FABIA: Factor Analysis for Bicluster Acquisition

Sepp Hochreiter[1,*], Ulrich Bodenhofer[1], Martin Heusel[1], Andreas Mayr[1], Andreas Mitterecker[1], Adetayo Kasim[3], Tatsiana Khamiakova[3], Suzy Van Sanden[3], Dan Lin[3], Willem Talloen[4], Luc Bijnens[4], Hinrich W. H. Göhlmann[4], Ziv Shkedy[3], and Djork-Arné Clevert[1,2]

[1]Institute of Bioinformatics, Johannes Kepler University, Linz, Austria
[2]Department of Nephrology and Internal Intensive Care, Charité, Berlin, Germany
[3]Institute for Biostatistics and Statistical Bioinformatics, Hasselt University, Hasselt, Belgium
[4]Johnson & Johnson Pharmaceutical Research & Development, a Division of Janssen Pharmaceutica, Beerse, Belgium

Received on XXXXX; revised on XXXXX; accepted on XXXXX

Associate Editor: XXXXXXX

ABSTRACT

Motivation: Biclustering of transcriptomic data, that is, clustering genes and samples simultaneously, is an important unsupervised approach to extract knowledge from gene expression measurements. However, most biclustering methods do not apply generative models which would allow to utilize well understood model selection techniques and to apply the Bayesian framework. The few existing generative models are restricted to additive models and therefore not suited to explain effects due to mRNA degradation or PCR amplification. Further, they assume Gaussian distributions which cannot explain the heavy tailed distributions of microarray data. We introduce a novel generative model for biclustering called "Factor Analysis for Bicluster Acquisition" (FABIA). FABIA is based on a multiplicative model that assumes realistic non-Gaussian signal distributions with heavy tails.

Results: On 100 simulated data sets with known true, artificially implanted biclusters, FABIA clearly outperformed all 11 competitors. The generative framework allows to determine the information content of each bicluster and hence to separate spurious biclusters from true biclusters as shown in the experiments. FABIA was tested on three microarray data sets with known sub-clusters, where it was two times the best and once the second best method among 11 biclustering approaches.

Availability: FABIA is available as an R package on Bioconductor (http://www.bioconductor.org). All data sets, results, and software can be found at http://www.bioinf.jku.at/software/fabia/fabia.html.

Contact: hochreit@bioinf.jku.at

1 INTRODUCTION

Recent array technologies like the Affymetrix array plates open up new possibilities for high-throughput expression profiling. The same is expected for next-generation transcriptome sequencing. These technologies in turn require advanced analysis tools to extract knowledge from the huge amount of data. If for the data analysis the experimental conditions like osmotic pressure or temperature are known, supervised techniques such as support vector machines are suitable to extract the dependencies between conditions and gene expression profiles or to identify condition-indicative genes. However, conditions may not be known or biologists and medical researchers are interested in dependencies within conditions or across conditions. For instance, it could be possible to refine pathways across conditions or to identify new subgroups within one condition. For these tasks, unsupervised methods like clustering and projection approaches are required. Conventional clustering techniques, such as hierarchical or k-means clustering, are typically not sufficient, because samples may only be similar to each other on a subset of genes and vice versa. In drug design, for example, researchers want to reveal how compounds affect gene expression; the compounds, however, may be similar to each other only on a subgroup of genes.

For unsupervised analysis of transcriptomic data, biclustering algorithms that simultaneously cluster the genes and the samples are of high interest. A *bicluster* of a transcriptomic data set is a pair of a gene set and a sample set for which the genes are similar to each other on the samples and vice versa. A sample may belong to different biclusters, e.g. if more than one pathway is active in that sample. A gene may belong to different biclusters, for example, if this gene participates in different pathways for different conditions. Thus, biclusters can overlap.

A survey over various biclustering approaches has been given by Madeira and Oliveira (2004). In principle, there exist four categories of biclustering methods: (1) variance minimization methods, (2) two-way clustering methods, (3) motif and pattern recognition methods, and (4) probabilistic and generative approaches. Transcriptomic data are usually supplied as a matrix, where each gene corresponds to one row and each sample to one column; the matrix entries themselves are the expression levels.

(1) Variance minimization methods define clusters as blocks in the matrix with minimal deviation of their elements. This definition has been already considered by Hartigan (1972) and extended by Tibshirani *et al.* (1999). The δ-cluster methods search for blocks of elements having a deviation ("variance") below δ. One example

[*] to whom correspondence should be addressed

are δ-ks clusters (Califano *et al.*, 2000), where the maximum and the minimum of each row need to differ less than δ on the selected columns. A second example are δ-pClusters (Wang *et al.*, 2002) which are defined as 2×2 sub-matrices with pairwise edge differences less than δ. A third example are the Cheng and Church (2000) δ-biclusters having a mean squared error below δ after fitting an additive model with a constant, a row, and a column effect. FLexible Overlapped biClustering (FLOC; Yang *et al.*, 2005) extend Cheng-Church δ-biclusters by dealing with missing values via an occupancy threshold θ and by using both l_1 and l_2 norms.

(2) Two-way clustering methods apply conventional clustering to the columns and rows and (iteratively) combine the results. Coupled Two-Way Clustering (CTWC; Getz *et al.*, 2000) iteratively performs standard clustering of the rows (columns) using previously constructed columns (rows) clusters as features. Also Interrelated Two-Way Clustering (ITWC; Tang *et al.*, 2001) using k-means and Double Conjugated Clustering (DCC; Busygin *et al.*, 2002) using self-organizing maps integrate the results of column and row clustering.

(3) Motif and pattern recognition methods define a bicluster as samples sharing a common pattern or motif. To simplify this task, some methods discretize the data in a first step, like xMOTIF (Murali and Kasif, 2003) or Bimax (Prelic *et al.*, 2006) which even binarizes the data and searches for blocks with an enrichment of ones. Order-Preserving Sub-Matrices (OPSM; Ben-Dor *et al.*, 2003) searches for blocks having the same order of values in their columns. Using partial models, only the column order on subsets must be preserved. Spectral clustering (SPEC; Kluger *et al.*, 2003) performs a singular value decomposition of the data matrix after normalization. SPEC extracts columns (samples) with the same conserved gene expression pattern using the fact that they are linearly dependent and span a subspace associated with a certain singular value.

(4) Probabilistic and generative methods use model-based techniques to define biclusters. Statistical-Algorithmic Method for Bicluster Analysis (SAMBA; Tanay *et al.*, 2002) uses a bi-partitioned graph, where both conditions and genes are nodes. An edge from a gene to a condition means that the gene responds to the condition. With a probabilistic objective, subgraphs are found that have a significantly higher connectivity than the overall graph. In another approach, Sheng *et al.* (2003) use Gibbs sampling to estimate the parameters of a simple frequency model for the expression pattern of a bicluster. However, the data must first be discretized and then only one bicluster with constant column values at each step can be extracted. Probabilistic Relational Models (PRMs; Getoor *et al.*, 2002; Segal *et al.*, 2003) and their extension ProBic (Van den Bulcke, 2009) are fully generative models which combine probabilistic modeling and relational logic. Another generative approach is cMonkey (Reiss *et al.*, 2006) which models biclusters by Markov chain processes. Both PRMs and cMonkey are able to integrate non-transcriptomic data sources.

In the plaid model family (Lazzeroni and Owen, 2002), the i-th bicluster is extracted by row and column indicator variables ρ_{ki} and κ_{ij}. The values of each bicluster are explained by a general additive model $\theta_{kij} = \mu_i + \alpha_{ki} + \beta_{ij}$. Parameters are estimated by a least square fit subject to $\sum_k \alpha_{ki}\rho_{ki}^2 = 0$ and $\sum_j \beta_{ij}\kappa_{ij}^2 = 0$ to enforce that α_{ki} and β_{ij} account for the deviation from mean μ_i. Gu and Liu (2008) generalized the plaid models to fully generative models called Bayesian BiClustering model (BBC). To avoid the high percentage of overlap in the plaid models, BBC constrains the overlapping of biclusters to only one dimension. Further it allows different error variances per bicluster. Caldas and Kaski (2008) also extended the plaid model to a fully generative model using a Bayesian framework and found that the plaid model is equivalent to the PRM model for specific parameters. Further it has been shown that “binary matrix factorization” (Meeds *et al.*, 2007) is the plaid model with $\alpha = \beta = 0$ (constant bicluster) if the weighting matrix is diagonal.

