Publication 5

Sami Hanhijärvi and Aleksi Kallio. 2012. Biclustering gene expression data with minimum description length. Technical report. Espoo, Finland: Aalto University, School of Science, Department of Information and Computer Science. 38 pages. Aalto University publication series SCIENCE + TECHNOLOGY 9/2012. Aalto-ST-9/2012. ISBN 978-952-60-4590-0. ISSN 1799-490X.

Aalto University publication series
SCIENCE + **TECHNOLOGY** 9/2012

Biclustering Gene Expression Data with Minimum Description Length

Sami Hanhijärvi and Aleksi Kallio

Aalto University
School of Science
Department of Information and Computer Science
Algorithmic Data Analysis

Aalto University publication series
SCIENCE + **TECHNOLOGY** 9/2012

ISBN 978-952-60-4590-0 (pdf)
ISSN-L 1799-4896
ISSN 1799-490X (pdf)

Unigrafia Oy
Helsinki 2012

Finland

Author
Sami Hanhijärvi and Aleksi Kallio

Name of the publication
Biclustering Gene Expression Data with Minimum Description Length

Publisher School of Science

Unit Department of Information and Computer Science

Series Aalto University publication series SCIENCE + TECHNOLOGY 9/2012

Field of research Data mining

Abstract

A bicluster in gene expression data consists of a subset of the genes and a subset of the experimental conditions, such that the genes in the bicluster behave in some sense similarly in the experimental conditions of the bicluster. In biclustering of gene expression data the goal is to find a collection of biclusters from the data.

We propose a novel biclustering algorithm, BiMDL, that finds biclusters of strong local correlation in the data matrix. The model is defined in a parameter-free way by using the Minimum Description Length (MDL) principle, and our algorithm has only few parameters. The algorithm is extended such that the biclusters express approximately disjoint regions in the data matrix.

BiMDL is compared to existing biclustering methods and shown to perform equally good or better in all the scenarios. The results suggest that the method is able to locate meaningful biological structures in the data matrix.

Keywords biclustering, minimum description length, gene expression

ISBN (printed) **ISBN (pdf)** 978-952-60-4590-0

ISSN-L 1799-4896 **ISSN (printed)** 1799-4896 **ISSN (pdf)** 1799-490X

Location of publisher Espoo **Location of printing** Espoo **Year** 2012

Pages 38

Contents

1. Introduction

Gene expression measurements produce matrices with expression levels for large amount of genes in different experimental conditions. Bidirectional hierarchical clustering has been the *de facto* method for analyzing the global structure of the data matrix. As the number of genes is large, often the actual target of the study is to identify subsets of genes and experimental conditions with coherent and exceptional values, i.e., to retrieve interesting local structures. Biclustering methods have been developed for this task.

Biclustering methods find pairs of row groups (e.g., genes) and column groups (e.g., experimental conditions) of the data matrix such that the rows in the row groups have in some sense a similar behavior in the columns of the column groups. For example, a bicluster may contain nearly constant values, or the values on each row may evolve similarly across columns. A classification of bicluster structures and their orientation in the data matrix was proposed by Madeira and Oliveira (2004), who also review most of the currently available biclustering methods. A more recent list of methods is presented by Bhattacharya and De (2009).

A bicluster that represents a biological process contains the genes that are related to that process and are regulated together, and the experimental conditions where the regulation mechanisms are active. The expression levels within such a bicluster are typically highly correlated between genes and also between experimental conditions. Therefore, the bicluster can be characterized by two linear components: one for the rows and one for the columns. In this paper, we describe a biclustering method that finds biclusters that have this structure, so that we can identify biological processes from the gene expression data.

Similar structures for biclusters have been previously considered by Ihmels *et al.* (2004) and Bhattacharya and De (2009). However, Ihmels *et al.*

(2004) did not allow negative correlation between genes, which we do allow. Bhattacharya and De (2009) found biclusters based on correlation between genes, but did not consider correlation between experimental conditions. Other biclustering structures have also been considered (Ben-Dor *et al.*, 2003; Ihmels *et al.*, 2004; Prelić *et al.*, 2006; Puolamäki *et al.*, 2008; Tanay *et al.*, 2002), to cite only a few. While all of the biclusters found by these methods probably represent biologically relevant regions in the data matrix, they may fail to identify all the genes and experimental conditions that are relevant to the biological process as the correlations between both genes and between experiments is not considered.

Furthermore, all of these methods, with the exception of Ben-Dor *et al.* (2003), use a model for a bicluster that is based on user-specified parameter values. Some parameters may be intuitive and their value easily defined. However, often the choice of parameter values is very difficult in practice, since they need to be determined for each data set and their meaning and effect may be difficult to understand. We define a parameter-free model by using Minimum Description Length (Rissanen, 1989). Furthermore, our biclustering algorithm has only few parameters. The algorithm captures relevant local correlation within the data matrix to the biclusters by extending each bicluster until no meaningful correlation can be extracted from its local surroundings. We also describe an algorithm to guarantee at most a certain level of overlap between biclusters, which utilizes the MDL-based model in the bicluster selection process. Furthermore, we show how to test the statistical significance of the found biclusters using the method by Hanhijärvi (2011).

2. Outline of Biclustering Algorithm

Our intuition for a good bicluster is that the genes within the bicluster are highly correlated, as well as the experimental conditions. Each gene and each experimental condition in such a bicluster is characterized by a single value, and the values within the bicluster can be reconstructed by multiplying the respective values for gene and experimental condition. It is biologically reasonable to assume that genes involved in the same process have similar expression levels over the relevant experimental conditions, but the magnitude of the expression levels may be gene-specific. Similar arguments apply for the experimental conditions. Therefore, we seek biclusters that are explained by a single linear component for genes and a single linear component for experimental conditions. Such structure corresponds to coherent values with a multiplicative model in the classification of Madeira and Oliveira (2004). An example of a good bicluster in our approach is illustrated in Figure 2.1. All the expression profiles have a very clear linear component.

