issue remains that it is not clear which biclusters are meaningful and should be considered for further analysis.

Except for xMotif, though, all biclustering methods achieve higher scores than HCL with different values for k, the number of clusters to be sought. This can be explained in terms of the data

Table 1. Biological relevance of biclusters with respect to a metabolic pathway map (MPM) for *Arabidopsis thaliana* and a protein-protein interaction network (PPI) for *Saccharomyces cerevisiae*. For each bicluster, a Z-test is carried out to check whether its score is significantly smaller or greater than the expected value for random gene groups; the table gives for each method the proportion of biclusters with statistically significant scores (significance level $\alpha = 10^{-3}$). The results for HCL are omitted as all scores equal 0%.

Method	proportion of disconnected gene pairs				average shortest distance in the graph			
	smaller		greater		smaller		greater	
	MPM	PPI	MPM	PPI	MPM	PPI	MPM	PPI
Bimax	58.9	14.0	19.5	64.0	85.3	58.0	3.4	16.0
CC	70.0	52.0	15.0	26.0	70.0	42.0	15.0	34.0
OPSM	42.8	18.8	28.6	50.0	92.9	56.3	0.0	43.8
Samba	41.6	0.0	37.5	100.0	75.6	25.6	13.1	46.2
xMotifs	49.0	2.0	17.0	92.0	84.0	4.0	3.0	72.0
ISA	25.0	58.0	25.0	22.0	50.0	70.0	25.0	22.0

set used: Since it refers to different types of stresses, it is likely that local, stress-dependent expression patterns emerge that are hard to find by traditional clustering techniques. This hypothesis is also supported by the fact that most functionally enriched biclusters only contain one or two overrepresented GO categories and that there is no clear tendency towards any of the categories.

Comparison to Metabolic and Protein Networks Under the assumption that the structure of a metabolic pathway map respectively a protein-protein interaction network is somehow reflected in the gene expression data, the degree of connectedness of the genes associated with a bicluster can be used to assess its biological relevance. In particular, one may expect that both the number of disconnected gene pairs and the average shortest distance between connected gene pairs tend to be smaller for the biclusters found than for random gene groups.

Table 1 shows that this holds for the data set and the metabolic pathway map used for *Arabidopsis thaliana*. If there exists a path between two genes of a bicluster in the metabolic graph, then with high probability the distance between these genes is significantly smaller than the average shortest distance between randomly chosen gene pairs. Although for most methods, the biclusters are better connected than random gene groups, the differences to the random case are not as striking as for the average gene pair distance. This indicates that combining gene expression data with pathway maps within a biclustering framework can be useful to focus on specific gene groups. Note that also hierarchical clustering with $k \in \{15, 30, 50, 100\}$ has been applied to these expression data; however, a single cluster usually contains almost all the genes of the data set, while the remaining clusters comprise only few genes. Accordingly, no significant differences to random clusters can be observed.

The results for the corresponding comparison for the protein-protein interaction, though, are ambiguous, cf. Table 1. As to the degree of disconnectedness, there is no clear tendency in the data which can be attributed to the fact that not all possible protein pairs have been tested for interaction. Focusing on connected gene pairs only, ISA and Bimax seem to mostly generate gene groups that have a low average distance within the protein network in comparison to random gene sets; for xMotif, the numbers suggest the opposite. Overall, the differences between the biclustering methods demonstrate that special care is necessary when integrating gene expression and protein interaction data: not only the incompleteness of the data needs to be taken into consideration, but also the confidence in the measurements has to be accounted for, see, e.g., Gilchrist *et al*. (2004).

