SUPPLEMENTARY MATERIAL (WEB PAGE)

Considered Biclustering 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 orginal expression matrix until the above constraint is fulfilled; later, previously deleted rows and columns are added to the resulting submatrix as along 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 stochastical 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, 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.

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:          $C^{\star} \leftarrow \{j \mid e_{ij} = 1 \wedge 1 \leq j \leq m\}$
5:          **for each** $(G, C) \in M$ **do**
6:              $C' \leftarrow C \cap C^{\star}$
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 $C'' = C^{\star}$ **then**
14:             $M \leftarrow M \cup \{(\{i\}, C^{\star})\}$
15:         **end if**
16:     **end for**
17:     **return** M
18: **end procedure**

Incremental Algorithm Running-Time Analysis

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

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 $\Theta(n \; \beta \; (m + m \; log_2\beta)) \; = \; \Theta(n \; m \; \beta \; \log_2 \beta) \; = \; \Theta(n \; m^2\beta)$; assuming $m \leq n$, the upper bound on β is exponential in m, hence, $\log_2 \beta = m$.

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. $\square$

Table 3. 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.

density	number of samples m				
$D^{6000 \times \cdots}$	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

Additional Tables and Figures

Table 2. Average response times for Bimax in comparison with the incremental approach on 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 running time measured in seconds over 100 matrices.

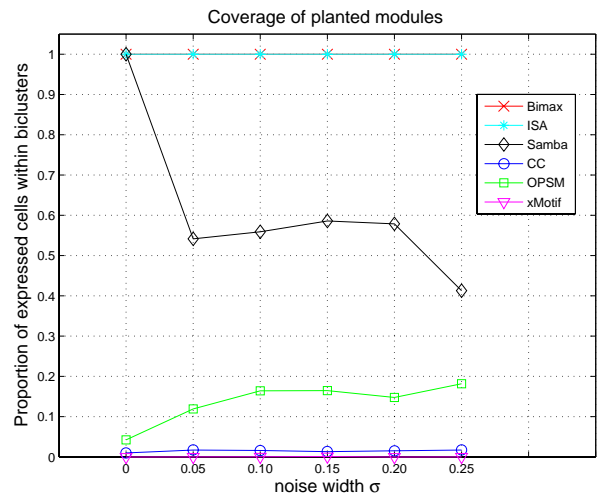| density		number of samples m				
$E^{6000 \times \cdots}$		50	150	250	350	450
Bimax	1 %	0.65	2.31	5.04	8.76	13.64
	2 %	0.96	7.53	22.04	43.78	73.04
	3 %	1.36	15.63	61.52	142.76	261.27
	4 %	2.00	29.45	117.88	363.02	754.05
	5 %	3.10	57.52	231.94	786.01	2128.9
Incremental	1 %	2	12.8	28.3	48.3	73
	2 %	5.6	52.4	140.3	319.8	777.2
	3 %	11.3	120.2	483.2	1619.7	3780.7
	4 %	19.4	299.3	2134.6	9938.2	21054.6
	5 %	34	985.8	11395	43213.3	134282.1