When constructing a bicluster, we also need to balance between coherence and comprehensiveness. The coherence of a bicluster decreases with increasing size: a large bicluster is often incoherent. Typically the balance is controlled by user-defined parameter values, and, hence, the biclusters

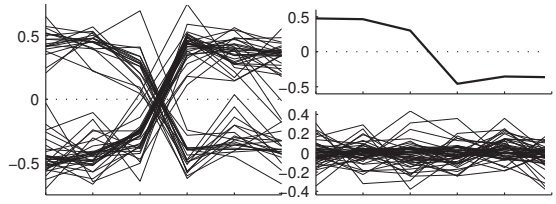


Figure 2.1. Example of a bicluster. Horizontal axis represents experimental conditions and vertical axis the relative expression levels. Left, each line represents the expression levels of a single gene over different experimental conditions. Top-right, strongest linear component of the expression levels. Bottom-right, residuals of the gene expression levels.

are strongly dependent on subjective choices. In Section 3, we use *Minimum Description Length* (MDL) to strike a theoretically founded balance between coherence and comprehensiveness, resulting in a parameter-free model for a bicluster. The result is a function to calculate the *description length* of a bicluster, which is used as the goodness measure of a bicluster. However, for clarity, we will start by describing the biclustering algorithm in this section and assume the model and description length are already defined.

2.1 Preliminaries

Gene expression data is represented by a real-valued input matrix D^* with N rows and M columns. Each row corresponds to a gene and each column corresponds to an experimental condition, where a value in a cell is the relative expression level of the gene in the respective experimental condition. The rows and columns of the input matrix D^* form sets $[N]$ and $[M]$, respectively, where $[N] = \{1, \ldots, N\}$.

We assume that the original input matrix D^* has been normalized to correct for microarray processing related artifacts. We further normalize D^* and obtain

$$D_{r,c} = \sqrt{\frac{NM}{S(D^*)}} D^*_{r,c},$$

where

$$S(D) = \sum_{c \in [M]} \sum_{r \in [N]} D^2_{r,c}$$

and $D_{a,b}$ indicates the submatrix of D that is determined by the set of rows $a \subseteq [N]$ and set of columns $b \subseteq [M]$. We use the notation $D_{r,c}$ to denote $D_{\{r\},\{c\}}$, which is a single cell as r and c are single indices. We will operate on D as we are not concerned about the absolute values but instead their relation to each other.

A bicluster is a submatrix of D, which is defined by a pair of sets (R, C) such that $R \subseteq [N]$ and $C \subseteq [M]$. Denote $n = |R|$ and $m = |C|$.

Assume for now that the description length $L(D|R, C)$ of the matrix D given the bicluster (R, C) is defined. The intuitive idea is that this measures the coding length of the whole data D when rows R and columns C are assumed to form a bicluster. Our definition for the description length is presented in Section 3. It is based on the cosine correlation between the rows in the bicluster calculated over the columns in the bicluster.

2.2 Finding bicluster seeds

As the number of potential biclusters is exponential in the number of rows and columns, it is not possible to exhaustively search biclusters that fit the model, *i.e.*, minimize $L(D|R,C)$. Instead, we first find bicluster seeds and extend them to biclusters. By a bicluster seed we mean a very small bicluster that defines an interesting location in the data matrix, and is a good place to start the search for a bicluster.

We next describe the method for finding bicluster seeds. We first discretize the matrix D to D^{01} by

$$D^{01}_{r,c} = \begin{cases} 1 & |D_{r,c}| \geq \delta \\ 0 & \text{otherwise} \end{cases},$$

where $|\cdot|$ denotes the absolute value and δ is a parameter. We then examine all the distinct triplets of columns $(i,j,k) \subset [M] \times [M] \times [M]$ of the matrix D^{01}. For each triplet, we collect all the rows to R that have 1s in all of the three columns (i,j,k). If the number of rows in R is at least a user defined threshold T, then a new bicluster seed is created from the triplet $C = \{i,j,k\}$ and R. The seed contains three columns and at least T rows.

Notice that other seeding methods can be used. For example, we could use as seeds the results of another biclustering algorithm or random 3 by 3 submatrices of the data.

2.3 Biclustering algorithm

Each seed B is considered separately. A bicluster is initialized to be the seed B. We update it iteratively by first fixing the set of columns and selecting a new set of rows, and then fixing the set of rows and selecting a new set of columns. This procedure is repeated until no more changes take place, and then the final set of rows and columns is returned.

For a fixed set of columns C, we would want to select the subset of rows $R \subseteq [N]$ that minimize the description length $L(D|R,C)$. However, as there are $\mathcal{O}(2^N)$ possible subsets of rows, we cannot do an exhaustive search and we have to resort to heuristic methods. We do a greedy search and include rows to the bicluster as follows. Let $X = D_{[N],C}$, *i.e.*, the matrix X is the submatrix of D that contains all the rows $[N]$ and only the columns C. Denote by $\overline{X}$ the rows of X that are normalized to unit length.

We compute the eigenvector corresponding to the largest eigenvalue of $\overline{XX}^\top$. The eigenvector has a value for each row in X and the magnitude represents, informally, the amount of agreement or disagreement between the respective row and the other rows. As the rows were first normalized to unit length, the agreement is measured by cosine correlation. The rows of X are then sorted descending according to magnitudes of the values in the eigenvector. This order is the order in which the rows are added to the bicluster. This procedure is similar to the grouping method by Freeman *et al.* (1999).

When we have the order of rows, we simply start from the three first rows and calculate the description length of the bicluster formed by those rows and the fixed set of columns. We then iteratively add rows to the bicluster according to the order and calculate the respective description lengths. Finally, we return the set of rows that gives the smallest description length as the new set of rows for the bicluster. Algorithm 1 gives the pseudo-code for selecting rows. Note that several optimizations can be used when implementing the algorithm. For example, the constant parts of the description length can be neglected, and using power search to find the strongest linear component enables to use the previous solution as a starting point, which drastically speeds up the search.

We use Algorithm 1 also for selecting the set of columns given the set of rows. We give the set of rows R and the input matrix D transposed as arguments for the algorithm. Therefore, we use the same procedure for selecting columns, ensuring equal treatment for both rows and columns. The set the algorithm returns is the new set of columns of the bicluster.

The algorithm for extending a bicluster seed by iteratively updating the set of rows and columns is given in Algorithm 2. Notice that the iteration can enter an infinite loop as the optimal set of rows for a given set of columns does not always mean that the same set of columns is optimal for the set of rows. Therefore, we store the sum of the description lengths at each iteration and halt the loop if a previously stored sum is met.

After the process has stopped, we return with a bicluster that, according to MDL, describes a local correlation structure in the data matrix.