The latter models (Gu and Liu, 2008; Caldas and Kaski, 2008) are generative models which have the advantage that (1) they select models using well-understood model selection techniques like maximum likelihood, (2) hyperparameter selection methods (e.g. to determine the number of biclusters) can rely on the Bayesian framework, (3) signal-to-noise ratios can be computed, (4) they can be compared to each other via the likelihood or posterior, (5) tests like the likelihood ratio test are possible, and (6) they produce a global model to explain all data. These models are additive and assume that all effects are Gaussian to utilize Gibbs sampling for parameter estimation. However after prefiltering, real microarray data sets are not Gaussian distributed and have heavy tails (Hardin and Wilson, 2009), even after log-transformation, which can be seen in Figures S8, S9, and S10 in the supplementary for gene expression data sets. In this paper, we propose a *generative multiplicative model tailored to the special characteristics of gene expression data.*

This paper is organized as follows. Section 2 introduces the multiplicative bicluster model class. Section 3 describes the model selection (training) algorithm for the new model class. Section 4 highlights how biclusters can be ranked according to the information they contained about the data. Section 5 describes how to extract bicluster members from our new models. Finally, Section 6 provides a experimental validation of the new method.

2 THE FABIA MODEL

We propose a multiplicative model class for analyzing gene expression data sets for several reasons. First, a multiplicative model allows to model heavy tailed data as we observed in gene expression data. Secondly, it can relate the strength of gene expression patterns to characteristics of the induced condition like elapsed time or concentration of compounds. After log transformation, also exponential dynamics like decay (mRNA or compound) or saturation can be modeled. Note that supervised multiplicative models, e.g. support vector machines, were successfully applied to log-transformed gene expression data sets. Further, artificial multiplicative effects are introduced during data preprocessing, for example if expression values are standardized then variations stemming from noise scale the signal.

We assume that the gene expression data set is preprocessed and filtered for genes that contain a signal (e.g. informative call or signal strength). The resulting data is given as a data matrix $\boldsymbol{X} \in \mathbb{R}^{n \times l}$, where every row corresponds to a gene and every column corresponds to a sample; the value x_{kj} corresponds to the expression level of the k-th gene in the j-th sample. The matrix $\boldsymbol{X}$ is the input to biclustering methods.

We define a *bicluster* as a pair of a row (gene) set and a column (sample) set for which the rows are similar to each other on the columns and vice versa. In a multiplicative model, two vectors are similar if one is a multiple of the other, that is the angle between them is zero or as realization of random variables their correlation coefficient is one. It is clear that such a linear dependency on subsets of rows and columns can be represented as an outer product $\boldsymbol{\lambda}\, \boldsymbol{z}^T$ of two vectors $\boldsymbol{\lambda}$ and $\boldsymbol{z}$. The vector $\boldsymbol{\lambda}$ corresponds to a *prototype column vector* that contains zeros for genes not participating in the

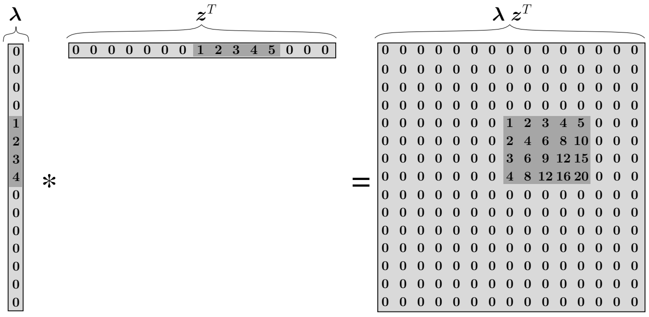

Fig. 1. The outer product $\boldsymbol{\lambda}\ \boldsymbol{z}^T$ of two sparse vectors results in a matrix with a bicluster. Note, that the non-zero entries in the vectors are adjacent to each other for visualization purposes only.

bicluster, whereas $\boldsymbol{z}$ is a vector of *factors* with which the prototype column vector is scaled for each sample; clearly $\boldsymbol{z}$ contains zeros for samples not participating in the bicluster. Vectors containing many zeros or values close to zero are called *sparse vectors*. Fig. 1 visualizes this representation by sparse vectors schematically.

The overall model for p biclusters and additive noise is

$$\boldsymbol{X} \ = \ \sum_{i=1}^{p} \boldsymbol{\lambda}_i \ \boldsymbol{z}_i^T \ + \ \boldsymbol{\Upsilon} \ = \ \boldsymbol{\Lambda} \ \boldsymbol{Z} \ + \ \boldsymbol{\Upsilon} \ , \qquad (1)$$

where $\boldsymbol{\Upsilon} \in \mathbb{R}^{n \times l}$ is additive noise and $\boldsymbol{\lambda}_i \in \mathbb{R}^n$ and $\boldsymbol{z}_i \in \mathbb{R}^l$ are the sparse prototype vector and the sparse vector of factors of the i-th bicluster, respectively. The second formulation above holds if $\boldsymbol{\Lambda} \in \mathbb{R}^{n \times p}$ is the sparse prototype matrix containing the prototype vectors $\boldsymbol{\lambda}_i$ as columns and $\boldsymbol{Z} \in \mathbb{R}^{p \times l}$ is the sparse factor matrix containing the transposed factors $\boldsymbol{z}_i^T$ as rows. Note that Eq. (1) formulates biclustering as sparse matrix factorization.

According to Eq. (1), the j-th sample $\boldsymbol{x}_j$, i.e., the j-th column of $\boldsymbol{X}$, is

$$\boldsymbol{x}_j \ = \ \sum_{i=1}^{p} \boldsymbol{\lambda}_i \ z_{ij} \ + \ \boldsymbol{\epsilon}_j \ = \ \boldsymbol{\Lambda} \ \tilde{\boldsymbol{z}}_j \ + \ \boldsymbol{\epsilon}_j \ , \qquad (2)$$

where $\boldsymbol{\epsilon}_j$ is the j-th column of the noise matrix $\boldsymbol{\Upsilon}$ and $\tilde{\boldsymbol{z}}_j = (z_{1j}, \ldots, z_{pj})^T$ denotes the j-th column of the matrix $\boldsymbol{Z}$. Recall that $\boldsymbol{z}_i^T = (z_{i1}, \ldots, z_{il})$ is the vector of values that constitutes the i-th bicluster (one value per sample), while $\tilde{\boldsymbol{z}}_j$ is the vector of values that contribute to the j-th sample (one value per bicluster).

The formulation in Eq. (2) facilitates a generative interpretation by a factor analysis model with p factors

$$\boldsymbol{x} \ = \ \sum_{i=1}^{p} \boldsymbol{\lambda}_i \ \tilde{z}_i \ + \ \boldsymbol{\epsilon} \ = \ \boldsymbol{\Lambda} \ \tilde{\boldsymbol{z}} \ + \ \boldsymbol{\epsilon} \ , \qquad (3)$$

where $\boldsymbol{x}$ are the observations, $\boldsymbol{\Lambda}$ is the loading matrix, $\tilde{z}_i$ is the value of the i-th factor, $\tilde{\boldsymbol{z}} = (\tilde{z}_1, \ldots, \tilde{z}_p)^T$ is the vector of factors, and $\boldsymbol{\epsilon} \in \mathbb{R}^n$ is the additive noise. Standard factor analysis assumes that the noise is independent of $\tilde{\boldsymbol{z}}$, that $\tilde{\boldsymbol{z}}$ is $\mathcal{N}(\boldsymbol{0}, \boldsymbol{I})$-distributed, and that $\boldsymbol{\epsilon}$ is $\mathcal{N}(\boldsymbol{0}, \boldsymbol{\Psi})$-distributed, where the covariance matrix $\boldsymbol{\Psi} \in \mathbb{R}^{n \times n}$ is a diagonal matrix expressing independent Gaussian noise. The parameter $\boldsymbol{\Lambda}$ explains the dependent (common) and $\boldsymbol{\Psi}$ the independent variance in the observations $\boldsymbol{x}$. Normality of the additive noise in gene expression is justified by the findings in (Hochreiter *et al.*, 2006).