Fig. 4: This figure shows for the first artificial scenario 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).
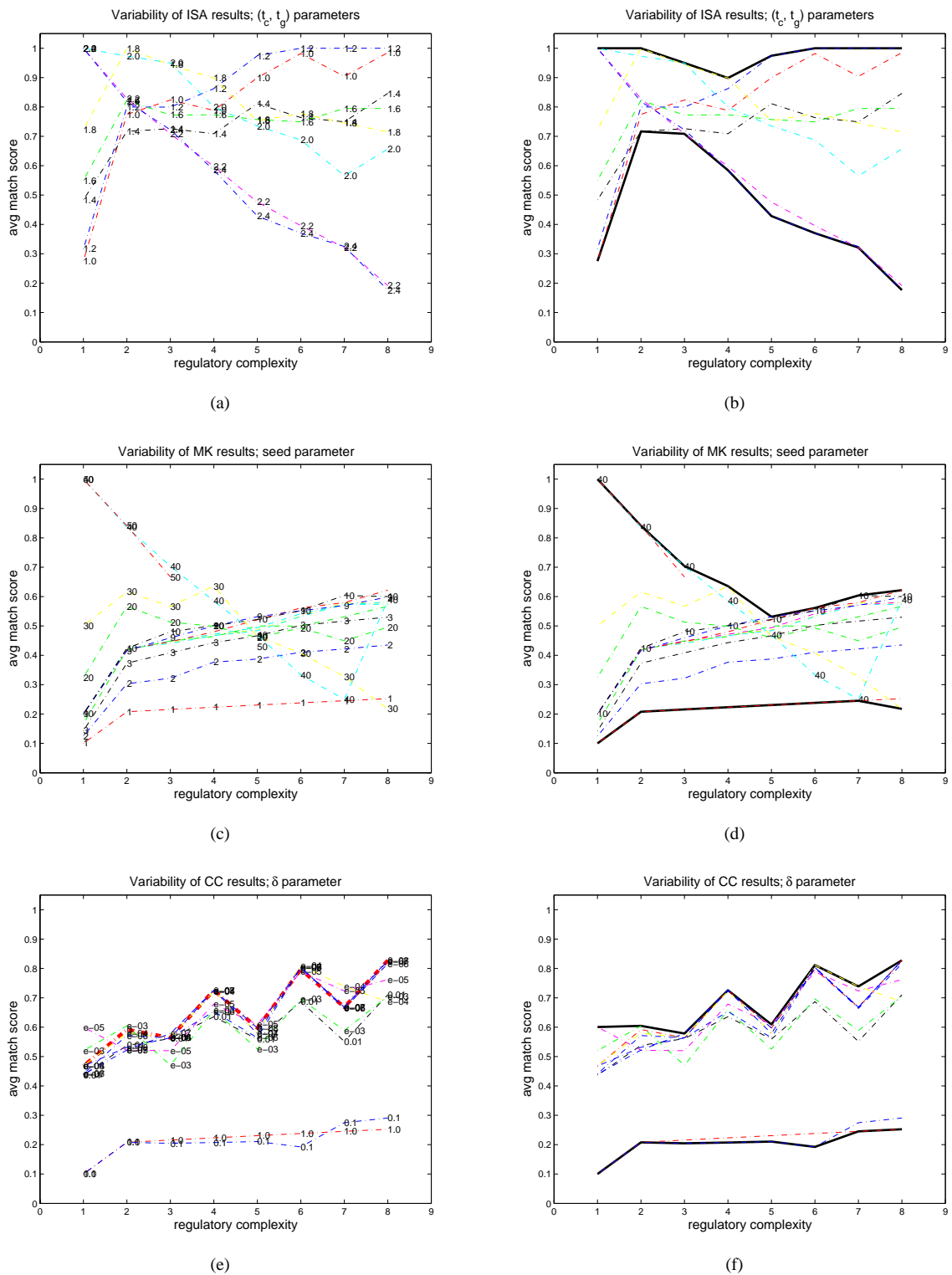
(a)

(b)

(c)

(d)

(e)

(f)

Fig. 5: Variability of the average bicluster relevance score depending on the parameter settings. 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.

Table 4. 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.

Algorithm	Default Parameter Settings	Changed values
Samba	$D = 40,\ N_1 = 4,\ N_2 = 6,\ k = 20,\ L = 30$	
ISA	$t_g = 1.8 - 4.0\ (\text{step } 0.1), t_c = 2.0, \text{nr. seeds} = 20000$	$t_g = 2.0, \text{nr. seeds} = 500$
CC	$\alpha = 1.2, \delta$ lower end of the range of expression values	$\delta \leq 0.5$
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	$s_d = 7, \alpha = 0.1, max\_length = 0.7m$

REFERENCES

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

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.

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

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.

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.

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

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.

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