Algorithm 1 Select_rows

Input: columns C, data matrix $D \in \mathbb{R}^{N \times M}$

Output: rows $R_{\min}$ that minimizes the description length of D on columns C

$X = D_{[N],C}$
$\overline{X}$ =normalize rows of X to unit length
v_1 =eigenvector of $\overline{XX}^{\top}$ corresponding to the largest eigenvalue
$t =$ indices of rows sorted according to descending order of $v_{1,r}^2$
$R = \{t_1\} \cup \{t_2\}$
$l_{min} = \infty$
$R_{min} = \emptyset$
for $n = 3$ **to** N
  $R = R \cup \{t_n\}$
  $\hat{l} = L(D|R, C)$
  if $\hat{l} < l_{min}$
    $R_{min} = R$
    $l_{min} = \hat{l}$
  end
end

Algorithm 2 Extend_bicluster_seed

Input: data matrix $D \in \mathbb{R}^{N \times M}$ with $S(D) = NM$, bicluster seed (R, C)

Output: bicluster (R, C), description lengths DL of encountered biclusters

$\hat{R} = \emptyset; \hat{C} = \emptyset; DL = \emptyset$
while $R \neq \hat{R}$ **and** $C \neq \hat{C}$
  $\hat{C} = C$; $\hat{R} = R$
  $R =$Select_rows(C, D)
  $C =$Select_rows$(R, D^{\top})$
  $l_C = L(D^{\top}|C, R)$
  $l_R = L(D|R, C)$
  if $l_R + l_C \in DL$
    break
  end
  $DL = DL \cup \{l_R + l_C\}$
end

2.4 Removing overlap

When all seeds have been extended to biclusters, it is possible that some of the produced biclusters overlap, or can even be exactly the same. We remove overlap among biclusters by discarding some of them. We measure the goodness of a bicluster by the description length $L(D|R, C)$. As the measure of overlap, we use an extended version of Jaccard's index

$$J(i, j) = \frac{|R_i \cap R_j||C_i \cap C_j|}{|R_i \cup R_j||C_i \cup C_j|},$$

where R_i is the set of rows in the bicluster i; R_j, C_i, and C_j are defined similarly. The index can obtain values from $[0, 1]$ and a large value signifies more overlap.

We start by calculating all the pairwise overlaps between biclusters. Then we select the pair of biclusters that have the largest overlap and discard the one that has higher description length $L(D|R, C)$. We then select a new pair of biclusters from the remaining set of biclusters that have the largest overlap. This is continued as long as the largest overlap exceeds a user-defined, maximum allowed overlap γ. This can vary between $[0, 1]$, meaning that with $\gamma = 1$ the overlap is completely ignored and the biclusters are allowed to completely overlap. Duplicates are still removed. With $\gamma < 1$, some overlap is accepted and with $\gamma = 0$, no overlap is allowed. When the procedure stops, the set of remaining biclusters is returned to the user as the final set of biclusters found in the data.

2.5 Choosing parameter values for seed generation

The model we define for a bicluster is parameter-free. However, since the search space is too large for an exhaustive search, we use the seed generation method to guide the search. The method requires two parameter values: the deviation bound δ and the minimum number of rows T. These affect the number and size of bicluster seeds found. While this has some effect in the final results, extending the bicluster seeds with MDL will reduce this effect and the found biclusters are expected to be more dependent on the MDL model than the initial parameters. Small values for both produce large number of seeds, while increasing the values will rapidly decrease the number of seeds. However, counting the number of seeds is very efficient and can be done quickly for even large data matrices. Therefore, searching for parameter values that result in a suitable

number of seeds is fast.

A possibility for finding upper bounds for δ is motivated by statistical significance testing. It is reasonable to set the parameter values low enough for the seeding method to find at least some seeds in random data. We suggest using binary search to find an upper bound for δ, given T, so that the mean number of seeds in a sample of random data is close to, say, 100. This procedure gives some indication of what parameter values are reasonable in the setting. We show in the experiments an example of this procedure.

3. Description length

In this chapter, we describe a mathematical model for biclusters and derive its description length using the Minimum Description Length principle.

3.1 Minimum Description Length

We introduce the basic principle of MDL by Rissanen (1989). For details, the reader is referred to the work by Grünwald (2007).

The idea of MDL is based on Kolmogorov complexity: complex data requires a long description and simple data requires only a short description. Informally, the MDL principle assumes that a model for the data is good if the data can be described minimally by using that model. For example, consider the following two nucleotide sequences:

ATGCATGCATGCATGCATGCATGC
GCCGATCCTCATCAAAAGTAAGTC

The first can be described by the sentence "ATGC 6 times", which requires 14 characters. As for the second, there is no obvious simple description of it except to write the sequence as is, which requires 24 characters. We can therefore conclude that the first sequence has clear structure and can be described easily, while the latter has no obvious structure and is lengthy to describe. To use MDL, one has to specify the family of models, *i.e.*, the set of possible descriptions of the data.

In MDL, the descriptions are not restricted to any natural language, and only the length of the description is of interest, usually in bits. The MDL idea is to calculate the description length, i.e., the number of bits required to describe the data with a selected model, plus the number of bits required to describe the model. If the description length is small,

then the model describes the structure in the data well. Conversely, if the description length is large, it is a poor model for the structure of the data.

One way of seeing the idea is to consider two people in different ends of a communication channel. The description length is the number of bits required to send the data through the communication channel so that the person on the other side is able to reconstruct the data without any errors. Minimizing the number of bits required is the MDL principle.

We use the formulation of Simple Refined MDL as described by Grünwald (2007), in which the likelihood of the data given model is first defined and then this likelihood is normalized over all possible data sets in the context. In more detail, we will proceed as follows. We will first define the likelihood of data D given a bicluster (R, C). This likelihood $P(D|R, C, \Theta)$ depends on a certain set of parameters Θ. We maximize the likelihood by finding estimates $\hat{\Theta}(D)$ for Θ based on the data. After this maximized likelihood is available, we normalize it over all possible data sets D and obtain the Normalized Maximum Likelihood density P_{nml} (Grünwald, 2007):

$$P_{nml}(D|R, C) = \frac{P(D|R, C, \hat{\Theta}(D))}{A_{D,R,C}}, \tag{3.1}$$

where

$$A_{D,R,C} = \int P(\hat{D}|R, C, \hat{\Theta}(\hat{D}))d\hat{D}.$$

The normalization removes the effect of the maximum likelihood estimators $\hat{\Theta}(D)$. If the integral is not defined, we need to constraint the space over which it is calculated, usually by constraining the possible values the estimators $\hat{\Theta}(D)$ can obtain.

Using the Normalized Maximum Likelihood density P_{nml}, the description length is defined by

$$L(D|R, C) = -\ln P_{nml}(D|R, C).$$

We measure the description length in logits, and therefore, the above formula has $\ln$ instead of $\log_2$. We will next define $P(D|R, C, \Theta(D))$, maximize it to obtain $P(D|R, C, \hat{\Theta}(D))$, normalize that to obtain $P_{nml}(D|R, C)$ and calculate $L(D|R, C)$.