The unity matrix as covariance matrix for $\tilde{\boldsymbol{z}}$ may be violated by overlapping biclusters, however we want to avoid to divide a real bicluster into two factors. Thus, we prefer uncorrelated factors over more sparseness. The factors can be decorrelated by setting $\hat{\boldsymbol{z}} := \boldsymbol{A}^{-1} \ \tilde{\boldsymbol{z}}$ and $\hat{\boldsymbol{\Lambda}} := \boldsymbol{\Lambda} \ \boldsymbol{A}$ with the symmetric invertible matrix $\boldsymbol{A}^2 = \mathrm{E}\big(\tilde{\boldsymbol{z}} \ \tilde{\boldsymbol{z}}^T\big)$:

$$\boldsymbol{\Lambda} \ \boldsymbol{z} \ = \ \boldsymbol{\Lambda} \ \boldsymbol{A} \ \boldsymbol{A}^{-1} \ \boldsymbol{z} \ = \ \hat{\boldsymbol{\Lambda}} \ \hat{\boldsymbol{z}} \quad \text{and}$$

$$\mathrm{E}\big(\hat{\boldsymbol{z}} \ \hat{\boldsymbol{z}}^T\big) \ = \ \boldsymbol{A}^{-1} \ \mathrm{E}\big(\tilde{\boldsymbol{z}} \ \tilde{\boldsymbol{z}}^T\big) \ \boldsymbol{A}^{-1} \ = \ \boldsymbol{A}^{-1} \boldsymbol{A}^2 \boldsymbol{A}^{-1} \ = \boldsymbol{I} \ .$$

Standard factor analysis does not consider sparse factors and sparse loadings which are essential in our formulation to represent biclusters. Sparseness is obtained by a component-wise independent *Laplace* distribution (Hyvärinen and Oja, 1999), which is now used as a prior on the factors $\tilde{\boldsymbol{z}}$ instead of the Gaussian:

$$p(\tilde{\boldsymbol{z}}) \ = \ \left(\tfrac{1}{\sqrt{2}}\right)^p \prod_{i=1}^{p} e^{- \ \sqrt{2} \ |\tilde{z}_i|}$$

Sparse loadings $\boldsymbol{\lambda}_i$ and, therefore sparse $\boldsymbol{\Lambda}$, are achieved by two alternative strategies. In the first model, called FABIA, we assume a component-wise independent *Laplace* prior for the loadings (like for the factors):

$$p(\boldsymbol{\lambda}_i) \ = \ \left(\tfrac{1}{\sqrt{2}}\right)^n \prod_{k=1}^{n} e^{- \ \sqrt{2} \ |\lambda_{ki}|} \qquad (4)$$

The FABIA model contains the product of Laplacian variables which is distributed proportionally to the 0-th order modified Bessel function of the second kind (Bithas *et al.*, 2007). For large values, this Bessel function is a negative exponential function of the square root of the random variable. Therefore, the tails of the distribution are heavier than those of the Laplace distribution. The Gaussian noise, however, reduces the heaviness of the tails such that the heaviness is between Gaussian and Bessel function tails — about as heavy as the tails of the Laplacian distribution. These *heavy tails* are exactly the desired model characteristics.

The second model, called FABIAS, uses a prior distribution for the loadings that is nonzero only in regions where the loadings are sparse. Following (Hoyer, 2004), we define sparseness as

$$\mathrm{sp}(\boldsymbol{\lambda}_i) \ = \ \frac{\sqrt{n} \ - \ \sum_{k=1}^{n} |\lambda_{ki}| \ / \ \sum_{k=1}^{n} \lambda_{ki}^2}{\sqrt{n} \ - \ 1}$$

leading to the prior with parameter spL

$$p(\boldsymbol{\lambda}_i) \ = \ \begin{cases} c & \text{for} \quad \mathrm{sp}(\boldsymbol{\lambda}_i) \ \leq \ \mathrm{spL} \\ 0 & \text{for} \quad \mathrm{sp}(\boldsymbol{\lambda}_i) \ > \ \mathrm{spL} \end{cases} . \qquad (5)$$

Relation to Independent Component Analysis. The FABIA and the FABIAS models are closely related to Independent Component Analysis (ICA; Comon, 1994; Bell and Sejnowski, 1995; Hyvärinen, 1999). ICA searches for a matrix factorization, where the components of $\tilde{\boldsymbol{z}}$ in model Eq. (3) without noise $\boldsymbol{\epsilon}$ should be statistically independent from each other. The matrix decomposition for ICA is

$$\boldsymbol{X} \ = \ \boldsymbol{\Lambda}_{\mathrm{ICA}} \ \boldsymbol{Z}_{\mathrm{ICA}}, \ \text{where} \ \ \boldsymbol{Z}_{\mathrm{ICA}} \ \boldsymbol{Z}_{\mathrm{ICA}}^T \ = \ \boldsymbol{I} \ .$$

If super-Gaussian priors (e.g. Laplacian) are assumed, contrast functions like the kurtosis of the components of $\boldsymbol{z}_{\mathrm{ICA}}$ are maximized for a given variance and sparse representations are obtained. Thus only $\boldsymbol{Z}_{\mathrm{ICA}}$ is sparse, but not $\boldsymbol{\Lambda}_{\mathrm{ICA}}$ as in FABIA and FABIAS.

3 MODEL SELECTION

The free parameters $\boldsymbol{\Lambda}$ and $\boldsymbol{\Psi}$ can be estimated by Expectation-Maximization (EM; Dempster *et al.*, 1977). With a prior probability on the loadings, the a posteriori of the parameters is maximized like in (Hochreiter *et al.*, 2006; Talloen *et al.*, 2007).

3.1 Variational Approach for Sparse Factors

Model selection is not straightforward because the likelihood

$$p(\boldsymbol{x} \mid \boldsymbol{\Lambda}, \boldsymbol{\Psi}) \ = \ \int p(\boldsymbol{x} \mid \tilde{\boldsymbol{z}}, \boldsymbol{\Lambda}, \boldsymbol{\Psi}) \ p(\tilde{\boldsymbol{z}}) \ d\tilde{\boldsymbol{z}}$$

cannot be computed analytically for a Laplacian prior $p(\tilde{\boldsymbol{z}})$. We employ a variational approach according to Girolami (2001) and Palmer *et al.* (2006) for model selection. They introduce a model family that is parametrized by

$\boldsymbol{\xi}$, where the maximum over models in this family is the true likelihood:

$$\arg\max_{\boldsymbol{\xi}} p(\boldsymbol{x}|\boldsymbol{\xi}) \ = \ \log p(\boldsymbol{x}) \ .$$

Using an EM algorithm, not only the likelihood with respect to the parameters $\boldsymbol{\Lambda}$ and $\boldsymbol{\Psi}$ is maximized, but also with respect to $\boldsymbol{\xi}$.

In the following, $\boldsymbol{\Lambda}$ and $\boldsymbol{\Psi}$ denote the actual parameter estimates. According to Girolami (2001) and Palmer *et al.* (2006), we obtain

$$\begin{aligned} \mathrm{E}\big(\tilde{\boldsymbol{z}}_j \mid \boldsymbol{x}_j\big) \ &= \big(\boldsymbol{\Lambda}^T \, \boldsymbol{\Psi}^{-1} \, \boldsymbol{\Lambda} \ + \ \Xi_j^{-1}\big)^{-1} \, \boldsymbol{\Lambda}^T \, \boldsymbol{\Psi}^{-1} \, \boldsymbol{x}_j \quad \text{and} \\ \mathrm{E}\big(\tilde{\boldsymbol{z}}_j \, \tilde{\boldsymbol{z}}_j^T \mid \boldsymbol{x}_j\big) \ &= \big(\boldsymbol{\Lambda}^T \, \boldsymbol{\Psi}^{-1} \, \boldsymbol{\Lambda} \ + \ \Xi_j^{-1}\big)^{-1} + \\ & \quad \mathrm{E}\big(\tilde{\boldsymbol{z}}_j \mid \boldsymbol{x}_j\big) \, \mathrm{E}\big(\tilde{\boldsymbol{z}}_j \mid \boldsymbol{x}_j\big)^T \ , \end{aligned}$$

where Ξ_j stands for $\mathrm{diag}\,(\boldsymbol{\xi}_j)$. The update for $\boldsymbol{\xi}_j$ is

$$\boldsymbol{\xi}_j \ = \ \mathrm{diag}\left(\sqrt{\mathrm{E}(\tilde{\boldsymbol{z}}_j \, \tilde{\boldsymbol{z}}_j^T \mid \boldsymbol{x}_j)}\right) \ .$$

3.2 New Update Rules for Sparse Loadings

The update rules for FABIA (Laplace prior on loadings) are

$$\boldsymbol{\Lambda}^{\mathrm{new}} \ = \ \frac{\frac{1}{l} \, \sum_{j=1}^{l} \boldsymbol{x}_j \, \mathrm{E}(\tilde{\boldsymbol{z}}_j \mid \boldsymbol{x}_j)^T \ - \ \frac{\alpha}{l} \, \boldsymbol{\Psi} \, \mathrm{sign}(\boldsymbol{\Lambda})}{\frac{1}{l} \sum_{j=1}^{l} \mathrm{E}(\tilde{\boldsymbol{z}}_j \, \tilde{\boldsymbol{z}}_j^T \mid \boldsymbol{x}_j)} \tag{6}$$

$$\mathrm{diag}\,(\boldsymbol{\Psi}^{\mathrm{new}}) \ = \ \boldsymbol{\Psi}^{\mathrm{EM}} \ + \ \mathrm{diag}\Big(\frac{\alpha}{l} \, \boldsymbol{\Psi} \, \mathrm{sign}(\boldsymbol{\Lambda})(\boldsymbol{\Lambda}^{\mathrm{new}})^T\Big)$$

where

$$\boldsymbol{\Psi}^{\mathrm{EM}} \ = \ \mathrm{diag}\Big(\frac{1}{l} \sum_{j=1}^{l} \boldsymbol{x}_j \boldsymbol{x}_j^T \ - \ \boldsymbol{\Lambda}^{\mathrm{new}} \frac{1}{l} \sum_{j=1}^{l} \mathrm{E}\,(\tilde{\boldsymbol{z}}_j \mid \boldsymbol{x}_j) \ \boldsymbol{x}_j^T\Big).$$