CONCLUSIONS

The present study compares five prominent biclusterings methods on the basis of both synthetic and real gene expression data sets; hierarchical clustering and a baseline biclustering algorithm, Bimax, proposed in this paper serve as a reference. The key results are:

- In general, the biclustering concept allows to identify groups of genes that cannot be found by a classical clustering approach that always operates on *all* experimental conditions. On the one hand side, this is supported by the observation that with increased regulatory complexity the ability of hierarchical clustering to recover the implanted transcription modules in an artificial scenario decreases substantially. On the other hand side, on real data the groups outputted by hierarchical clustering for different similarity measures and parameters do not exhibit any siginificant enrichment according to GO annotations and metabolic pathway information. In contrast, most biclustering methods under consideration are capable of dealing with overlapping transcription modules and generate functionally enriched clusters. Furthermore, the biclustering algorithms appear to be more robust regarding high noise levels.
- There are significant performance differences among the five biclustering methods. On the real data sets, ISA, Samba, and OPSM provide similarly good results: a large portion of the resulting biclusters is functionally enriched and indicates a strong correspondence with known pathways. In the context of the synthetic scenarios, Samba is slightly more robust regarding increased regulatory complexity, but also more sensitive regarding noise than ISA. As to OPSM, the outcomes can change considerably with the noise level as the comparison on the artificial data reveals; however, as it uses a rank-based notion of bicluster homogeneity, this observation is mainly due to the chosen synthetic scenario and must not be generalized. The remaining two algorithms, CC and xMotif, both tend to generate large biclusters that often represent gene groups with unchanged expression levels and therefore not necessarily contain interesting patterns in terms of, e.g., co-regulation. Accordingly, the scores for CC and xMotif are significantly lower than for the other biclustering methods under consideration.
- The Bimax baseline algorithm presented in this paper achieves similar scores as the best performing biclustering techniques in this study. This may be explained by the rather global evaluation approach pursued here, and a more specific analysis may lead to different results. Nevertheless, the reference method can be useful as a preprocessing step by which potentially relevant biclusters may be identified; later, the chosen biclusters can be used, e.g., as an input for more

accurate biclustering methods in order to speed up the processing time and to increase the bicluster quality. An advantage of Bimax is that it is capable of generating all optimal biclusters, given the underlying binary data model.

ACKNOWLEDGEMENT

Amela Prelić, Stefan Bleuler, Philip Zimmermann, and Anja Wille have been supported by the SEP program at ETH Zürich under the Poly Project TH-8/02-2.

REFERENCES

Alexe, G., Alexe, S., Crama, Y., Foldes, S., L.Hammer, P., Simeone, B., (2002) Consensus Algorithms for the Generation of All Maximal Bicliques, *Technical Report TF-DIMACS-2002-52*

Azuaje, F., (2002) A Cluster Validity Framework for Genome Expression Data. *Bioinformatics*, **18**, 319-320.

Ben-Dor, A., Chor, B., Karp, R., Yakhini, Z., (2002) Discovering Local Structure in Gene Expression Data: The Order-Preserving Sub-Matrix Problem, *Proceedings of the 6th Annual International Conference on Computational Biology*, **1-58113-498-3**, 49-57.

Berriz, G.F., King, O.D., Bryant, B., Sander, C., Roth, F.P., (2003) Characterizing Gene Sets with FuncAssociate, *BioInformatics*, **19(18)**, 2502-4.

Bezdek, J.C., (1981) Pattern Recognition with Fuzzy Objective Function Algorithms, *Plenum Press, New York*.

Cheng,Y., Church,G., (2000) Biclustering of Expression Data, *ISMB*, 93-103.

Datta, S., Datta, S., (2003) Comparisons and Validation of Statistical Clustering Techniques for Microarray Gene Expression Data, *Bioinformatics*, **19**, 459-466.

Gilchrist, M.A., Salter, L.A., Wagner, A. (2004) A statistical framework for combining and interpreting proteomic datasets, *Bioinformatics*, **20(5)**, 689–700.

Gasch, A.P., Spellman, P.T., Kao, C.M., Carmel-Harel, O., Eisen, M.B., Storz, G., Botstein, D., Brown, P.O., (2000) Genomic Expression Programs in the Response of Yeast Cells to Environmental Changes, *Mol. Biol. Cell*, **11**, 4241-4257.

