

Comparison of Biclustering Methods: 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:

^aComputer Engineering and Networks Laboratory

^bInstitute for Plant Sciences and Functional Genomics Center Zurich

^cColab

^dSeminar for Statistics, ETH Zurich, Switzerland

TIK-Report No. 227

Computer Engineering and Networks Laboratory

ETH Zurich

Gloriastrasse 35, ETH-Zentrum, 8092 Zurich, Switzerland

Revised Version

February 2006

(First Version: July 2005)

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

1 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 empirical 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 usually consider a specific objective function.

The main goal of this paper is to provide a systematic comparison and evaluation of prominent biclustering methods in the light of gene classification. 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 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.

2 Related Work

There exist several studies that address the issue of comparing and validating one-dimensional clustering methods (Kerr and Churchill, 2001; Yeung *et al.*, 2001; Azuaje, 2002; Datta and Datta, 2003; Gat-Viks *et al.*, 2003; Handl *et al.*, 2005). All of them make use of different quantitative measures 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). 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. As discussed in (Handl *et al.*,

2005), external indices are preferable in order to assess the performance of an algorithm on a given data set, while internal indices can be used to investigate why a particular method does not perform well.

In the context of biclustering, mainly external validation has been used. Biological analyses and interpretations by human experts are most common for the evaluation of a single, newly proposed biclustering algorithm (Cheng and Church, 2000; Getz *et al.*, 2000; Ben-Dor *et al.*, 2002; Murali and Kasif, 2003; Bergmann *et al.*, 2003; Getz *et al.*, 2003; Ihmels *et al.*, 2004); they are usually descriptive and qualitative only, and therefore not suited for comparing multiple methods. In terms of quantitative measures, many papers rely on known classifications and categorizations given by tumor types (Tanay *et al.*, 2002; Kluger *et al.*, 2003; Murali and Kasif, 2003), GO annotations (Tanay *et al.*, 2002; Tanay *et al.*, 2004), metabolic pathways (Ihmels *et al.*, 2002), or promoter motifs (Ihmels *et al.*, 2004), which are closely related to the specific data sets under consideration. Some authors also investigate *in silico* data sets with implanted biclusters where the optimal outcome is known beforehand (Ihmels *et al.*, 2002; Ben-Dor *et al.*, 2002; Bergmann *et al.*, 2003; Yang *et al.*, 2002).

Most biclustering papers are concerned with the introduction and validation of a new approach, while only a few contain quantitative comparisons to existing methods. (Cheng and Church, 2000), and (Kluger *et al.*, 2003), validate the biclustering results in comparison to hierarchical clustering and singular value decomposition respectively. (Tanay *et al.*, 2002), and (Yang *et al.*, 2002; Yang *et al.*, 2003), provide a comparison to the algorithm by (Cheng and Church, 2000), regarding synthetic data respectively the problem formulation introduced in (Cheng and Church, 2000). In (Ihmels *et al.*, 2004), two biclustering techniques (Cheng and Church, 2000; Getz *et al.*, 2000) as well as five classical clustering methods are tested with respect to the problem formulation used by the iterative signature algorithm proposed in (Ihmels *et al.*, 2002). In most of the studies, the comparison has been carried out with regard to the gene dimension.

3 Biclustering Methods

3.1 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 are briefly described in the following; they are all based on greedy search strategies.

Cheng and Church’s Algorithm (CC) (Cheng and Church, 2000) define a bicluster to be a submatrix for which the *mean squared residue score* is below a user-defined threshold δ , where 0 represents the minimum possible value. In order to identify the largest δ -bicluster in the data, they propose a two-phase strategy: first, rows and columns are removed from the original expression matrix until the above constraint is fulfilled; later, previously deleted rows and columns are added to the resulting submatrix as long as the bicluster score does not exceed δ . This procedure is iterated several times where previously found biclusters are masked with random values. Recently, (Yang *et al.*, 2003) proposed an improved version of this algorithm which avoids the problem of random interference caused by masked biclusters.

Samba (Tanay *et al.*, 2002) presented a graph-theoretic approach to biclustering in combination with a statistical data model. In this framework, the expression matrix is modelled as a bipartite graph, a bicluster is defined as a subgraph, and a likelihood score is used in order to assess the significance of observed subgraphs. A corresponding heuristic algorithm called Samba aims at finding highly significant and distinct biclusters. In a recent study (Tanay *et al.*, 2004), this approach has been extended to integrate multiple types of experimental data.

Order Preserving Submatrix Algorithm (OPSM) In (Ben-Dor *et al.*, 2002), a bicluster is defined as a submatrix that preserves the order of the selected columns for all of the selected rows. In other words, the expression values of the genes within a bicluster induce an identical linear ordering across the selected samples. Based on a stochastic model, the authors developed a deterministic algorithm to find large and statistically significant biclusters. This concept has been taken up in a recent study by (Liu and Wang, 2003).