The update rules for FABIAS must take into account that each $\boldsymbol{\lambda}_i$ from $\boldsymbol{\Lambda}$ has a prior with restricted support. Therefore the sparseness constraints $\mathrm{sp}(\boldsymbol{\lambda}_i) \ \leq \ \mathrm{spL}$ from Eq. (5) hold. These constraints are ensured by a projection of $\boldsymbol{\lambda}_i$ after each $\boldsymbol{\Lambda}$ update according to Hoyer (2004). The projection is a convex quadratic problem which minimizes the Euclidean distance to the original vector subject to the constraints. The projection problem can be solved fast by an iterative procedure where the l_2-norm of the vectors is fixed to 1. We update $\mathrm{diag}(\boldsymbol{\Psi}^{\mathrm{new}}) = \boldsymbol{\Psi}^{\mathrm{EM}}$ and project each updated prototype vector to a sparse vector with sparseness spL giving the overall projection:

$$\boldsymbol{\Lambda}^{\mathrm{new}} \ = \ \mathrm{proj}\left(\frac{\frac{1}{l} \, \sum_{j=1}^{l} \boldsymbol{x}_j \, \mathrm{E}\,(\tilde{\boldsymbol{z}}_j \mid \boldsymbol{x}_j)^T}{\frac{1}{l} \sum_{j=1}^{l} \mathrm{E}(\tilde{\boldsymbol{z}}_j \, \tilde{\boldsymbol{z}}_j^T \mid \boldsymbol{x}_j)}, \mathrm{spL}\right)$$

3.3 Extremely Sparse Priors

Some gene expression data sets are sparser than Laplacian. For example, during estimating DNA copy numbers with Affymetrix SNP 6 arrays, we observed a kurtosis larger than 30 (FABIA results shown at http://www.bioinf.jku.at/software/fabia/fabia.html). We want to adapt our model class to deal with such sparse data sets. Toward this end, we define extremely sparse priors both on the factors and the loadings utilizing the following (pseudo) distributions:

Generalized Gaussians:	$p(z) \ \propto \ \exp\big(- \, \lvert z \rvert^{\beta}\big)$
Jeffrey's prior:	$p(z) \ \propto \ \exp(- \, \ln \lvert z \rvert) \ = \ 1/\lvert z \rvert$
Improper prior:	$p(z) \ \propto \ \exp\big(\lvert z \rvert^{-\beta}\big)$

For the first distribution, we assume $0 < \beta \leq 1$ and for the third $0 < \beta$. Note, the third distribution may only exist on the interval $[\epsilon, a]$ with $0 < \epsilon < a$. We assume that ϵ is sufficiently small.

For the *loadings*, we need the derivatives of the negative log-distributions for optimizing the log-posterior. These derivatives are proportional to $\lvert z \rvert^{-\mathrm{spl}}$, where $\mathrm{spl} \ = \ 0$ corresponds to the Laplace prior and $\mathrm{spl} \ > \ 0$ to sparser priors. The update rule is as in Eq. (6), where $\mathrm{sign}(\boldsymbol{\Lambda})$ is replaced by $\lvert\boldsymbol{\Lambda}\rvert^{-\mathrm{spl}} \ \mathrm{sign}(\boldsymbol{\Lambda})$ with element-wise operations (absolute value, sign, exponentiation, multiplication).

For the *factors*, we represent the priors through a convex variational form according to Palmer *et al.* (2006). That is possible because $g(z) \ = \ -\ln p(\sqrt{z})$ is increasing and concave for $z \ > \ 0$ (first order derivatives are larger and second order smaller than zero). According to Palmer *et al.* (2006), the update for $\boldsymbol{\xi}_j$ is

$$\boldsymbol{\xi}_j \ \propto \ \mathrm{diag}\Big(\mathrm{E}\big(\tilde{\boldsymbol{z}}_j \, \tilde{\boldsymbol{z}}_j^T \mid \boldsymbol{x}_j\big)^{\mathrm{spz}}\Big)$$

for all $\mathrm{spz} \geq 1/2$, where $\mathrm{spz} = 1/2$ ($\beta = 1$) represents the Laplace prior and $\mathrm{spz} > 1/2$ leads to sparser priors.

3.4 Data Preprocessing and Initialization

The data $\boldsymbol{x}$ may be centered either to zero mean or to zero median which we prefer to obtain sparser raw data. Then the data should be scaled to unit second moments to allow initialization of the parameters in the same range. See the supplementary for justification of these preprocessing steps.

The iterative model selection procedure requires initialization of the parameters $\boldsymbol{\Lambda}$, $\boldsymbol{\Psi}$, and $\boldsymbol{\xi}_j$. The simplest strategy is to initialize $\boldsymbol{\Lambda}$ randomly while ensuring that $\boldsymbol{\Psi} \ = \ \mathrm{diag}\Big(\mathrm{covar}(\boldsymbol{x}) \ - \ \boldsymbol{\Lambda}\boldsymbol{\Lambda}^T\Big) \ \geq \ \delta \ > \ 0$. The variational parameter vectors $\boldsymbol{\xi}_j$ are initialized as vectors of ones. An alternative initialization strategy can be based on ICA. The ICA solution supplies factors $\boldsymbol{Z}_{\mathrm{ICA}}$ that are sparse and decorrelated.

4 INFORMATION CONTENT OF BICLUSTERS

A highly desired property for biclustering algorithms is the ability to rank the extracted biclusters analogously to principal components which are ranked according to the data variance they explain. We rank biclusters according to the information they contain about the data. The information content of $\tilde{\boldsymbol{z}}_j$ for the j-th observation $\boldsymbol{x}_j$ is the mutual information between $\tilde{\boldsymbol{z}}_j$ and $\boldsymbol{x}_j$:

$$\mathrm{I}(\boldsymbol{x}_j; \tilde{\boldsymbol{z}}_j) \ = \ \mathrm{H}(\tilde{\boldsymbol{z}}_j) \ - \ \mathrm{H}(\tilde{\boldsymbol{z}}_j \mid \boldsymbol{x}_j) \ = \frac{1}{2} \ \ln \big\lvert \boldsymbol{I}_p \ + \ \Xi_j \ \boldsymbol{\Lambda}^T \ \boldsymbol{\Psi}^{-1} \ \boldsymbol{\Lambda} \big\rvert$$

The independence of $\boldsymbol{x}_j$ and $\tilde{\boldsymbol{z}}_j$ across j gives

$$\mathrm{I}(\boldsymbol{X}; \boldsymbol{Z}) \ = \ \frac{1}{2} \ \sum_{j=1}^{l} \ln \big\lvert \boldsymbol{I}_p \ + \ \Xi_j \ \boldsymbol{\Lambda}^T \ \boldsymbol{\Psi}^{-1} \ \boldsymbol{\Lambda} \big\rvert \ .$$

For the FARMS summarization algorithm ($p \ = \ 1$ and $\Xi_j \ = \ 1$), this information is the negative logarithm of the I/NI call (Talloen *et al.*, 2007).

To assess the information content of one factor, we consider the case that factor $\tilde{z}_i$ is removed from the final model. This corresponds to setting $\xi_{ij} = 0$ (by ξ_{ij}, we denote the i-th entry in $\boldsymbol{\xi}_j$) and therefore the explained covariance $\xi_{ji} \ \boldsymbol{\lambda}_i \ \boldsymbol{\lambda}_i^T$ is removed:

$$\boldsymbol{x}_j \mid (\tilde{\boldsymbol{z}}_j \setminus z_{ij}) \ \sim \ \mathcal{N}\big(\boldsymbol{\Lambda} \, \tilde{\boldsymbol{z}}_j|_{z_{ij}=0} \ , \ \boldsymbol{\Psi} \ + \ \xi_{ij} \ \boldsymbol{\lambda}_i \ \boldsymbol{\lambda}_i^T\big)$$

The information of z_{ij} given the other factors is

$$\begin{aligned} \mathrm{I}(\boldsymbol{x}_j; z_{ij} \mid (\tilde{\boldsymbol{z}}_j \setminus z_{ij})) \ &= \ \mathrm{H}(z_{ij} \mid (\tilde{\boldsymbol{z}}_j \setminus z_{ij})) - \mathrm{H}(z_{ij} \mid (\tilde{\boldsymbol{z}}_j \setminus z_{ij}), \boldsymbol{x}_j) \\ &= \ \frac{1}{2} \ \ln \big(1 \ + \ \xi_{ij} \ \boldsymbol{\lambda}_i^T \boldsymbol{\Psi}^{-1} \boldsymbol{\lambda}_i\big) \ . \end{aligned}$$

Again independence across j gives

$$\mathrm{I}(\boldsymbol{X}; \boldsymbol{z}_i^T \mid (\boldsymbol{Z} \setminus \boldsymbol{z}_i^T)) \ = \ \frac{1}{2} \ \sum_{j=1}^{l} \ \ln \Big(1 \ + \ \xi_{ij} \ \boldsymbol{\lambda}_i^T \boldsymbol{\Psi}^{-1} \boldsymbol{\lambda}_i\Big) \ .$$

This information content gives that part of information in $\boldsymbol{x}$ that $\boldsymbol{z}_i^T$ conveys across all examples. Note that also the number of nonzero $\boldsymbol{\lambda}_i$'s (size of the bicluster) enters into the information content.