Gat-Viks, I., Sharan, R., Shamir, R., (2003) Scoring Clustering Solutions by Their Biological Relevance, *Bioinformatics*, **19**, 2381-2389.

Halkidi, M., Batistakis, Y., Vazirgiannis, M., (2001) On Clustering Validation Techniques, *Journal of Intelligent Information Systems*, **17:2/3**, 107-145.

Hartigan, J.A. and Wong, M.A. (1979) A k-means Clustering Algorithm. *Applied Statistics*, **28**, 100-108.

Hartigan, J.A., (1972) Direct Clustering of a Data Matrix. *Journal of the American Statistical Organization*, **67**, 123-129.

Hartigan, J.A., (1975) Clustering Algorithms, *New York: John Willey and Sons, Inc.*

Ideker, T., Ozier, O, Schwikowski, B., Siegel, Andrew F., (2002) Discovering Regulatory and Signaling Circuits in Molecular Interaction Networks, *Bioinformatics*, **18**, S233-40

Ihmels, J., Bergmann, Barkai, N., (2004) Defining Transcription Modules Using Large-Scale Gene Expression Data, *Bioinformatics*, **20**, 1993–2003.

Ihmels, J., Friedlander, G., Bergmann, S., Sarig, O., Ziv, Y., Barkai, N., (2002) Revealing Modular Organization in the Yeast Transcriptional Network, *Nature Genetics*, **31**, 370–7.

Kerr, M. K., Churchill, G. A., (2001) Bootstrapping Cluster Analysis: Assessing the Reliability of Conclusions From Microarray Experiments, *PNAS*, **98/16**, 8961-8965.

Madeira, S.C., Oliveira, A.L., (2004) Biclustering Algorithms for Biological Data Analysis: A Survey, *IEEE/ACM Transactions on Computational Biology and Bioinformatics* , **1**, 24-45.

Murali, T.M., Kasif, S., (2003) Extracting Conserved Gene Expression Motifs from Gene Expression Data, *Pacific Symposium on Biocomputing*, **8**, 77-88.

Sharan, R., Maron-Katz, A., Shamir, R., (2003) CLICK and EXPANDER: A System for Clustering and Visualizing Gene Expression Data, *Bioinformatics*, **14**, 1787-1799.

Sharan, R., Shamir, R., (2000) CLICK: A Clustering Algorithm with Applications to Gene Expression Analysis, *Proceedings ISMB'00*, 307 - 316.

Sokal, R.R., Michener, C.D. (1958), A Statistical Method for Evaluating Systematic Relationships, *University of Kansas Science Bulletin*, **38**, 1409-1438.

Salwinski, Lukasz, Miller, C. S., Smith, A. J., Pettit, F. K., Bowie, J. U., Eisenberg, D. (2004), The Database of Interacting Proteins: 2004 update. *Nucl. Acids Res.*, **32**, D449-451.

Tanay, A., Sharan, R., Shamir, R., (2002) Discovering Statistically Significant Biclusters in Gene Expression Data, *Bioinformatics*, **18**, 136S-144.

The Gene Ontology Consortium, (2000) Gene Ontology: Tool for the Unification of Biology, *Nature Genetics*, **25**, 93-103.

Westfall, P.H., Young, S.S. (1993) Resampling-Based Multiple Testing, *Wiley, New York*.

Wille, A., Zimmermann, P., Vranova, E. Fürholz, A., Laule O, Bleuler, S., Prelic, A., von Rohr, P., Thiele, L., Zitzler, E., Gruissem, W., and Bühlmann, P. (2004) Sparse graphical Gaussian modeling of the isoprenoid gene network in Arabidopsis thaliana, *Genome Biology*, **5(11)**, R92.

Xenarios, I., Salwinski, L., Duan, X.J., Higney, P., Kim, S.M., Eisenberg, D., (2002) DIP, the Database of Interacting Proteins: a Research Tool for Studying Cellular Networks of Protein Interactions, *Nucleic Acids Res.* , **30(1)**, 303-5.