3.2 Likelihood of bicluster columns

The likelihood $P(D|R, C, \Theta(D))$ is defined as a product of two parts: the likelihood for the part of the given data D that is covered by the columns

C of a given bicluster (R, C), and the likelihood of the data outside the columns of C. We start from the first part. Denote $X = D_{[N],C}$. Let $X = [x_1^\top, \ldots, x_N^\top]^\top$, *i.e.*, represent the submatrix of D that has all the rows and only the columns in C as a set of vectors X. The idea is to model the vectors x_i, where $i \in R$, with a set of variables that captures the common structure within the bicluster. The other vectors x_i, where $i \notin R$, are modeled as background with no clear structure.

We first scale the vectors in X to unit vectors: denote $||x_i|| = \sqrt{x_i^\top x_i}$ and $\overline{x}_i = x_i/||x_i||$. We model the lengths $||x_i||$ for all $i \in [N]$ as Gaussian variables. The likelihood of all of the lengths is

$$\prod_{i \in [N]} \phi(||x_i||, 0, \sigma_x^2), \tag{3.2}$$

where $\phi(\cdot, 0, \sigma^2)$ is a Gaussian probability density function with mean 0 and variance σ^2. The variance σ_x^2 is a parameter that is later estimated from the data. Note that the variance does not depend on the set of rows R. Thus, for any set of rows R and a fixed set of columns C, the likelihood in Equation (3.2) is constant, and therefore, it does not affect the results of Algorithm 1 as the eventual description length is partly based on this likelihood. We do not want the lengths to affect the search, since cosine correlation also ignores the lengths of the vectors.

We model the unit vectors $\overline{x}_i$ corresponding to the rows within the bicluster, $i \in R$, by two components

$$\overline{x}_i = \alpha_i v + \varepsilon_i,$$

where v is the unit length strongest linear component in the bicluster, $\alpha_i = \overline{x}_i^\top v$, *i.e.*, the cosine between v and $\overline{x}_i$, and ε_i is the residual vector orthogonal to v. However, the absolute value of the cosine coefficient α_i does not need to be represented explicitly, as it is completely determined by the length of the residual vector ε_i and the fact that $\overline{x}_i$ is unit length. We only need the sign for α_i, since

$$||\overline{x}_i||^2 \quad = \quad \overline{x}_i^\top \overline{x}_i = \alpha_i^2 + ||\varepsilon_i||^2 = 1$$

and hence

$$\alpha_i = \text{sign}(\overline{x}_i^\top v)\sqrt{1 - ||\varepsilon_i||^2}.$$

Therefore, when v is fixed and $i \in R$, we express each $\overline{x}_i$ as

$$\overline{x}_i = \left(\text{sign}(\overline{x}_i^\top v)\sqrt{1 - ||\varepsilon_i||^2} v + \varepsilon_i\right). \tag{3.3}$$

In other words, to reconstruct $\overline{x}_i$ for $i \in R$, it is sufficient to know the strongest linear component v, common to all $\overline{x}_i$, the general direction in relation to v, *i.e.*, $\text{sign}(\overline{x}_i^\top v)$, and the residual vector ε_i.

he only values we need to model for the sign function are 1 and -1. If $\text{sign}(\overline{x}_i^\top v) = 0$, then according to Equation (3.3) $\varepsilon_i = \overline{x}_i$ and the length of ε_i is 1 which is an incidation that $\text{sign}(\overline{x}_i^\top v) = 0$. We use the uniform density to model $\text{sign}(\overline{x}_i^\top v)$ for all $i \in R$, which results in the likelihood 2^{-n}.

We model the residual vectors ε_i with the multivariate normal density $\Phi(\varepsilon_i, \mu, \Sigma)$, where μ is the mean vector parameter and Σ is the covariance matrix parameter. We use as μ a zero vector and as Σ the matrix $\sigma_\varepsilon^2 I$, where I is the identity matrix and σ_ε^2 is a variance parameter that will be later estimated from data. The likelihood of all residual vectors ε_i, where $i \in R$, is then

$$\prod_{i \in R} \Phi(\varepsilon_i, 0, \sigma_\varepsilon^2 I) = \prod_{i \in R} \prod_{j=1}^{m} \phi(\varepsilon_{ij}, 0, \sigma_\varepsilon^2). \tag{3.4}$$

The biclusters with small values for ε_i, *i.e.*, that have a clear structure and strong coherence, have a large likelihood and are therefore considered good. Other symmetric densities with mode at 0 could have been used instead of $\Phi(\varepsilon_i, \mu, \Sigma)$, but since the multivariate normal density is easier to handle later on, it was chosen.

We model the unit vectors $\overline{x}_i$ corresponding to the rows with $i \notin R$ by using the uniform density over the m dimensional unit sphere (Huber, 1982)

$$B_m = \left(\frac{\pi^{m/2}}{\Gamma(m/2)} \right)^{-1}. \tag{3.5}$$

The value B_m is the inverse of the surface area of the m-dimensional unit sphere. The likelihood of the vectors $\overline{x}_i$ with $i \notin R$ is then $B_m^{(N-n)}$ as there are $N - m$ such vectors. We use uniform density for these vectors as we consider them to be background noise with no clear structure, and therefore, their contents should not affect the likelihood.

Combining the likelihoods in Equations (3.2) and (3.4) with the likelihoods of the signs 2^{-n} and the unit vectors not in the bicluster $B_m^{(N-n)}$, the likelihood of the data D on the columns C of the bicluster is

$$\begin{aligned} & P(D_{[N],C} | R, \sigma_x^2, \sigma_\varepsilon^2, v) \\ = & \left(\prod_{i \in [N]} \phi(||x_i||, 0, \sigma_x^2) \right) B_m^{(N-n)} \left(\prod_{i \in R} \prod_{j=1}^{m} \phi(\varepsilon_{ij}, 0, \sigma_\varepsilon^2) \right) 2^{-n}, \end{aligned} \tag{3.6}$$

where σ_ε^2 depends on v as shown in Equation (3.3). The parts in order are the lengths of all of the rows, the directions of the rows outside of the bicluster, the residuals of the rows within the bicluster, and finally the signs of the rows within the bicluster with respect to the strongest linear component v. The likelihood is the density of the data D given the bicluster (R, C) and it depends on the parameters $\Theta = \{\sigma_x^2, \sigma_\varepsilon^2, v\}$. We will next find estimators $\hat{\Theta}(D)$ for these parameters by maximizing the likelihood in Equation (3.6).