5 EXTRACTING MEMBERS OF BICLUSTERS

After model selection in Section 3 and ranking of bicluster in Section 4, the i-th bicluster has soft gene memberships given by the absolute values of $\boldsymbol{\lambda}_i$ and soft sample memberships given by the absolute values of $\boldsymbol{z}_i^T$.

However, applications may need hard memberships. We determine the members of bicluster i by selecting absolute values λ_{ki} and z_{ij} above thresholds thresL and thresZ, respectively.

First, the second moment of each factor is normalized to 1 resulting in a factor matrix $\hat{\boldsymbol{Z}}$ (in accordance with $\mathrm{E}(\tilde{\boldsymbol{z}}\tilde{\boldsymbol{z}}^T) = \boldsymbol{I}$). Consequently, $\boldsymbol{\Lambda}$ is rescaled to $\hat{\boldsymbol{\Lambda}}$ such that $\boldsymbol{\Lambda}\boldsymbol{Z} = \hat{\boldsymbol{\Lambda}}\hat{\boldsymbol{Z}}$. Now the threshold thresZ can be chosen to determine which percentage of samples will on average belong to a bicluster. For a Laplace prior, this percentage can be computed by $\frac{1}{2}\exp(-\sqrt{2}/\mathrm{thresZ})$.

In the default setting, for each factor $\hat{\boldsymbol{z}}_i$, only one bicluster is extracted. In gene expression, an expression pattern is either absent or present but not negatively present. Therefore, the i-th bicluster is either determined by the positive or negative values of $\hat{z}_{ij}$. Which one of these two possibilities is chosen is decided by whether the sum over $\left|\hat{z}_{ij}\right| > \mathrm{thresZ}$ is larger for the positive or negative $\hat{z}_{ij}$.

The threshold thresL for the loadings is more difficult to determine, because normalization would lead to a rescaling of the already normalized factors. Since biclusters may overlap, the contribution of $\lambda_{ki} z_{ij}$ that are relevant must be estimated. Therefore, we first estimate the standard deviation of $\boldsymbol{\Lambda}\boldsymbol{Z}$ by

$$\mathrm{sdLZ} \;=\; \sqrt{\frac{1}{p\;l\;n} \sum_{(i,j,k)=(1,1,1)}^{(p,l,n)} \left(\hat{\lambda}_{ki}\;\hat{z}_{ij}\right)^2}\;.$$

We set this standard deviation to the product of both thresholds which is solved for thresL: $\mathrm{thresL} \;=\; \mathrm{sdLZ}\;/\;\mathrm{thresZ}$. However, an optimal thresL depends on the sparseness parameters and on the characteristics of the biclustering problem.

6 EXPERIMENTS

6.1 Evaluating Biclustering Results

We introduce a novel procedure for comparing two sets of biclusters, where a bicluster is a set of matrix elements. Previous comparison measures like the measures in (Gu and Liu, 2008) do not take into account that one element may belong to more than one bicluster. Another aspect is that missing a whole, but small, bicluster can be more serious than missing the same number of elements in a larger bicluster, because incomplete biclusters can be extended in a post-processing step or by supervised learning.

We compute the similarity of bicluster sets as follows:

(1) Compute similarity index of all pairs of biclusters, where one is from the first set and the other from the second set;
(2) Assign the biclusters of one set to biclusters of the other set by maximizing the assignment through the Munkres algorithm (Munkres, 1957);
(3) Divide the sum of similarities of the assigned biclusters by the number of biclusters of the larger set.

Step (3) is essential to ensure that sets with a single bicluster and sets with all possible biclusters do not obtain the maximal score. Note, that the same procedure can analogously be used to compare results of ordinary clustering results.

It remains to define the similarity of two biclusters. In (Boyce and Ellison, 2001), different similarity indices for sets have been compared. We choose 4 out of the best 5 indices and excluded the Baroni-Urbani & Buser index. It also uses zero-zero matches which is inappropriate in gene expression analysis where only differentially expressed genes are of interest. For two biclusters $\mathcal{A}$ and $\mathcal{B}$, we define a as number of elements that are both in bicluster $\mathcal{A}$ and in bicluster $\mathcal{B}$ (joint occurrences), b as number of elements in bicluster $\mathcal{B}$, but not in bicluster $\mathcal{A}$, and c as number of elements in bicluster $\mathcal{A}$, but not in bicluster $\mathcal{B}$. We use the following similarity indices for sets:

Jaccard index (“ja”):	$\frac{a}{a\,+\,b\,+\,c}$
Kulczynski index (“ku”):	$\frac{1}{2}\left(\frac{a}{a\,+\,b} \;+\; \frac{a}{a\,+\,c}\right)$
Ochiai index (“oc”):	$\frac{a}{\sqrt{(a\,+\,b)\;(a\,+\,c)}}$
Sørensen index (“so”):	$\frac{2\;a}{2\;a\,+\,b\,+\,c}$

The following holds for all four indices: the higher the similarity, the higher the value. The highest value is 1 and it is only obtained for two identical sets.

6.2 Compared Methods

We compare the following 13 biclustering methods:

(1) FABIA: our new method with sparse prior Eq. (4)
(2) FABIAS: our new method with sparseness projection Eq. (5)
(3) MFSC: matrix factorization with sparseness constraints (Hoyer, 2004)
(4) plaid: plaid model (Lazzeroni and Owen, 2002)
(5) ISA: iterative signature algorithm (Ihmels *et al.*, 2004)
(6) OPSM: order-preserving sub-matrices (Ben-Dor *et al.*, 2003)
(7) SAMBA: statistical-algorithmic method for bicluster analysis (Tanay *et al.*, 2002)
(8) xMOTIF: conserved motifs (Murali and Kasif, 2003)
(9) Bimax: divide-and-conquer algorithm (Prelic *et al.*, 2006)
(10) CC: Cheng-Church δ-biclusters (Cheng and Church, 2000)
(11) plaid_t: improved plaid model (Turner *et al.*, 2003)
(12) FLOC: flexible overlapped biclustering, a generalization of Cheng-Church δ-biclusters (Yang *et al.*, 2005)
(13) spec: spectral biclustering (Kluger *et al.*, 2003)

For evaluating the methods, we used: for (1)–(3) our R package `fabia`, for (4) the authors' software[1], for (5) and (6) the software BicAT (Barkow *et al.*, 2006), for (7) the software EXPANDER (Shamir *et al.*, 2005), for (8)–(13) the R package `biclust` (Kaiser and Leisch, 2008).

In all experiments, rows (genes) were standardized to mean 0 and variance 1. For fair comparison, the parameters of the methods were optimized on additional toy data sets. If more than one setting was close to the optimum, all near optimal parameter settings were tested. In the following, these variants are denoted as *method_variant* (e.g. plaid_ss). A complete list of all settings and variants is available in the supplementary.

6.3 Simulated Data Sets with Known Biclusters

Benchmark data sets published in (Prelic *et al.*, 2006) and (Li *et al.*, 2009) are small (50 to 100 genes), have low noise, have equally sized biclusters, and have only simultaneous row and column overlaps. FABIA performed very well on these data sets (see supplementary S6.3.1 and S6.3.2). However, we use more realistic simulated data sets as shown in supplementary S6.3, where Fig. S8, S9 and S10 show density and moments for real gene expression data and S7 for our simulated data. Our simulated data match the gene expression data better especially by the heavy tails. We assumed to have $n = 1000$ genes and $l = 100$ samples. We implanted $p = 10$ multiplicative biclusters with the model given by Eq. (1).