Yang, J., Wang, H., Wang, W., Yu, P.S., (2003) Enhanced Biclustering on Expression Data. *BIBE 2003*, 321-327.

Yeung,. K.Y., Haynor, D.R., Ruzzo, W.L., (2001) Validating Clustering for Gene Expression Data, *Bioinformatics*, **17**, 309-318.

Zien, A., Küffner, R., Zimmer, R., Lengauer T., (2000), Analysis of gene expression data with pathway scores, *International Conference on Intelligent Systems for Molecular Biology (ISMB 2000)*, 407–417.

APPENDIX

Bimax Algorithm

The following algorithm realizes the divide-and-conquer strategy as illustrated in Fig. 1. Note that special operations are required for processing the W submatrices. As mentioned in the discussion of the reference model, the algorithm needs to guarantee that no duplicate or non-optimal biclusters are generated. The problem arises because W contains parts of the biclusters found in U or V, and as a consequence we need to ensure that the algorithm only considers those biclusters in W that extend over both U and V. The parameter Z serves this goal. It contains pairs of disjoint column sets that restricts the number of admissible biclusters. A bicluster (G, C) is admissible, if there is at least one column set pair (C_L, C_R) in Z such that (G, C) shares one or more columns with both sets, i.e., $C \cap C_L \neq \emptyset$ and $C \cap C_R \neq \emptyset$.

procedure $Bimax(E)$
  $Z \leftarrow (\{1, \ldots, m\}, \{1, \ldots, m\})$
  $M \leftarrow conquer(E, (\{1, \ldots, n\}, \{1, \ldots, m\}), Z)$
  return M
end procedure

procedure $conquer(E, (G, C), Z)$
  if $\forall i \in G, j \in C : e_{ij} = 1$ **then**
    return $\{(G, C)\}$
  end if
  $(G_U, G_V, G_W, C_U, C_V) = divide(E, (G, C), Z)$
  $M_U \leftarrow \emptyset, M_V \leftarrow \emptyset, M_W \leftarrow \emptyset$
  if $G_U \neq \emptyset \wedge C_U \neq \emptyset$ **then**
    $Z' \leftarrow update(Z, C_U, C_U)$
    $M_U \leftarrow conquer(E, (G_U \cup G_W, C_U), Z')$
  end if
  if $G_V \neq \emptyset \wedge C_V \neq \emptyset$ **then**
    $Z' \leftarrow update(Z, C_V, C_V)$
    $M_V \leftarrow conquer(E, (G_V \cup G_W, C_V), Z')$
  end if
  if $G_W \neq \emptyset$ **then**
    $Z' \leftarrow update(Z, C_U, C_V)$
    $M_W \leftarrow conquer(E, (G_W, C_U \cup C_V), Z')$
  end if
  return $M_U \cup M_V \cup M_W$
end procedure

procedure $divide(E, (G, C), Z)$
  $G' \leftarrow reduce(E, (G, C), Z)$
  choose $i \in G'$ with $0 < \sum_{j \in C} e_{ij} < |C|$
  if such an $i \in G'$ exists **then**
    $C_U \leftarrow \{j \mid j \in C \wedge e_{ij} = 1\}$
  else

```
32:         C_U = C
33:     end if
34:     C_V ← C \ C_U
35:     G_U ← ∅, G_V ← ∅, G_W ← ∅
36:     for each i ∈ G' do
37:         C* ← {j | j ∈ C ∧ e_ij = 1}
38:         if C* ⊆ C_U then
39:             G_U ← G_U ∪ {i}
40:         else if C* ⊆ C_V then
41:             G_V ← G_V ∪ {i}
42:         else
43:             G_W ← G_W ∪ {i}
44:         end if
45:     end for
46:     return (G_U, G_V, G_W, C_U, C_V)
47: end procedure

48: procedure update(Z, C_U, C_V)
49:     Z' ← ∅
50:     for each (C_L, C_R) ∈ Z do
51:         Z' = Z' ∪ {(C_L ∩ C_U, C_R ∩ C_V)}
52:         Z' = Z' ∪ {(C_L ∩ C_V, C_R ∩ C_U)}
53:     end for
54:     return Z'
55: end procedure

56: procedure reduce(E, (G, C), Z)
57:     G' ← ∅
58:     for each i ∈ G do
59:         C* ← {j | j ∈ C ∧ e_ij = 1}
60:         for each (C_L, C_R) ∈ Z do
61:             if C_L ∩ C* ≠ ∅ ∧ C_R ∩ C* ≠ ∅ then
62:                 G' = G' ∪ {i}
63:             end if
64:         end for
65:     end for
66:     return G'
67: end procedure
```