The estimator that maximizes the likelihood with respect to σ_x^2 is

$$\hat{\sigma}_x^2 = \frac{1}{N} \sum_{i \in [N]} ||x_i||^2 = \frac{1}{N} \sum_{i \in [N]} \sum_{j=1}^{m} x_{ij}^2 = \frac{1}{N} S(D_{[N],C}), \tag{3.7}$$

which is the usual maximum likelihood estimator of variance. Similary, the estimator for σ_ε^2 is

$$\hat{\sigma}_\varepsilon^2 = \frac{1}{nm} \sum_{i \in R} \sum_{j=1}^{m} \varepsilon_{ij}^2 = \frac{1}{nm} \sum_{i \in R} ||\varepsilon_i||^2. \tag{3.8}$$

The residuals ε_i depend on v and, hence, the estimator σ_ε^2 requires knowing v. We use the estimator $\hat{\sigma}_\varepsilon^2$ in the likelihood $P(D_{[N],C}|R, \sigma_x^2, \sigma_\varepsilon^2, v)$ and maximize it to find the estimator for the strongest linear component v:

$$\begin{aligned}
\hat{v} &= \arg\max_v P(D_{[N],C}|R, \sigma_x^2, \hat{\sigma}_\varepsilon^2, v) \\
&= \arg\max_v \prod_{i \in R} \prod_{j=1}^{m} \phi(\varepsilon_{ij}, 0, \hat{\sigma}_\varepsilon^2) \\
&= \arg\max_v \left(2\pi\hat{\sigma}_\varepsilon^2\right)^{-nm/2} \exp(-\frac{1}{2}\hat{\sigma}_\varepsilon^{-2} \sum_{i \in R} \sum_{j=1}^{m} \varepsilon_{ij}^2) \\
&= \arg\max_v \left(2\pi e\hat{\sigma}_\varepsilon^2\right)^{-nm/2} \\
&= \arg\max_v \left(\frac{2\pi e}{nm} \sum_{i \in R} ||\varepsilon_i||^2\right)^{-nm/2} \\
&= \arg\min_v \sum_{i \in R} ||\varepsilon_i||^2 \\
&= \arg\min_v \sum_{i \in R} ||\overline{x}_i - \alpha_i v||^2 \\
&= \arg\min_v \sum_{i \in R} (\overline{x}_i - (\overline{x}_i^\top v)v)^\top (\overline{x}_i - (\overline{x}_i^\top v)v) \\
&= \arg\max_v \sum_{i \in R} (\overline{x}_i^\top v)^2 \\
&= \arg\max_v v^\top (\sum_{i \in R} \overline{x}_i \overline{x}_i^\top) v.
\end{aligned}$$

The maximum likelihood estimator of v is, therefore, the eigenvector corresponding to the largest eigenvalue of $\sum_{i \in R} \overline{x}_i \overline{x}_i^\top$. This is because the

eigenvalues λ_k and eigenvectors v_k satisfy

$$(\sum_{i \in R} \overline{x}_i \overline{x}_i^\top) v_k = \lambda_k v_k,$$

and multiplying both sides with $v_k^\top$ yields

$$v_k^\top (\sum_{i \in R} \overline{x}_i \overline{x}_i^\top) v_k = \lambda_k,$$

since $v_k^\top v_k = 1$ for all $k \in [m]$. We index the eigenvalues and eigenvectors such that $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_m$. Therefore, the maximum likelihood estimator for v is $\hat{v} = v_1$.

With the estimator $\hat{v}$, we can express the estimator $\hat{\sigma}_\varepsilon^2$ in Equation (3.8) as

$$\begin{aligned}
\hat{\sigma}_\varepsilon^2 &= \frac{1}{nm} \sum_{i \in R} \sum_{j=1}^{m} \varepsilon_{ij}^2 \\
&= \frac{1}{nm} \sum_{i \in R} \sum_{j=1}^{m} (\overline{x}_{ij} - \alpha_i \hat{v}_j)^2 \\
&= \frac{1}{nm} \sum_{i \in R} ||\overline{x}_i - \alpha_i \hat{v}||^2 \\
&= \frac{1}{nm} \left(\sum_{i \in R} \overline{x}_i^\top \overline{x}_i - v_1^\top (\sum_{i \in R} \overline{x}_i \overline{x}_i^\top) v_1 \right) \\
&= \frac{1}{nm} (n - \lambda_1) . \qquad (3.9)
\end{aligned}$$

Using Equations (3.7), (3.8) and (3.9) in Equation (3.6) and simplifying, we get the maximum likelihood

$$\begin{aligned}
& P(D_{[N],C} | R, \hat{\sigma}_x^2, \hat{\sigma}_\varepsilon^2, \hat{v}) \\
= & \left(\frac{2\pi e}{N} S(D_{[N],C}) \right)^{-N/2} B_m^{(N-n)} \left(\frac{2\pi e}{nm} (n - \lambda_1^2) \right)^{-nm/2} 2^{-n}. \qquad (3.10)
\end{aligned}$$

3.3 Rest of the model

We describe in this section the likelihood of the data in D that is on the columns that do not belong to the bicluster. We also describe the encodings for the bicluster location in the matrix D and the size of the matrix D.

We model the submatrix $D_{[N],[M]\setminus C}$ as a collection of Gaussian variables:

$$P(D_{[N],[M]\setminus C} | \sigma_y^2) = \prod_{r \in [N]} \prod_{c \in [M]\setminus C} \phi(D_{r,c}, 0, \sigma_y^2).$$

That is, each entry $D_{r,c}$ in the matrix is assumed to be generated by a normal distribution with zero mean and variance σ_y^2. Similarly to Equation (3.7), the maximum likelihood estimator for σ_y^2 is

$$\hat{\sigma}_y^2 = \frac{1}{N(M-m)} S(D_{[N],[M]\setminus C}).$$

With this, the maximum likelihood for the submatrix is

$$P(D_{[N],[M]\setminus C}|\hat{\sigma}_y^2) = \left(\frac{2\pi e}{N(M-m)} S(D_{[N],[M]\setminus C})\right)^{-N(M-m)/2}. \tag{3.11}$$

Algorithms 1 and 2 are used to locate biclusters in the given input matrix D. The matrix remains constant throughout the search, and therefore, we are not concerned about the description length of the complete data, but the part of it that depends on the bicluster: its size and goodness of fit to the model. We describe here how the size of the data and the location of the biclusters could be encoded such that they remain the same for any bicluster of the same matrix D.

To encode the size of the data set, we use the unbounded encoding for positive integers (Grünwald, 2007) with which the code length is $2\ln N + 2$ in bits, where N is the integer to encode. The size of the data matrix has thus a code length of $2(\ln N + \ln M + 2)\ln 2$.