Iterative Signature Algorithm (ISA) The authors of (Ihmels *et al.*, 2002; Ihmels *et al.*, 2004) consider a bicluster to be a transcription module, i.e., a set of co-regulated genes together with the associated set of regulating conditions. Starting with an initial set of genes, all samples are scored with respect to this gene set and those samples are chosen for which the score exceeds a predefined threshold. In the same way, all genes are scored regarding the selected samples and a new set of genes is selected based on another user-defined threshold. The entire procedure is repeated until the set of genes and the set of samples converge, i.e., do not change anymore. Multiple biclusters can be identified by running the iterative signature algorithm on several initial gene sets.

xMotif In the framework proposed by (Murali and Kasif, 2003), biclusters are sought for which the included genes are nearly constantly expressed—across the selection of samples. In a first step, the input matrix is preprocessed by assigning each gene a set of statistically significant *states*. These states define the set of valid biclusters: a bicluster is a submatrix where each gene is exactly

in the same state for all selected samples. To identify the largest valid biclusters, an iterative search method is proposed that is run on different random seeds, similarly to ISA.

3.2 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 a reference method, namely Bimax, that uses a simple data model reflecting the fundamental idea of biclustering, while allowing to determine all optimal biclusters in reasonable time. This method has the benefit of providing a basis to investigate

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

Note that the underlying binary data model, which is described below, is only used by Bimax and does not represent the platform on the basis of which the different algorithms are compared. All methods under consideration are employed using their specific data models.

3.2.1 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 $G \subseteq \{1, \dots, n\}$ that jointly respond across a subset of samples $C \subseteq \{1, \dots, m\}$. 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.

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

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

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.

3.2.2 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 $\Theta(nm\beta \log \beta)$, where β is the number of all inclusion-maximal biclusters in $E^{n \times m}$ (see supplementary material). However, the memory requirements of this algorithm are of order $\Omega(nm\beta)$ which causes practical problems, especially for larger matrices.

In this paper, though, we propose and use a fast divide-and-conquer approach, the binary inclusion-maximal biclustering algorithm (Bimax) that requires much less memory resources ($O(nm \min\{n, m\})$), while providing a worst-case running-time complexity that for matrices containing disjoint biclusters only is of order $O(nm\beta)$ and for arbitrary matrices is of order $O(nm\beta \min\{n, m\})$. The complete algorithm and the proof of the general upper bound for the running-time complexity are given in the appendix. Bimax 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 three submatrices, one of which contains 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 , i.e., G_W is empty, the two matrices can be processed independently from each other. However, if U and V have a set G_W of rows in common as shown in Fig. 1, special care is necessary to only generate those biclusters in V that share at least one common column with C_V .

3.2.3 Limitations

Theoretically, the number of inclusion-maximal biclusters can be exponential 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 $1.13e^{+15}$ (see supplementary material). The running time for such a

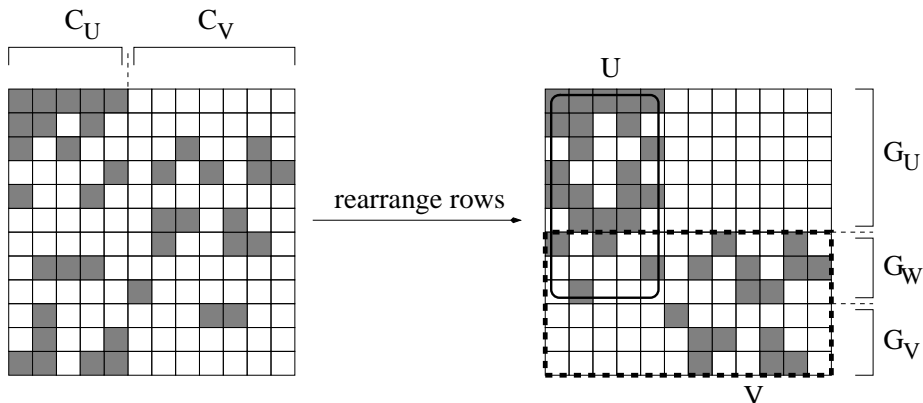


Figure 1: Illustration of the Bimax algorithm. To divide the input matrix into two smaller, possibly overlapping submatrices U and V , first the set of columns is divided into two subsets C_U and C_V , here by taking the first row as a template. Afterwards, the rows of E are resorted: first come all 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 and V which are decomposed recursively.

matrix is below 1 second on a 3 GHz Intel Xeon machine, and about 10 minutes for corresponding 6000×450 -matrices.

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, further speed-ups are achievable.

4 Comparison Methodology

In general, a fair comparison of clustering and biclustering approaches is inherently a difficult task because every method uses a different problem formulation and algorithm which may work well in certain scenarios and fail in others. Here, the main goal is to define a common setting that reflects the general basis of the majority of the biclustering studies available and in particular of those techniques considered in this paper.

First, the comparison focuses on the identification of (locally) co-expressed genes as in (Cheng and Church, 2000; Tanay *et al.*, 2002; Ben-Dor *et al.*, 2002; Ihmels *et al.*, 2002; Ihmels *et al.*, 2004; Tanay *et al.*, 2004). Classification of

samples or inference of regulatory mechanisms may be other tasks for which biclustering can be used; however, considering mainly the gene dimension has the advantage of various available annotations—in contrast to the condition dimension—and of the possibility to compare the results with classical clustering techniques.