[1] http://www-stat.stanford.edu/~owen/plaid/

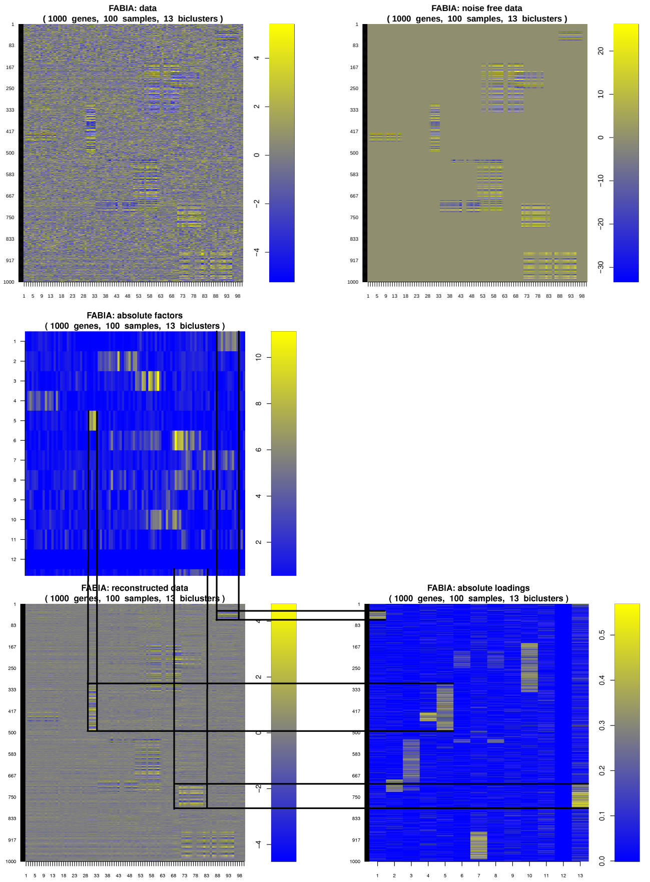

Fig. 2. An example of FABIA model selection. The data have 10 true biclusters. We have trained the model with 13 biclusters. Only for visualization purposes, the biclusters are generated as contiguous blocks. Top: data (left) and noise-free data (right). Middle: factors $\boldsymbol{Z}$. Bottom: data reconstructed by the FABIA model as $\boldsymbol{\Lambda}\ \boldsymbol{Z}$ (left) and loadings (right). The lines indicate three biclusters and connect each bicluster in the reconstructed data with its corresponding factors (middle) and loadings (bottom right).

Table 1. A: Results on the 100 simulated data sets. Average similarity scores to the true biclusters as defined in Subsection 6.1 (standard deviation in brackets). Best results are printed bold and second best in italics ("better" means significantly better according to both a paired t-test and a McNemar test of correct elements in biclusters). **B:** The last two rows show the p-values of a two-sided Spearman rank correlation test on (i) the information content and (ii) the similarity to true biclusters.

A: Scores for finding true biclusters

method	ja	ku	oc	so
FABIA	*0.478*(1e-2)	*0.574*(1e-2)	*0.568*(1e-2)	*0.564*(1e-2)
FABIAS	**0.564**(3e-3)	**0.676**(3e-3)	**0.669**(3e-3)	**0.662**(3e-3)
MFSC	0.057(2e-3)	0.113(3e-3)	0.106(3e-3)	0.100(3e-3)
plaid_ss	0.045(9e-4)	0.195(2e-4)	0.119(1e-3)	0.081(2e-3)
plaid_ms	0.072(4e-4)	0.169(9e-4)	0.141(5e-4)	0.124(5e-4)
plaid_ms_5	0.083(6e-4)	0.195(2e-3)	0.165(1e-3)	0.144(9e-4)
ISA_1	0.046(8e-5)	0.137(6e-5)	0.101(1e-5)	0.076(5e-5)
ISA_2	0.077(3e-3)	0.129(4e-3)	0.123(4e-3)	0.117(4e-3)
ISA_3	0.039(3e-3)	0.067(5e-3)	0.064(5e-3)	0.062(4e-3)
OPSM	0.012(1e-4)	0.061(1e-4)	0.033(8e-5)	0.023(2e-4)
SAMBA	0.006(5e-5)	0.025(9e-5)	0.017(9e-5)	0.012(1e-4)
xMOTIF	0.002(6e-5)	0.011(1e-4)	0.006(1e-4)	0.003(1e-4)
Bimax	0.004(2e-4)	0.018(9e-4)	0.011(5e-4)	0.007(3e-4)
CC	0.001(7e-6)	0.011(2e-4)	0.004(2e-5)	0.002(1e-5)
plaid_t_ab	0.046(5e-3)	0.167(1e-2)	0.111(1e-2)	0.078(9e-3)
plaid_t_a	0.037(4e-3)	0.173(9e-3)	0.100(8e-3)	0.064(6e-3)
FLOC	0.006(3e-5)	0.015(2e-5)	0.013(1e-5)	0.011(5e-5)
spec_1	0.032(5e-4)	0.085(2e-3)	0.068(1e-3)	0.057(1e-3)
spec_2	0.011(5e-4)	0.027(1e-3)	0.024(1e-3)	0.021(1e-3)

B: p-value of rank correlation between information and similarity score

method	ja	ku	oc	so
FABIA	1.7e-05	6.8e-08	2.4e-06	1.7e-05
FABIAS	6.1e-03	9.8e-04	1.5e-03	6.1e-03

The $\boldsymbol{\lambda}_i$'s are generated by (i) randomly choosing the number N_i^{λ} of genes in bicluster i from $\{10, \ldots, 210\}$, (ii) choosing N_i^{λ} genes randomly from $\{1, \ldots, 1000\}$, (iii) adding $\mathcal{N}(0, 0.2)$ noise to $\boldsymbol{\lambda}_i$ components that are not in bicluster i, and (iv) adding an $\mathcal{N}(\pm 3, 1)$ signal to $\boldsymbol{\lambda}_i$ components that are in bicluster i, where the sign is chosen randomly for each gene.

The $\boldsymbol{z}_i$'s are generated by (i) randomly choosing the number $N_i^{\boldsymbol{z}}$ of samples in bicluster i from $\{5, \ldots, 25\}$, (ii) choosing $N_i^{\boldsymbol{z}}$ samples randomly from $\{1, \ldots, 100\}$, (iii) adding $\mathcal{N}(0, 0.2)$ noise to $\boldsymbol{z}_i$ components that are not in bicluster i, and (iv) adding an $\mathcal{N}(2, 1)$ signal to $\boldsymbol{z}_i$ components that are in bicluster i.

Finally, we draw the $\boldsymbol{\Upsilon}$ entries (additive noise on all entries) according to $\mathcal{N}(0, 3)$ and compute the data $\boldsymbol{X}$ according to Eq. (1).

This data generation procedure is repeated independently 100 times to create 100 simulated data sets. Figure 2 visualizes a FABIA result on a simulated data set, where, in contrast to our 100 benchmark data sets, the biclusters have been created as contiguous blocks for visualization purposes. Table 1 shows the results for these 100 simulated data sets. The low p-values of a two-sided Spearman's rank correlation ρ test on information content and similarity to true biclusters demonstrate that true biclusters can indeed be identified by their information content (cf. Section 4). The methods are evaluated by the average similarity score to the true biclusters as defined in Subsection 6.1. Our new methods FABIA and FABIAS outperform all other methods considerably.

The other methods showed similar characteristics as observed by Gu and Liu (2008) for biclusters created by an additive model: ISA has problems with multiple overlapping clusters; SAMBA and OPSM excluded many relevant biclusters; SAMBA, Bimax, xMOTIF, CC, and FLOC found many small random biclusters (overfitting). MFSC extracted equally sized biclusters which did not reflect the true biclusters structure, but explained the data well; spec produces a partition of the samples for each gene set. The plaid models tend to find large overlapping clusters.

6.4 Gene Expression Data Sets

We consider three gene expression data sets which have been provided by the Broad Institute and were previously analyzed by Hoshida *et al.* (2007). They first clustered the samples using additional data sets and then confirmed the clusters by gene set enrichment analysis. Our goal was to study how well biclustering methods are able to re-identify these clusters without any additional information.

(A) The *"breast cancer" data set* (van't Veer *et al.*, 2002) was aimed at discovering a predictive gene signature for the outcome of a breast cancer therapy. We removed the outlier array S54 which leads to a data set with 97 samples and 1213 probe sets. After standardization, skewness was 0.45 and excess kurtosis 0.93. In (Hoshida *et al.*, 2007), three biologically meaningful sub-classes were found, where 50 out of 61 cases from class 1 and 2 were estrogen receptor positive and only 3 out of 36 from class 3.

(B) The *"multiple tissue types" data set* (Su *et al.*, 2002) are gene expression profiles from human and mouse samples across diverse tissues and cell lines aimed at constructing a reference for the mammalian transcriptome. The data set contains 102 samples with 5565 probe sets. After standardization, skewness was 0.15 and excess kurtosis 1.3. Biclustering should be able to re-identify the tissue types.