We encode the bicluster location with uniform code over all bicluster locations, and therefore, we use a single bit for each row and column to signify if the row or column is included in the bicluster or not. The corresponding code length is $(N + M)\ln 2$. An alternative encoding would have been to first encode the number of rows (or columns) and then encode the indices to the rows (or columns). Such encoding would have the length $((n+1)\ln N + (m+1)\ln M)\ln 2$. The latter encoding is shorter than the former if n or m are small or close to the dimensions of the data, N or M, respectively. The choice of encoding is made out of preference: we do not want to favor any size of biclusters but treat all biclusters on equal ground.

We have shown that the data D can be completely encoded with the above definitions. Furthermore, the encoding is constant for a constant D, except for the likelihoods in Equations (3.10) and (3.11).

3.4 Normalized Maximum Likelihood

Combining the maximum likelihoods in Equations (3.10) and (3.11), we get

$$P(D|R,C,\hat{\sigma}_x^2,\hat{\sigma}_\varepsilon^2,\hat{\sigma}_y^2,\hat{v}) = P(D_{[N],[M]\setminus C}|\hat{\sigma}_y^2)P(D_{[N],C}|R,\hat{\sigma}_x^2,\hat{\sigma}_\varepsilon^2,\hat{v}).$$

This depends on the maximum likelihood estimators $\hat{\sigma}_x^2$, $\hat{\sigma}_\varepsilon^2$, $\hat{\sigma}_y^2$ and $\hat{v}$. According to the Simple Refined MDL (Grünwald, 2007), we renormalize the likelihood to remove the dependence on the estimators. Therefore, we calculate the Normalized Maximum Likelihood density

$$\overline{P}(D|R,C) = \frac{P(D|R,C,\hat{\sigma}_x^2,\hat{\sigma}_\varepsilon^2,\hat{\sigma}_y^2,\hat{v})}{\int P(\hat{D}|R,C,\hat{\sigma}_x^2,\hat{\sigma}_\varepsilon^2,\hat{\sigma}_y^2,\hat{v})d\hat{D}}, \tag{3.12}$$

where the integration is over the space of matrices of the same size as D and with $S(D) = NM$. We plug in the definition for $P(D|R,C,\hat{\sigma}_x^2,\hat{\sigma}_\varepsilon^2,\hat{\sigma}_y^2,\hat{v})$, simplify and obtain

$$\overline{P}(D|R,C) = \frac{S_{1,D}^{-N(M-m)/2} S_{2,D}^{-N/2} \left(n-\lambda_{1,D}\right)^{-nm/2}}{\int S_{1,\hat{D}}^{-N(M-m)/2} S_{2,\hat{D}}^{-N/2} \left(n-\lambda_{1,\hat{D}}\right)^{-nm/2} d\hat{D}}, \tag{3.13}$$

where $S_{1,D} = S(D_{[N],[M]\setminus C})$ and $S_{2,D} = S(D_{[N],C})$. The dependence of $\lambda_{1,D}$ on D is highlighted as it is calculated from the directions of the vectors in the bicluster.

Let us next focus on the integral in the denominator. The term

$$S_{1,\hat{D}}^{-N(M-m)/2} = \left(\sum_{r\in[N]}\sum_{c\in[M]\setminus C} \hat{D}_{r,c}^2\right)^{-N(M-m)/2}$$

only depends on the columns that are not in the bicluster $\hat{D}_{[N],[M]\setminus C}$. Similary, the terms $S_{2,\hat{D}}^{-N/2}\left(n-\lambda_{1,\hat{D}}\right)^{-nm/2}$ only depend on the columns that are in the bicluster $\hat{D}_{[N],C}$, as $\lambda_{1,\hat{D}}^2$ is calculated from $\hat{D}_{R,C}$ and $S_{1,\hat{D}} = \sum_{r\in[N]}\sum_{c\in C}\hat{D}_{r,c}^2$. Therefore, these parts can be integrated separately with the restriction that $S_{1,\hat{D}} + S_{2,\hat{D}} = NM$, *i.e.*, the sum of squares for both parts must sum to NM. This is because we required that $\hat{D}$ is normalized to $S(\hat{D}) = NM$. Controlling the proportions of sum of squares with s, *i.e.*, $S_{1,\hat{D}} = sNM$ and $S_{2,\hat{D}} = (1-s)NM$, we can write the integral

in Equation (3.13) as

$$\begin{aligned}
&\int S_{1,\hat{D}}^{-N(M-m)/2} S_{2,\hat{D}}^{-N/2} \left(n - \lambda_{1,\hat{D}}\right)^{-nm/2} d\hat{D} \\
=&\int_0^1 \left(\int_{S(Y)=sNM} (sNM)^{-N(M-m)/2} dY \right. \\
&\left. \int_{S(X)=(1-s)NM} ((1-s)NM)^{-N/2} \left(n - \lambda_{1,X}\right)^{-nm/2} dX \right) ds
\end{aligned} \tag{3.14}$$

where $X \in \mathbb{R}^{N \times m}$ and $Y \in \mathbb{R}^{N \times (M-m)}$.

The integral over the space $Y \in \mathbb{R}^{N \times (M-m)}$ with $S(Y) = sNM$, *i.e.*, the columns not in the bicluster, is equal to integrating over the $N(M-m)$-dimensional sphere with radius $(sNM)^{\frac{1}{2}}$. The corresponding volume (Huber, 1982) is

$$\frac{\pi^{N(M-m)/2} (sNM)^{N(M-m)/2-1}}{\Gamma(N(M-m)/2)},$$

which results to

$$\int_{S(Y)=sNM} (sNM)^{-N(M-m)/2} dY = \frac{\pi^{N(M-m)/2} (sNM)^{-1}}{\Gamma(N(M-m)/2)}. \tag{3.15}$$

The second inner integral in Equation (3.14)

$$\begin{aligned}
&\int_{S(X)=(1-s)NM} ((1-s)NM)^{-N/2} \left(n - \lambda_{1,X}\right)^{-nm/2} dX \\
=&((1-s)NM)^{-N/2} \int_{S(X)=(1-s)NM} \left(n - \lambda_{1,X}\right)^{-nm/2} dX
\end{aligned} \tag{3.16}$$

is more difficult. It does not have a known analytical solution as it involves calculating over all sets of N vectors of length m and finding out the set n of them that together have a covariance matrix with the largest λ_1 over all subsets of n rows, scaled to unit length. However, we can write

$$\begin{aligned}
&\int_{S(X)=(1-s)NM} \left(n - \lambda_{1,X}\right)^{-nm/2} dX \\
=&V_s \int_{S(X)=(1-s)NM} \left(n - \lambda_{1,X}\right)^{-nm/2} \frac{1}{V_s} dX \\
=&V_s f(N,n,m)
\end{aligned} \tag{3.17}$$

where

$$V_s = \frac{\pi^{Nm/2} ((1-s)NM)^{Nm/2-1}}{\Gamma(Nm/2)}$$

and

$$f(N,n,m) = \int_{S(X)=(1-s)NM} \left(n - \lambda_{1,X}\right)^{-nm/2} \frac{1}{V_s} dX. \tag{3.18}$$

The function $f(N, n, m)$ does not depend on s, since the integrand only depends on the directions of the row vectors of X and V_s^{-1} cancels out the effect of the volume of the space $X \in \mathbb{R}^{N \times m} : S(X) = (1-s)NM$. We will show how to approximate $f(N, n, m)$ in the next section.