Second, external indices are used to assess the methods under consideration as in most biclustering papers. The reasons are: (i) it is not clear how to extend notions such as homogeneity and separation (Gat-Viks *et al.*, 2003) to the biclustering context (to our best knowledge, no general internal indices have been suggested so far for biclustering), and (ii) there are some issues with internal measures, due to which (Gat-Viks *et al.*, 2003), and (Handl *et al.*, 2005), recommend external indices for evaluating the performance of (bi)clustering methods. We consider both synthetic and real data sets for the performance assessment. Only the latter allow reliable statements about the biological usefulness of a specific approach, and further biological data, namely GO annotations, as in (Tanay *et al.*, 2002; Tanay *et al.*, 2004), metabolic pathways maps, similarly to (Ihmels *et al.*, 2002), and protein-protein interactions, are used here. In contrast, the former data sets inherently reflect only certain aspects of biological reality, but they have the advantage that the optimal solutions are known beforehand and that the complexity can be controlled and arbitrarily scaled to different levels.

Finally, various biclustering concepts and structures can be considered when using *in silico* data; (Madeira and Oliveira, 2004), propose several categories on the basis of which they classify existing biclustering approaches. Here, we investigate two types of bicluster concepts: biclusters with constant expression values and biclusters following an additive model where the expression values are varying over the conditions. The former type can be used to test methods designed to identify—according to the terminology in (Madeira and Oliveira, 2004)—biclusters with constant and coherent values, while the latter type, where the expression values show the same trend for all genes included, serves as a basis to validate algorithms tailored to biclusters with coherent values and coherent evolutions. Concerning the biclustering structure, we consider two scenarios: (i) multiple biclusters without any overlap in any dimension and (ii) multiple biclusters with overlap.

4.1 Validation Using Synthetic Data

4.1.1 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 ;

on the basis of which two scenarios have been created.

In the first scenario, 10 non-overlapping transcription modules, each 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 sets contain 10 implanted biclusters and have been used to study the effects 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, the original 10 biclusters overlap where d is an indicator for the overlap degree; overall, nine different levels have been considered with $d = 0, 1, \dots, 8$.²

Moreover, we have investigated for each scenario two types of biclusters: (i) constant biclusters and (ii) additive biclusters. In the first case, the corresponding gene expression matrix E is defined by setting the expression value e_{ij} of gene i at condition j to $e_{ij} = \max_{1 \leq k \leq t} r_{ki} \cdot a_{kj}$; E is a binary matrix where the cells contained in biclusters are set to 1. In the second case, E is constructed as follows

$$e_{ij} = \begin{cases} m + (j - 1) & \text{if } \max_{1 \leq k \leq t} r_{ki} \cdot a_{kj} \neq 0 \\ U[0, m - 1] & \text{else} \end{cases}$$

where $U[l, u]$ is a uniformly randomly chosen integer in the interval $[l, u]$. In the resulting matrix, all cells belonging to an implanted bicluster have a value greater than or equal to m , while the background contains random numbers in the range of 0 to $m - 1$. Within each bicluster, the values increase column-wise by one.

4.1.2 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.

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

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

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

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

for $1 \leq i \leq t$, $1 \leq j \leq m' + d$. 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 $d \in \{0, \dots, 8\}$.

Definition 2 Let $G_1, G_2 \subseteq \{1, \dots, n\}$ 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 be 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 $M_{\text{opt}} = M$.

4.2 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; Ihmels *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.

4.3 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). For the reference method, Bimax, the discretization threshold has been set to $\underline{e} + (\bar{e} - \underline{e})/2$ where \underline{e} and \bar{e} represent the minimum respectively maximum expression values in the data matrix.

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.

5 Results

5.1 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 $n = 100, m = 100, \dots, 108$ 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.

Note that the input matrices have not been discretized beforehand, e.g., converted into binary matrices as required by the reference method Bimax. Instead, for each algorithm the corresponding preprocessing procedures have been employed as described in the relevant papers.

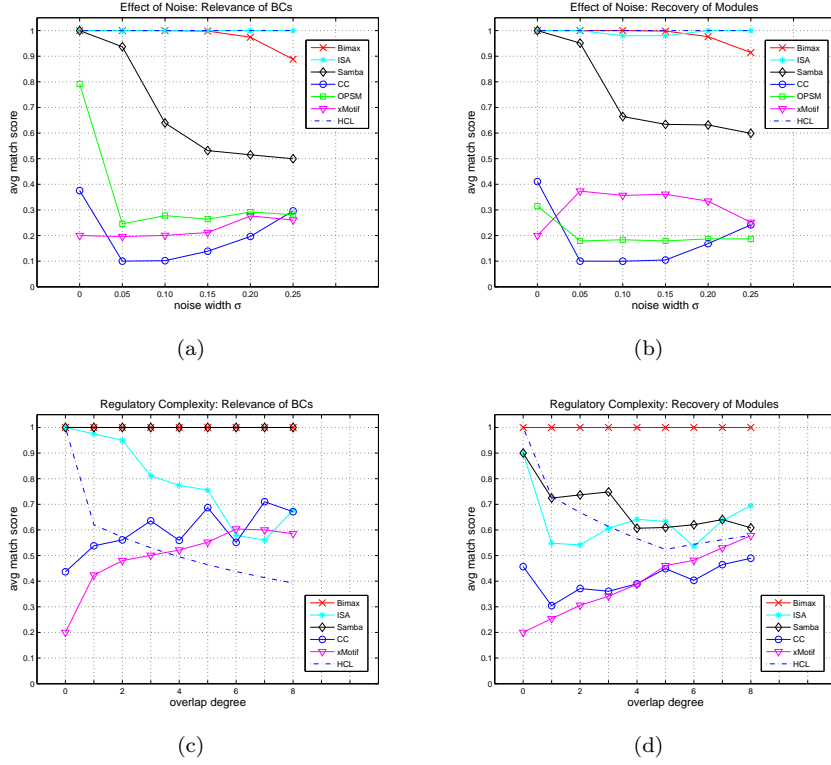


Figure 2: Results for the artificial scenarios where the implanted biclusters are characterized by constant expression values: (a), (b) non-overlapping modules with increasing noise levels; (c), (d) overlapping modules with increasing overlap degree and no noise. Note that OPSM is excluded in the lower two figures as explained in the results section.