(C) The *"diffuse large-B-cell lymphoma (DLBCL)" data set* (Rosenwald *et al.*, 2002) was aimed at predicting the survival after chemotherapy. It consists of 180 samples and 661 probe sets, and after standardization the skewness was -0.05 and excess kurtosis 0.35. In (Hoshida *et al.*, 2007), three classes were found: *OxPhos* (oxidative phosphorylation), *BCR* (B-cell response), and *HR* (host response). These subclasses should be found by biclustering.

The biclustering results are summarized in Table 2. The performance was assessed by comparing known classes of samples in the data sets with the sample sets identified by biclustering as defined in Subsection 6.1, in this case on sample clusters instead of biclusters. For multiple tissue samples, the plaid models perform best and our methods FABIA and FABIAS are second best. Our methods found more biclusters than defined by the tissue types. Additional clusters may stem from the different organisms that are considered. For breast cancer and DLBCL data sets, our new methods FABIA and FABIAS detected the clusters most accurately.

The new methods FABIA and FABIAS have considerably fewer genes in their bicluster than the next best performing method, plaid.

6.5 Drug Design

In a drug design project, Affymetrix GeneChip HT HG-U133+ PM array plates with 96 samples (12 × 8) per plate were used to analyze the effect of different compounds on gene expression. The compounds are selected to be active on a cancer cell line and were tested in groups of three replicates.

Raw expression data were summarized with FARMS (Hochreiter *et al.*, 2006) and informative probe sets are selected by I/NI calls (Talloen *et al.*, 2007). The preprocessed data matrix was 1413×95 (one array was missing) with skewness of -0.39 and excess kurtosis larger than 3.0 (i.e. heavier tails than Laplace). We tested FABIA on this data set. Biclusters were extracted with $\mathrm{thresZ} = 1.5$ for an average cluster size of 5 to 6 for the Laplacian prior ($\frac{1}{2}\exp(-\sqrt{2}\,1.5) \approx 0.06$).

FABIA found four biclusters. The first bicluster consisted of two replicate sets (6 arrays), the second consisted of 5 replicate sets with one replicate missing (14 arrays). The third bicluster consisted of 3 replicate sets and an additional array (10 arrays). The fourth bicluster consisted of arrays located at the last column of the plate — corresponding to border arrays which dry out. In the meantime, this problem has been fixed by Affymetrix. That replicates are clustered together shows that our biclustering approach works correctly.

Table 2. Results on the (A) breast cancer, (B) multiple tissue samples, (C) diffuse large-B-cell lymphoma (DLBCL) data sets measured by the score from in Subsection 6.1. An *"nc"* entry means that the method did not converge for this data set. Best results are in bold and second best in italics (again "better" means significantly better according to a paired t-test).

	(A) breast cancer	(B) multiple tissues	(C) DLBCL
method	ja – ku – oc – so	ja – ku – oc – so	ja – ku – oc – so
FABIA	**0.52–0.69–0.67–0.65**	0.53–0.59–0.59–0.59	**0.37–0.48–0.48–0.47**
FABIAS	**0.52**–*0.67–0.65–0.64*	0.44–0.54–0.54–0.54	*0.35–0.46–0.46–0.45*
MFSC	0.17–0.29–0.27–0.26	0.31–0.44–0.44–0.44	0.18–0.29–0.28–0.28
plaid_ss	0.39–0.47–0.45–0.44	0.56–*0.66–0.65–0.64*	0.30–0.40–0.40–0.39
plaid_ms	0.39–0.47–0.46–0.44	0.50–0.63–0.62–0.60	0.28–0.38–0.38–0.38
plaid_ms_5	0.29–0.36–0.36–0.35	0.23–0.25–0.25–0.25	0.21–0.29–0.29–0.28
plaid_a_ss	0.37–0.46–0.44–0.43	**0.65–0.71–0.71–0.71**	0.28–0.40–0.39–0.38
plaid_a_ms	0.34–0.40–0.39–0.39	*0.58–0.65–0.65–0.65*	0.27–0.37–0.37–0.37
plaid_a_ms_5	0.16–0.18–0.18–0.18	0.20–0.20–0.20–0.20	0.18–0.26–0.26–0.25
ISA_1	0.01–0.02–0.02–0.01	*nc – nc – nc – nc*	0.01–0.03–0.02–0.01
ISA_2	0.05–0.07–0.07–0.07	*nc – nc – nc – nc*	0.03–0.05–0.05–0.05
ISA_3	0.02–0.03–0.03–0.03	*nc – nc – nc – nc*	0.03–0.05–0.05–0.05
OPSM	0.04–0.08–0.07–0.06	0.04–0.07–0.07–0.07	0.03–0.21–0.10–0.05
SAMBA_01	0.01–0.02–0.02–0.01	0.01–0.02–0.02–0.02	0.01–0.02–0.02–0.02
SAMBA_05	0.02–0.04–0.03–0.03	0.03–0.05–0.04–0.04	0.02–0.04–0.03–0.03
xMOTIF	0.07–0.19–0.15–0.12	0.11–0.29–0.23–0.18	0.05–0.19–0.12–0.08
Bimax	0.01–0.01–0.01–0.01	0.10–0.33–0.02–0.13	0.07–0.04–0.24–0.18
CC	0.11–0.24–0.22–0.19	*nc – nc – nc – nc*	0.05–0.18–0.13–0.09
plaid_t_ab	0.24–0.31–0.30–0.29	0.38–0.49–0.48–0.46	0.17–0.25–0.24–0.22
plaid_t_a	0.23–0.28–0.28–0.27	0.39–0.48–0.48–0.48	0.11–0.26–0.22–0.19
spec_1	0.12–0.17–0.16–0.15	0.37–0.48–0.47–0.47	0.05–0.07–0.07–0.06
spec_2	0.07–0.11–0.10–0.10	0.21–0.31–0.30–0.30	0.08–0.14–0.14–0.14
FLOC	0.04–0.18–0.11–0.07	*nc – nc – nc – nc*	0.03–0.19–0.10–0.05

The bicluster with highest information content (2 sets of replicates) extracted genes that are related to mitosis (GO gene set enrichment analysis gave $p < 10^{-13}$ for the hypergeometric test). Regulation of mitosis genes is biologically plausible as inhibiting cell division would be consistent with an active compound which did not kill the cells. The compounds of this bicluster are now under investigation by Johnson & Johnson Pharmaceuticals.

7 CONCLUSION

We have introduced a novel biclustering method called "Factor Analysis for Bicluster Acquisition" (FABIA) that is a generative multiplicative model that assumes realistic non-Gaussian signal distributions with heavy tails. The generative model allows to rank biclusters according to their information content. Model selection is performed by maximum a posteriori via an EM algorithm based on a variational approach.

On 100 simulated data sets with known true biclusters, FABIA clearly outperformed all 11 competing methods. On three gene expression data sets with previously verified sub-clusters, FABIA was once the second best and twice the best-performing method. Finally, FABIA has been successfully applied to drug design to find compounds with similar effects on gene expression.

ACKNOWLEDGMENT

Funding: This work has been funded by Janssen Pharmaceutica N.V. and the Belgium IWT project 80536.

REFERENCES

Barkow, S., Bleuler, S., Prelic, A., Zimmermann, P., and Zitzler, E. (2006). BicAT: A biclustering analysis toolbox. *Bioinformatics*, **22**(10), 1282–1283.

Bell, A. J. and Sejnowski, T. J. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, **7**(6), 1129–1159.

Ben-Dor, A., Chor, B., Karp, R., and Yakhini, Z. (2003). Discovering local structure in gene expression data: the order-preserving submatrix problem. *J. Comput. Biol.*, **10**(3–4), 373–384.

Bithas, P. S., Sagias, N. C., Tsiftsis, T. A., and Karagiannidis, G. K. (2007). Distributions involving correlated generalized gamma variables. In *Proc. Int. Conf. on Applied Stochastic Models and Data Analysis*, volume 12, Chania.

Boyce, R. L. and Ellison, P. C. (2001). Choosing the best similarity index when performing fuzzy set ordination on binary data. *J. Veg. Sci.*, **12**, 711–720.

Busygin, S., Jacobsen, G., and Kramer, E. (2002). Double conjugated clustering applied o leukemia microarray data. In *Proc. 2nd SIAM Int. Conf. on Data Mining / Workshop on Clustering High Dimensional Data*.

Caldas, J. and Kaski, S. (2008). Bayesian biclustering with the plaid model. In *Proc. IEEE Int. Workshop on Machine Learning for Signal Processing*, volume XVIII, pages 291–296.

Califano, A., Stolovitzky, G., and Tu, Y. (2000). Analysis of gene expression microarays for phenotype classification. In *Proc. Int. Conf. on Computational Molecular Biology*, pages 75–85.

Cheng, Y. and Church, G. M. (2000). Biclustering of expression data. In *Proc. Int. Conf. on Intelligent Systems for Molecular Biology*, volume 8, pages 93–103.