Applying the Equations (3.15), (3.16) and (3.17) to Equation (3.14), we get

$$\frac{\pi^{NM/2} NM^{(N(m-1)-2)/2}}{\Gamma(N(M-m)/2)\Gamma(Nm/2)} f(N,n,m) \int_0^1 s^{-1}(1-s)^{N(m-1)/2-1} ds. \qquad (3.19)$$

The integrand approaches infinity as s approaches 0. Therefore, we must bound the integration from below with s_{min}. This means that the columns of D not in the bicluster have to have at least some nonzero values, and their sum of squares has to be at least NMs_{min}. The upper limit does not need to be restricted, since $N \geq 3$ and $m \geq 3$, and therefore, the integrand is finite as s approaches 1.

Let us assume $q = N(m-1)/2-1$ is an integer. We can then approximate the integral in Equation (3.19) with

$$\begin{aligned}
\int_{s_{min}}^1 s^{-1}(1-s)^q ds &= \int_{s_{min}}^1 s^{-1} \sum_{i=0}^{q} \binom{q}{i} (-s)^i ds \\
&= \sum_{i=0}^{q} \binom{q}{i} (-1)^i \int_{s_{min}}^1 s^{i-1} ds \\
&= -\ln s_{min} - \sum_{i=1}^{q} \binom{q}{i} (-1)^{i-1} \frac{1}{i} (1 - s_{min}^i) \\
&\approx -\ln s_{min} - H_q,
\end{aligned}$$

where H_q is the qth harmonic number and $s_{\min}^i \approx 0$. This can be further approximated by

$$-\ln s_{min} - H_q \quad \approx \quad -\ln s_{min} - \ln q - \gamma,$$

where $\gamma \approx 0.5772156649$ is the Euler–Mascheroni constant (Knuth, 1997). The effect of this term in the description length is $\ln(-\ln s_{min} - \ln q - \gamma)$. When s_{min} is set to a small enough number, this term has very little effect in the overall description length. Furthermore, for a non-integer value for q, the value for the integral in Equation (3.19) is similar to integer values for q. Therefore, we will remove the integral from Equation (3.19) and obtain

$$\frac{\pi^{NM/2} NM^{(N(m-1)-2)/2}}{\Gamma(N(M-m)/2)\Gamma(Nm/2)} f(N,n,m)$$

as the denominator in Equation (3.13). Finally, the Normalized Maximum Likelihood in Equation (3.12) becomes

$$\begin{aligned}\overline{P}(D|R,C) =& S_{1,D}^{-N(M-m)/2} S_{2,D}^{-N/2} \left(n - \lambda_{1,D}\right)^{-nm/2} f(N,n,m)^{-1} \\ & \cdot \frac{\Gamma(N(M-m)/2)\Gamma(Nm/2)}{\pi^{NM/2} N M^{(N(m-1)-2)/2}}.\end{aligned} \tag{3.20}$$

3.5 Approximating $f(N,n,m)$

Unfortunately the function $f(N,n,m)$ in Equation (3.18) cannot be computed directly. We will approximate it by using sampling to discover the distribution of $n - \lambda_1^2$ in the space $\mathbb{R}^{N\times m}$, where N is the height of D and m is the number of columns in the bicluster C. We have to do this for all N, n, and m that are possible to encounter with input matrix D. However, since $n - \lambda_1$ only depends on the directions of the row vectors, we can write Equation (3.18) as

$$\begin{aligned} f(N,n,m) &= \int\limits_{S(X)=(1-s)NM} \left(n - \lambda_{1,X}\right)^{-nm/2} V_s^{-1} dX \\ &= \int \left(n - \lambda_{1,\overline{X}}\right)^{-nm/2} B_m^N d\overline{X}, \end{aligned} \tag{3.21}$$

where the last integration is over the space of all possible N unit vectors in m-dimensions. Therefore, we can directly sample N vectors from the space of m-dimensional unit vectors. This can be done by simply drawing normally distributed values independently for each cell of X and normalizing each row to unit length (Marsaglia, 1972).

Algorithm 3 describes the method for obtaining k samples for all $n \in \{3,\ldots,N\}$ given N and m. This is a slightly modified version of Algorithm 1 that is used to select the rows of a bicluster.

When a sufficient number of samples have been obtained, we can proceed to calculate the integral in Equation (3.21). Monte Carlo integration, $\frac{1}{k}\sum_{i=1}^{k} \Lambda_{n,i}^{-nm/2}$, could not be used as the integrand is numerically extremely difficult to handle for all but very small matrices. Instead, we use the samples $\Lambda_{n,i}$ to find the distribution of $n - \lambda_{1,n}$ in the space of N unit vectors of m-dimensions and integrate over that distribution. With this and substituting $z = n - \lambda_{1,n}$, Equation (3.21) becomes

$$\int \left(n - \lambda_{1,\overline{X}}\right)^{-nm/2} B_m^N d\overline{X} = \int\limits_{n\epsilon}^{n(1-1/m)} z^{-nm/2} p(z) dz, \tag{3.22}$$

Algorithm 3 Sample values

Input: number of rows N, number of columns m, number of samples k

Output: matrix of values Λ

for $j = 1$ **to** k
  sample $X \in \mathbb{R}^{N \times m}$
  $\overline{X}$ =normalize rows of X to unit length
  v_1 =maximum eigenvector of $\overline{X}\overline{X}^\top$
  t =sort rows $[N]$ in descending order of $v_{1,r}^2$
  $R = t(1) \cup t(2)$
  for $n = 3$ **to** N
    $R = R \cup t(n)$
    $\lambda_{1,n} = \max_v v^\top (\sum_{r \in R} x_r x_r^\top) v$
    $\Lambda_{n,j} = n - \lambda_{1,n}$
  end
end

where $p(z)$ is the distribution of $n - \lambda_{1,n}$ in the space of N unit vectors of m-dimensions. The idea is therefore to find an appropriate $p(z)$ from $\Lambda_{n,i}$ such that $p(z)$ fits well to the values of $\Lambda_{n,i}$ and that the integral can be solved.