5.1.1 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.

Noise is imitated by adding random values drawn from a normal distribution to each cell of the original gene expression matrix. The noise level, i.e., the standard deviation σ , is systematically increased, and for each noise value, 10 different data matrices have been generated from the original gene expression matrix E . 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 constant biclusters, while Fig. 3(a) and 3(b) depict the results for the matrices where the implanted biclusters represent trends over the

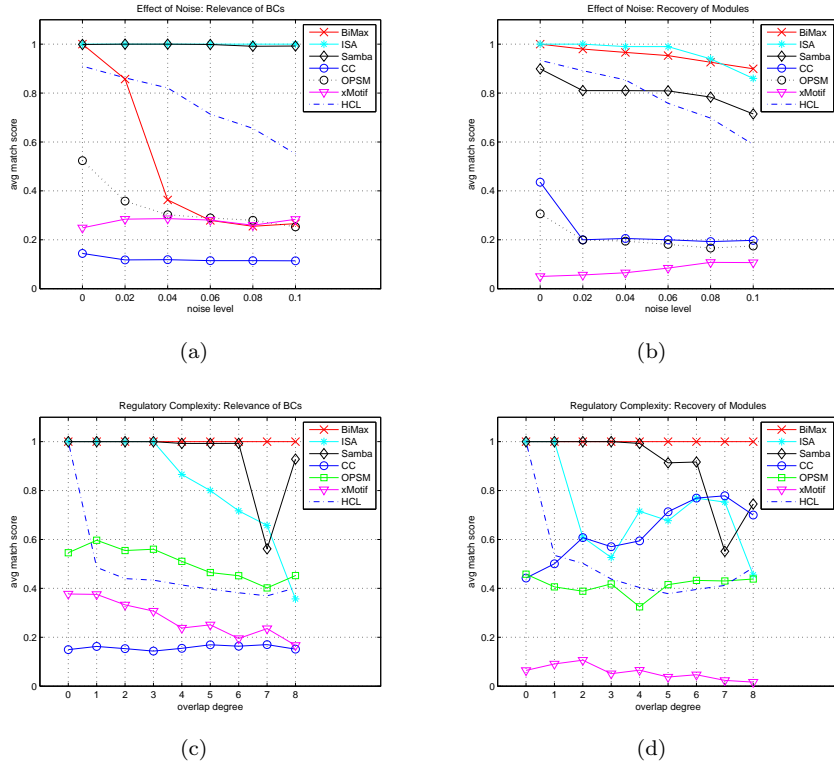


Figure 3: Results for the artificial scenarios where the biclusters follow an additive model: (a), (b) non-overlapping modules with increasing noise levels; (c), (d) overlapping modules with increasing overlap degree and no noise.

conditions.

In the absence of noise, ISA, Samba, and Bimax are able to identify a high percentage ($> 90\%$) of implanted transcription modules; as 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. In the case of constant biclusters, this phenomenon can be explained by the fact 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)). For the specific scenario with the particular type of additive biclusters considered here, CC tends to find large groups of genes extending over a few columns only, which is due to the used greedy heuristic; theoretically, the implanted biclusters achieve the optimal mean residue score. Since xMotif is mainly designed to find biclusters with coherent row values, the underlying bicluster problem formulation is not

well suited for the second bicluster type. A similar argument applies to OPSM which seeks clear trends of up- or down-regulation and cannot be expected to perform well in the scenarios with constant biclusters. The high average bicluster relevance in Fig. 2(a) is rather an artifact 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. In contrast, biclusters following an additive model with respect to the condition dimension represent optimal order-preserving submatrices. In this setting, the correspondence between the implanted biclusters and those found by OPSM is about 50%, cf. Fig. 3(a) and 3(b). A potential reason for the unexpectedly low scores is the way the heuristic algorithm works: per number of columns, only a single bicluster is considered—however, the implanted biclusters all extend over the same number of columns.

Concerning the influence of noise, ISA is only marginally affected by either type of noise and still recovers more than 90% of all implanted modules even for high noise levels. The same holds for Bimax in the constant bicluster case, but for the other bicluster type a substantial decrease in the relevance score can be observed in Fig. 3(a). This reveals a potential problem with discretization approaches: as noise blurs the differences between background and biclusters, many small submatrices emerge that not necessarily are meaningful. With HCL, noise has no observable effects in the constant bicluster scenarios, while for the second bicluster type increasing noise leads to a decrease in performance. The latter observation is due to the fact that background and biclusters are not that clearly separated in the data sets with biclusters exhibiting trends. Samba seems to be sensitive to noise in the constant bicluster case 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. In the case of additive biclusters, noise has only little effect on the performance of Samba. Concerning OPSM, noise affects the outcome; the scores slightly decrease. Remarkably, the performance of CC on the constant bicluster matrices 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.