Bimax Running-Time Analysis

THEOREM 1. *The running-time complexity of the Bimax algorithm is $O(nm\beta)$, where β is the number of all inclusion-maximal biclusters in $E^{n\times m}, m \leq n$.*

Proof of Theorem 1. To derive an upper bound for the running-time complexity, we will first calculate the number of steps required to execute the procedure $conquer$ once, disregarding the recursive procedure calls. Afterwards, the maximum number of invokations of $conquer$ will be determined, which then leads to the overall running-time complexity.

As to the procedure $reduce$, one can observe that the number of columns stored in all pairs $(C_L, C_R) \in Z$ does not exceed $2m$. If Z is implemented as a list and C^* is represented by an array, the entire loop including lines 60 to 64 can be executed in $O(m)$ time. Accordingly, one call to $reduce$ takes $O(nm)$ steps. The same upper bound holds for the procedure $update$.

The partitioning of a submatrix is accomplished by the procedure $divide$. We assume that all sets except of C^* are implemented using list structures, while C^* is stored in an array. Thereby, the inclusion-tests can be performed in time $O(m)$, and the entire loop takes $O(nm)$ steps. Overall, the running time of the procedure amounts to $O(nm)$.

The main procedure $conquer$ requires $O(nm)$ steps to check whether (G, C) represents a bicluster (lines 7 to 9), and $O(1)$ steps to perform the union operations at line 24, again assuming a list implementation. Altogether, one invokation of $conquer$ takes $O(nm)$ time.

The question now is how many times $conquer$ is executed. Taking into account that every invokation of $conquer$ returns at least one inclusion-maximal bicluster, there are at maximum β procedure calls that do not perform any further recursive calls. In other words, the corresponding recursion tree, where each node represents one instance of $conquer$ and every directed edge stands for a recursive invokation, has at most β leaves. Each inner node of the recursion tree has an outdegree of 1, 2, or 3 depending on whether W or V are empty (U is always non-empty except of the special case that E contains only 0-cells). Suppose an instance of $conquer$ in the tree that only has one child to which the submatrix U is passed. U has at least one row that contains a 1 in all columns of U; this is the row according to which the partitioning in the parent is performed. Now, either there is another row in U that contains both 0s and 1s (line 28) or all remaining rows only contain 1s. In the former case, the partitioning of U produces at least two submatrices and therefore the outdegree of the child is at least two. In the latter case, the submatrix resulting from the partitioning contains only 1s, which in turn, means that the following invokation of $conquer$ is a leave in the recursion tree. Therefore, at least one third of all inner nodes have an outdegree greater than 1.

We first give an upper bound for the number of inner nodes with more than one child, and for this purpose disregard all nodes with outdegree 1. Consider a tree where all inner nodes have an outdegree of 2 or more and the number of leaves equals β. Then the number of inner nodes is less than $2^{(\log_2 \beta)+1} = 2\beta$. For the recursion tree, this means that there are at maximum $3 \cdot 2\beta$ inner nodes, and as a consequence the overall number of nodes and invokations of $conquer$ is of order $O(\beta)$.

By combining the two main results, (i) one $conquer$ call needs $O(nm)$ steps and (ii) there are at maximum $O(\beta)$ invokations of $conquer$, we obtain the upper bound for the running-time of the Bimax algorithm. □