Comon, P. (1994). Independent component analysis – a new concept? *Signal Processing*, **36**(3), 287–314.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *J. R. Stat. Soc. B Met.*, **39**(1), 1–22.

Getoor, L., Friedman, N., Koller, D., and Taskar, B. (2002). Learning probabilistic models of link structure. *J. Mach. Learn. Res.*, **3**, 679–707.

Getz, G., Levine, E., and Domany, E. (2000). Coupled two-way clustering analysis of gene microarray data. *P. Natl. Acad. Sci. USA*, **97**(22), 12079–12084.

Girolami, M. (2001). A variational method for learning sparse and overcomplete representations. *Neural Comput.*, **13**(11), 2517–2532.

Gu, J. and Liu, J. S. (2008). Bayesian biclustering of gene expression data. *BMC Genomics*, **9**(Suppl. 1), S4.

Hardin, J. and Wilson, J. (2009). A note on oligonucleotide expression values not being normally distributed. *Biostatistics*, **10**(3), 446–450.

Hartigan, J. A. (1972). Direct clustering of a data matrix. *J. Am. Stat. Assoc.*, **67**(337), 123–129.

Hochreiter, S., Clevert, D.-A., and Obermayer, K. (2006). A new summarization method for Affymetrix probe level data. *Bioinformatics*, **22**(8), 943–949.

Hoshida, Y., Brunet, J.-P., Tamayo, P., Golub, T. R., and Mesirov, J. P. (2007). Subclass mapping: Identifying common subtypes in independent disease data sets. *PLoS ONE*, **2**(11), e1195.

Hoyer, P. O. (2004). Non-negative matrix factorization with sparseness constraints. *J. Mach. Learn. Res.*, **5**, 1457–1469.

Hyvärinen, A. (1999). Survey on independent component analysis. *Neural Computing Surveys*, **2**, 94–128.

Hyvärinen, A. and Oja, E. (1999). A fast fixed-point algorithm for independent component analysis. *Neural Comput.*, **9**(7), 1483–1492.

Ihmels, J., Bergmann, S., and Barkai, N. (2004). Defining transcription modules using large-scale gene expression data. *Bioinformatics*, **20**(13), 1993–2003.

Kaiser, S. and Leisch, F. (2008). A toolbox for bicluster analysis in R. In P. Brito, editor, *Compstat 2008 – Proceedings in Computational Statistics*, pages 201–208, Heidelberg. Physica Verlag.

Kluger, Y., Basri, R., Chang, J. T., and Gerstein, M. B. (2003). Spectral biclustering of microarray data: Coclustering genes and conditions. *Genome Res.*, **13**, 703–716.

Lazzeroni, L. and Owen, A. (2002). Plaid models for gene expression data. *Stat. Sinica*, **12**(1), 61–86.

Li, G., Ma, Q., Tang, H., Paterson, A. H., and Xu, Y. (2009). QUBIC: a qualitative biclustering algorithm for analyses of gene expression data. *Nucleic Acids Res.*, **37**(15), e101.

Madeira, S. C. and Oliveira, A. L. (2004). Biclustering algorithms for biological data analysis: A survey. *IEEE ACM T. Comput. Bi.*, **1**(2), 24–45.

Meeds, E., Ghahramani, Z., Neal, R. M., and Roweis, S. T. (2007). Modeling dyadic data with binary latent factors. In *Advances in Neural Information Processing Systems 19*, pages 977–984.

Munkres, J. (1957). Algorithms for the assignment and transportation problems. *J. Soc. Ind. Appl. Math.*, **5**(1), 32–38.

Murali, T. M. and Kasif, S. (2003). Extracting conserved gene expression motifs from gene expression data. In *Pac. Symp. Biocomputing*, pages 77–88.

Palmer, J., Wipf, D., Kreutz-Delgado, K., and Rao, B. (2006). Variational EM algorithms for non-Gaussian latent variable models. In *Advances in Neural Information Processing Systems 18*, pages 1059–1066.

Prelic, A., Bleuler, S., Zimmermann, P., Wille, A., Bühlmann, P., Gruissem, W., Hennig, L., Thiele, L., and Zitzler, E. (2006). A systematic comparison and evaluation of biclustering methods for gene expression data. *Bioinformatics*, **22**(9), 1122–1129.

Reiss, D. J., Baliga, N. S., and Bonneau, R. (2006). Integrated biclustering of heterogeneous genome-wide datasets for the inference of global regulatory networks. *BMC Bioinformatics*, **2**(7), 280–302.

Rosenwald, A., Wright, G., Chan, W. C., Connors, J. M., Campo, E., Fisher, R. I., Gascoyne, R. D., Muller-Hermelink, H. K., Smeland, E. B., Giltnane, J. M., Hurt, E. M., Zhao, H., Averett, L., Yang, L., Wilson, W. H., Jaffe, E. S., Simon, R., Klausner, R. D., Powell, J., Duffey, P. L., Longo, D. L., Greiner, T. C., Weisenburger, D. D., Sanger, W. G., Dave, B. J., Lynch, J. C., Vose, J., Armitage, J. O., Montserrat, E., L'opez-Guillermo, A., Grogan, T. M., Miller, T. P., LeBlanc, M., Ott, G., Kvaloy, S., Delabie, J., Holte, H., Krajci, P., Stokke, T., and Staudt, L. M. (2002). The use of molecular profiling to predict survival after chemotherapy for diffuse large-B-cell lymphoma. *New Engl. J. Med.*, **346**, 1937–1947.

Segal, E., Taskar, B., Gasch, A., Friedman, N., and Koller, D. (2003). Decomposing gene expression into cellular processes. In *Pac. Symp. Biocomputing*, pages 89–100.

Shamir, R., Maron-Katz, A., Tanay, A., Linhart, C., Steinfeld, I., Sharan, R., Shiloh, Y., and Elkon, R. (2005). EXPANDER – an integrative program suite for microarray data analysis. *BMC Bioinformatics*, **6**, 232.

Sheng, Q., Moreau, Y., and De Moor, B. (2003). Biclustering micrarray data by Gibbs sampling. *Bioinformatics*, **19**(Suppl. 2), ii196–ii205.

Su, A. I., Cooke, M. P., A.Ching, K., Hakak, Y., Walker, J. R., Wiltshire, T., Orth, A. P., Vega, R. G., Sapinoso, L. M., Moqrich, A., Patapoutian, A., Hampton, G. M., Schultz, P. G., and Hogenesch, J. B. (2002). Large-scale analysis of the human and mouse transcriptomes. *P. Natl. Acad. Sci. USA*, **99**(7), 4465–4470.

Talloen, W., Clevert, D.-A., Hochreiter, S., Amaratunga, D., Bijnens, L., Kass, S., and Göhlmann, H. W. H. (2007). I/NI-calls for the exclusion of non-informative genes: a highly effective feature filtering tool for microarray data. *Bioinformatics*, **23**(21), 2897–2902.

Tanay, A., Sharan, R., and Shamir, R. (2002). Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, **18**(Suppl. 1), S136–S144.

Tang, C., Zhang, L., Zhang, I., and Ramanathan, M. (2001). Interrelated two-way clustering: an unsupervised approach for gene expression data analysis. In *Pro. 2nd IEEE Int. Symp. on Bioinformatics and Bioengineering*, pages 41–48.

Tibshirani, R., Hastie, T., Eisen, M., Ross, D., Botstein, D., and Brown, P. (1999). Clustering methods for the analysis of DNA microarray data. Technical report, Dept. of Health Research and Policy, Dept. of Genetics and Dept. of Biochemestry, Stanford University.

Turner, H., Bailey, T., and Krzanowski, W. (2003). Improved biclustering of microarray data demonstrated through systematic performance tests. *Comput. Stat. Data An.*, **48**(2), 235–254.

Van den Bulcke, T. (2009). *Robust algorithms for inferring regulatory networks based on gene expression measurements and biological prior information*. Ph.D. thesis, Katholieke Universiteit Leuven.

van't Veer, L. J., Dai, H., van de Vijver, M. J., He, Y. D., Hart, A. A., Mao, M., Peterse, H. L., van der Kooy, K., Marton, M. J., Witteveen, A. T., Schreiber, G. J., Kerkhoven, R. M., Roberts, C., Linsley, P. S., Bernards, R., and Friend, S. H. (2002). Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, **415**, 530–536.

Wang, H., Wang, W., Yang, J., and Yu, P. S. (2002). Clustering by pattern similarity in large data sets. In *Proc. 2002 ACM SIGMOD Int. Conf. on Management of Data*, pages 394–405.

Yang, J., Wang, H., Wang, W., and Yu., P. S. (2005). An improved biclustering method for analyzing gene expression profiles. *Int. J. Artif. Intell. T.*, **14**(5), 771–790.