5.1.2 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(c) and 2(d) (constant biclusters) as well as Fig. 3(c) and 3(d) (ad-

ditive biclusters) depict the results for different overlap degrees in the absence of noise, cf. the description of the data sets on Page 9. 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: most of the biclusters found ($> 90\%$) represent hidden modules³; however, not all implanted modules are recovered. While OPSM is not significantly affected by the overlap degree (only the non-constant bicluster data sets have been considered as OPSM cannot handle identical expression values), ISA appears to be more sensitive to increased regulatory complexity, especially with the second bicluster type. An explanation for this is the normalization step in the first preprocessing step of the algorithm. With increasing overlap, the expression value range after normalization becomes narrower. As a result, the differences between unchanged and up- or down-regulated expression values blur 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, the performance increases with larger overlaps degrees, but the gene match scores are still lower than the ones by Bimax, Samba, and ISA; again, this is due to the fact that the number of background cells diminishes with larger overlaps. xMotif shows the same behavior on the data matrices with constant biclusters. 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.

5.2 Real Data

Any artificial scenario inevitably is biased regarding the underlying model and only reflects certain aspects of biological reality. Therefore, the algorithms are tested in the following on real data sets⁴ and the biological relevance of the obtained biclusters is evaluated with respect to GO annotations, metabolic pathway maps, and protein-protein interaction data.

5.2.1 Functional Enrichment

The histogram in Fig. 4 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

³The outlier in Fig. 3(c) at overlap degree 7 may be explained by random effects within the Samba algorithm. Repeated applications of Samba on the same matrix, though, yielded similar scores.

⁴The gene expression matrices have been normalized using mean centering.

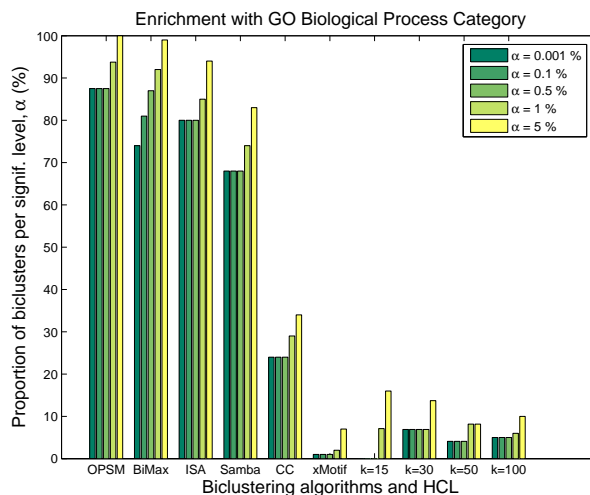


Figure 4: Proportion of biclusters significantly enriched by any GO Biological Process category (*Saccharomyces cerevisiae*) for the six selected biclustering methods as well as for hierarchical clustering with $k \in \{15, 30, 50, 100\}$. The columns are grouped method-wise, and different bars within a group represent the results obtained for five different significance levels α .

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 functionally 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 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.

5.2.2 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

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
xMotif	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

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%.

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).

6 Conclusions

The present study compares five prominent biclustering methods with respect to their capability of identifying groups of (locally) co-expressed genes; hierarchical clustering and a baseline biclustering algorithm, Bimax, proposed in this paper serve as a reference. To this end, different synthetic gene expression data sets corresponding to different notions of biclusters as well as real transcription profiling data are considered. 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 significant 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.
- 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. While Samba and ISA can be used to find multiple biclusters with both constant and coherently increasing values, OPSM is mainly tailored to identify a single bicluster of the latter type. Proposed extensions of the OPSM approach such as (Liu and Wang, 2003) may resolve these issues. 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.

Acknowledgment

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.
- Bergmann, S., Ihmels, J., Barkai, N., (2003) Iterative Signature Algorithm for the Analysis of Large-scale Gene Expression Data, *Phys Rev E Stat Nonlin Soft Matter Phys*, **67(3)**, 031902.
- 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.
- Getz, G., Levine, E., Domany, E., (2000) Coupled Two-way Clustering Analysis of Gene Microarray Data, *PNAS*, **97(22)**, 12079–12804.
- Getz, G., Gal, H., Kela, I., Notterman, D. A., Domany, E., (2003) Coupled Two-way Clustering Analysis of Breast Cancer and Colon Cancer Gene Expression Data, *Bioinformatics*, **19(9)**, 1079–1089.

- 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**, 370–377.
- Handl, J., Knowles, J., Kell, D.B., (2005) Computational Cluster Validation in Post-Genomic Data Analysis, *Bioinformatics*, **21/15**, 3201–3212.
- Kerr, M. K., Churchill, G. A., (2001) Bootstrapping Cluster Analysis: Assessing the Reliability of Conclusions From Microarray Experiments, *PNAS*, **98/16**, 8961–8965.
- Kluger, Y., Basri, R., Chang, J. T., Gerstein, M., (2003) Spectral Biclustering of Microarray Cancer Data: Co-clustering Genes and Conditions, *Genome Research*, **13**, 703–716.
- Liu, J., Wang, W., (2003) OP-Clusters: Clustering by tendency in high dimensional space, *Proceedings of the 3rd IEEE International Conference on Data Mining (ICDM)*, 187-194.
- 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.
- Tanay, A., Sharan, R., Kupiec, M., Shamir, R., (2004) Revealing Modularity and Organization in the Yeast Molecular Network by Integrated Analysis of Highly Heterogeneous Genomewide Data, *PNAS*, **101-9**, 2981-2986.
- 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, W., Wang, H., Yu, P.S., (2002) Delta-Clusters: Capturing Subspace Correlation in a Large Data Set, *IEEE International Conference on Data Engineering (ICDE)*, 517-528.
- 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.

A Appendix

A.1 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 V submatrices. As mentioned in the discussion of the reference model, the algorithm needs to guarantee that only optimal, i.e., inclusion-maximal biclusters are generated. The problem arises because V contains parts of the biclusters found in U , and as a consequence we need to ensure that the algorithm only considers those biclusters in V that extend over C_V . The parameter Z serves this goal. It contains sets of columns that restricts the number of admissible biclusters. A bicluster (G, C) is admissible, if (G, C) shares one or more columns with each column set C^+ in Z , i.e., $\forall C^+ \in Z : C \cap C^+ \neq \emptyset$.

```

1: procedure Bimax( $E$ )
2:    $Z \leftarrow \emptyset$ 
3:    $M \leftarrow \text{conquer}(E, (\{1, \dots, n\}, \{1, \dots, m\}), Z)$ 
4:   return  $M$ 
5: end procedure

6: procedure conquer( $E, (G, C), Z$ )
7:   if  $\forall i \in G, j \in C : e_{ij} = 1$  then
8:     return  $\{(G, C)\}$ 
9:   end if
10:   $(G_U, G_V, G_W, C_U, C_V) = \text{divide}(E, (G, C), Z)$ 
11:   $M_U \leftarrow \emptyset, M_V \leftarrow \emptyset$ 
12:  if  $G_U \neq \emptyset$  then
13:     $M_U \leftarrow \text{conquer}(E, (G_U \cup G_W, C_U), Z)$ 
14:  end if
15:  if  $G_V \neq \emptyset \wedge G_W = \emptyset$  then
16:     $M_V \leftarrow \text{conquer}(E, (G_V, C_V), Z)$ 
17:  else if  $G_W \neq \emptyset$  then
18:     $Z' \leftarrow Z \cup \{C_V\}$ 
19:     $M_V \leftarrow \text{conquer}(E, (G_W \cup G_V, C_U \cup C_V), Z')$ 
20:  end if
21:  return  $M_U \dot{\cup} M_V$ 
22: end procedure

23: procedure divide( $E, (G, C), Z$ )
24:   $G' \leftarrow \text{reduce}(E, (G, C), Z)$ 
25:  choose  $i \in G'$  with  $0 < \sum_{j \in C} e_{ij} < |C|$ 
26:  if such an  $i \in G'$  exists then
27:     $C_U \leftarrow \{j \mid j \in C \wedge e_{ij} = 1\}$ 
28:  else
29:     $C_U = C$ 

```



```

30:   end if
31:    $C_V \leftarrow C \setminus C_U$ 
32:    $G_U \leftarrow \emptyset, G_V \leftarrow \emptyset, G_W \leftarrow \emptyset$ 
33:   for each  $i \in G'$  do
34:      $C^* \leftarrow \{j \mid j \in C \wedge e_{ij} = 1\}$ 
35:     if  $C^* \subseteq C_U$  then
36:        $G_U \leftarrow G_U \cup \{i\}$ 
37:     else if  $C^* \subseteq C_V$  then
38:        $G_V \leftarrow G_V \cup \{i\}$ 
39:     else
40:        $G_W \leftarrow G_W \cup \{i\}$ 
41:     end if
42:   end for
43:   return  $(G_U, G_V, G_W, C_U, C_V)$ 
44: end procedure

45: procedure reduce( $E, (G, C), Z$ )
46:    $G' \leftarrow \emptyset$ 
47:   for each  $i \in G$  do
48:      $C^* \leftarrow \{j \mid j \in C \wedge e_{ij} = 1\}$ 
49:     if  $C^* \neq \emptyset \wedge \forall C^+ \in Z : C^+ \cap C^* \neq \emptyset$  then
50:        $G' = G' \cup \{i\}$ 
51:     end if
52:   end for
53:   return  $G'$ 
54: end procedure

```

A.2 Bimax Running-Time Analysis

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

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 invocations 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 column sets stored in Z is bounded by the number of rows, n , and each column set contains at most m elements. If Z is implemented as a list and C^* is represented by an array, the if statement in line 49 can be executed in $O(nm)$ time. Accordingly, one call to *reduce* takes $O(n^2m)$ steps resp. $O(m^2n)$ steps, if $n > m$ and the transposed matrix is considered. Overall, the running time complexity is of order $O(nm \min\{n, m\})$.

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 valid bicluster (lines 7 to 9), and $O(1)$ steps to perform the union operations at lined 18 and 21, again assuming a list implementation. Altogether, one invocation of *conquer* takes $O(nm)$ time.

The question now is how many times *conquer* is executed. Taking into account that every invocation 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 invocation, has at most β leafs. Each inner node of the recursion tree has an outdegree of 1 or 2, on whether G_W and G_V are empty (G_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 25) or all remaining rows only contain 1s. In the former case, the partitioning of U produces a non-empty set G_W and therefore the outdegree of the child is two. In the latter case, the submatrix resulting from the partitioning contains only 1s, which in turn, means that the following invocation of *conquer* is a leaf in the recursion tree. Therefore, at least one half 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 and the number of leafs 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 $2 \cdot 2\beta$ inner nodes, and as a consequence the overall number of nodes and invocations of *conquer* is of order $O(\beta)$.

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

B Supplementary Material

B.1 Incremental Algorithm

The incremental procedure, see below, is based on work by (Alexe *et al.*, 2002), who propose a method to find all inclusion-maximal cliques in general graphs. Shortly summarized, each node in the input graph is visited, and all maximal

cliques are found that contain that node. A visit-to-a-node operation comprises an iteration through all other nodes of the graph as well, and each newly found bicluster is globally extended to its maximality. For the special class of bipartite graphs we are dealing with, it is important to notice that several steps of the above method are redundant: it suffices to iterate through only one partition of the graph nodes—in matrix terminology this means we will have to iterate either through the set of rows or columns, but not both. Moreover, extending new biclusters can be avoided with a guarantee that no bicluster will be missed this way.

```

1: procedure IncrementalAlgorithm( $E$ )
2:    $M \leftarrow \emptyset$ 
3:   for  $i \leftarrow 1$  to  $n$  do
4:      $C^* \leftarrow \{j \mid e_{ij} = 1 \wedge 1 \leq j \leq m\}$ 
5:     for each  $(G, C) \in M$  do
6:        $C' \leftarrow C \cap C^*$ 
7:       if  $\exists (G'', C'') \in M$  with  $C'' = C'$  then
8:          $M \leftarrow M \setminus \{(G'', C'')\} \cup \{(G'' \cup \{i\}, C'')\}$ 
9:       else
10:         $M \leftarrow M \cup \{(G'' \cup \{i\}, C')\}$ 
11:      end if
12:    end for
13:    if  $\nexists (G'', C'') \in M$  with  $C'' = C^*$  then
14:       $M \leftarrow M \cup \{\{i\}, C^*\}$ 
15:    end if
16:  end for
17:  return  $M$ 
18: end procedure

```

B.2 Incremental Algorithm Running-Time Analysis

Theorem 2 *The running-time complexity of the Incremental Algorithm is of order $\Theta(nm\beta \log \beta)$, where β is the number of all inclusion-maximal biclusters in $E^{n \times m}$.*

Lemma 1 *Given the binary matrix $E^{n \times m}$, a duplicate row or column in E does not contribute to the total number of all inclusion-maximal biclusters in E .*

Lemma 2 *Given the binary matrix $E^{n \times m}$, the upper bound on the number of all inclusion-maximal biclusters in E is $(2^{\min(n,m)} - 1)$.*

Proof of Theorem 2. The incremental algorithm proceeds in stages: at stage i , a row/gene i of the matrix is considered and the steps within the outer **for** instruction are performed. The set of instructions within steps 5 to 12 amounts to: i) computing an intersection of the sets of samples (having value 1) corresponding to gene i and a currently considered bicluster, which takes $\Theta(m)$, and

ii) the search through the list M , followed by a set equality comparison operations, which costs further $\Theta(m \log_2 \beta)$, assuming that binary search through the list M is made. This inner cycle (steps 5 - 12) is performed β times, and the outer one n times, where n is the number of rows of the matrix E . We then obtain $\Theta(n \beta (m + m \log_2 \beta)) = \Theta(n m \beta \log_2 \beta)$. Note that the worst-case running time complexity amounts to $O(n m^2 \beta)$ in the case that $m \leq n$, because the upper bound on β is then exponential in m , hence, $\log_2 \beta \leq m$.

In the algorithm proposed by (Alexe *et al.*, 2002), the main differences to our incremental approach is an additional step that is performed within the steps 7 to 11 of globally extending newly created biclusters to their maximality, and an additional "absorption check" operation is made which costs $\Theta(n m \log_2 \beta)$. Hence, the difference in the running-time complexities. \square

B.3 Additional Tables and Figures

density	number of samples m				
$D^{6000 \times \dots}$	50	150	250	350	450
1 %	530.0	3475.5	7594.2	12405.5	17919.9
2 %	1468.7	11829.2	28938.8	53438.2	86657.3
3 %	2490.1	21693.7	62005.3	132435.8	238598.5
4 %	3933.7	44463.7	155929.8	367228.8	694202
5 %	6554.9	100213.8	390835	956255	1838979.7
	1.13e+15	1.43e+45	1.81e+75	2.29e+105	2.91e+135

Table 2: Average number of inclusion-maximal biclusters for random matrices with 6000 genes and varying number of columns and densities, i.e., proportion of 1 cells to 0-cells. Each number gives the average over 100 matrices. The last row comprises the theoretical upper bounds for the number of inclusion-maximal biclusters.

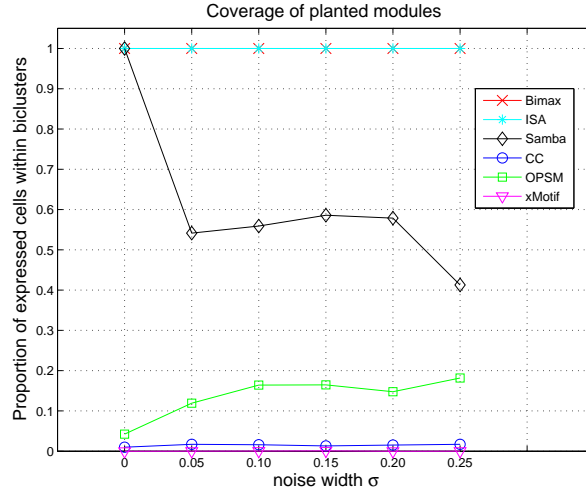


Figure 5: This figure shows for the first artificial scenario with constant biclusters what proportion of computed biclusters contain over-expressed cells. As argued in the article, the two methods CC and xMotif tend to produce large biclusters covering the background area of the input matrix, i.e., the cells containing 0).

Algorithm	Default Parameter Settings
Samba	$D = 40, N_1 = 4, N_2 = 6, k = 20, L = 30$
ISA	$t_g = 1.8 - 4.0$ (step 0.1), $t_c = 2.0$, nr. seeds = 20000
CC	$\alpha = 1.2, \delta$ lower end of the range of expression values
OPSM	$l = 100$
xMotifs	$n_s = 10, n_d = 1000, s_d = 7 - 10, \alpha$ not given, P value 10^{-10} , max_length not given
Algorithm	Changed values
Samba	
ISA	$t_g = 2.0$, nr. seeds = 500
CC	$\delta \leq 0.5$, for biclusters with increasing values $\delta = 0.1$
OPSM	
xMotifs	$s_d = 7, \alpha = 0.1, max_length = 0.7$ for increasing noise matrix p-value 10^{-7}

Table 3: Parameter settings used for different biclustering methods. Default settings (i.e., the parameter values recommended/used by the authors of original papers) were occasionally changed in order to force the methods to output at least a single bicluster. The changed values are reported in the third column (an empty third column cell indicates the default values have always been used). For the meaning of different parameters, please refer to the original papers.

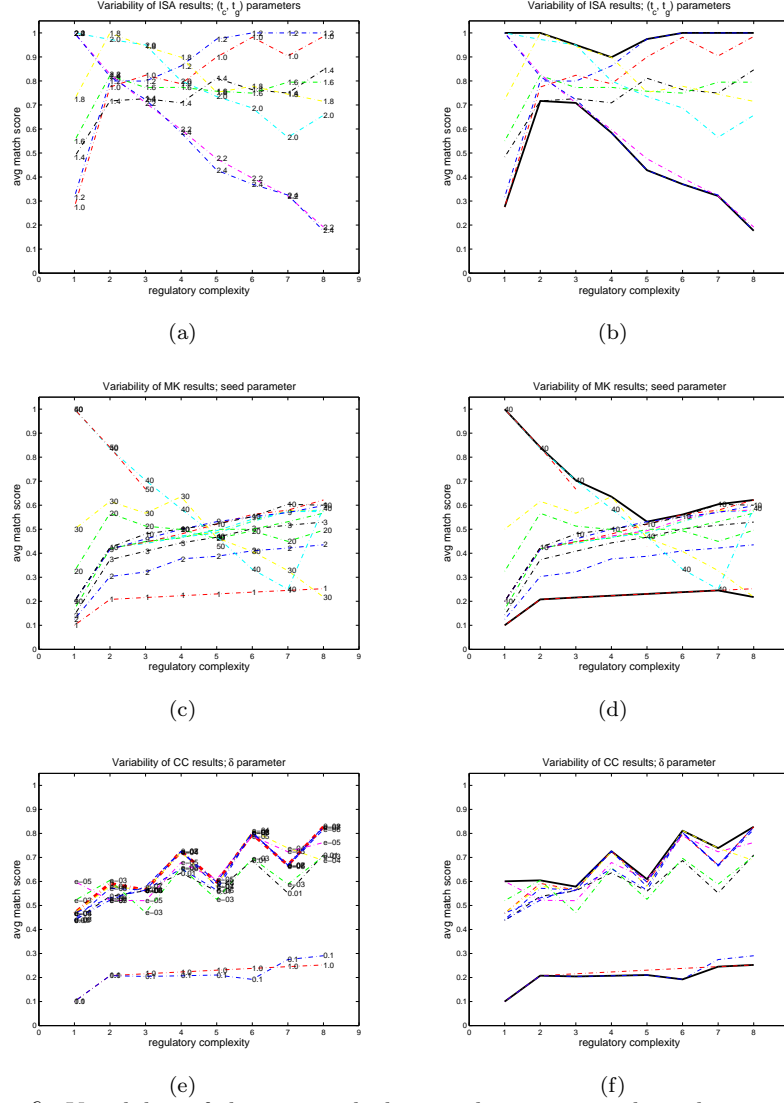


Figure 6: Variability of the average bicluster relevance score depending on the parameter settings (constant biclusters). The plotted values represent averages over the biclusters obtained by ISA, xMotif and CC. (a), (b): For ISA, we varied the (t_g, t_c) parameters, in all cases, $t_g = t_c$, with $1.0 \leq t_g \leq 2.4$; the value recommended by authors is $(2.0, 2.0)$. (c), (d): As to xMotif, the size of the random seeds was changed in the range 1 – 50; values recommended by the authors are in the range 7 – 10. (e), (f): For CC, the homogeneity threshold, δ , has been systematically varied; the red bold line in (e) shows the results obtained for $\delta = 0$, i.e., when only perfect biclusters are sought.