The integration limits $n(1 - 1/m)$ and ϵn in Equation (3.22) come from the limits of $n - \lambda_{1,n}$. The sum of eigenvalues is equal to the sum of the lengths of the unit vectors and is thus equal to n, $\sum_{i=1}^{m} \lambda_{i,n} = n$. The maximum eigenvalue $\lambda_{1,n}$ is at least as large as the minimum eigenvalue $\lambda_{m,n}$ and they are equal if $\lambda_{1,n} = \frac{n}{m}$. Therefore, the upper limit has the value $n(1 - 1/m)$. The lower limit ϵn is user defined and it dictates how close a bicluster may come to a perfect match. If all the vectors n in the bicluster were allowed to be equal, which means the bicluster has perfect coherence, then $\lambda_{1,n} = n$ and the integrand in Equation (3.22) would not be finite. Therefore, we must limit the biclusters away from perfect fit, *i.e.*, limit $\lambda_{1,n}$ away from n. We will use $\epsilon = 0.01$ in the experiments as it produced a continuous description length for increasing n, and we didn't discover biclusters with a better fit in random data.

We use the gamma distribution for $p(z)$ in Equation (3.22). There are two reasons for choosing the gamma distribution. First, the gamma distribution fitted very well to all of the histograms of the values for several tested combinations of N, m and n. Second, the form of the integrand in Equation (3.22) resembles a part of the gamma probability distribution

function and thus will help later on when calculating the integral with the fitted gamma distribution function.

We fit a gamma distribution to the values $\Lambda_{i,\cdot}$ and obtain the maximum likelihood estimates for the the scale $\hat{\theta}$ and shape $\hat{k}$ parameters (Forbes *et al.*, 2010). The estimates depend on N, n and m. For clarity, the dependence is not explicitly shown in the following equations. Using the estimates, the integral in Equation (3.22) can then be written as

$$\begin{aligned} & \int_{n\epsilon}^{n(1-1/m)} z^{-nm/2} p(z) dz, \\ \approx & \int_{\epsilon n}^{n(1-1/m)} z^{-nm/2} z^{\hat{k}-1} e^{-z/\hat{\theta}} (\hat{\theta}^{\hat{k}} \Gamma(\hat{k}))^{-1} dz \\ = & \Gamma(\hat{k})^{-1} \hat{\theta}^{-nm/2} \int_{\epsilon n/\hat{\theta}}^{n(1-1/m)/\hat{\theta}} y^{\hat{k}-nm/2-1} e^{-y} dy, \end{aligned} \tag{3.23}$$

where $y = z/\hat{\theta}$. The integral in Equation (3.23) is of the form

$$F(\alpha, a, b) = \int_a^b y^{\alpha-1} e^{-y} dy, \tag{3.24}$$

which is a generalization of the incomplete gamma function, in which it is required that $\alpha > 0$ and $a = 0$. We use integration by parts to find out the chain rule for $\alpha < 0$

$$\begin{aligned} \int_a^b y^{\alpha-1} e^{-y} dy &= \frac{1}{\alpha} \left(b^\alpha e^{-b} - a^\alpha e^{-a} \right) + \frac{1}{\alpha} \int_a^b y^\alpha e^{-y} dy \\ \Leftrightarrow F(\alpha, a, b) &= \frac{1}{\alpha} \left(b^\alpha e^{-b} - a^\alpha e^{-a} \right) + \frac{1}{\alpha} F(\alpha + 1, a, b). \end{aligned}$$

For any $\alpha \geq 0$, we use existing implementations of the incomplete gamma function $I(\alpha, a)$ to calculate

$$F(\alpha, a, b) = I(\alpha, b) - I(\alpha, a).$$

Combining Equations (3.23), (3.24), and (3.22) with (3.21), we get

$$f(N, n, m) = \Gamma(\hat{k})^{-1} \hat{\theta}^{-nm/2} F\left(\hat{k} - \frac{nm}{2}, \frac{\epsilon n}{\hat{\theta}}, \frac{n(1 - \frac{1}{m})}{\hat{\theta}} \right). \tag{3.25}$$

Finally, using Equation (3.25) in Equation (3.20) we get the Normalized Maximum Likelihood

$$\begin{aligned} & \overline{P}(D|R, C) \\ = & S_{1,D}^{-N(M-m)/2} S_{2,D}^{-N/2} \left(n - \lambda_{1,D}\right)^{-nm/2} F\left(\hat{k} - \frac{nm}{2}, \frac{\epsilon n}{\hat{\theta}}, \frac{n(1 - \frac{1}{m})}{\hat{\theta}} \right)^{-1} \\ & \cdot \frac{\Gamma(N(M-m)/2) \Gamma(Nm/2) \Gamma(\hat{k}) \hat{\theta}^{nm/2}}{\pi^{NM/2} N M^{(N(m-1)-2)/2}}, \end{aligned} \tag{3.26}$$

and the corresponding description length is

$$
\begin{aligned}
& L(D|R,C) \\
=& -\ln \overline{P}(D|R,C) \\
=& \frac{N(M-m)}{2}\ln S_{1,D} + \frac{N}{2}\ln S_{2,D} + \frac{nm}{2}\ln(n-\lambda_{1,D}) \\
& -\ln\Gamma\left(\frac{N(M-m)}{2}\right) - \ln\Gamma\left(\frac{Nm}{2}\right) - \ln\Gamma(\hat{k}) - \frac{nm}{2}\ln\hat{\theta} \\
& + \frac{NM}{2}\ln\pi + \frac{N(m-1)-2}{2}\ln NM + \ln F\left(\hat{k}-\frac{nm}{2}, \frac{\epsilon n}{\hat{\theta}}, \frac{n(1-\frac{1}{m})}{\hat{\theta}}\right).
\end{aligned}
\tag{3.27}
$$

4. Experiments

In this section, we present the experimental and evaluative results of the proposed biclustering method. We will use the name BiMDL+DS for the biclustering method with the seed generation component presented above. We will also experiment with using random 3 by 3 seeds and extending them using the proposed method. This method we will call BiMDL+RND.

We compare BiMDL+DS and BiMDL+RND to BiMax by Prelić *et al.* (2006), Order Preserving Submatrix Algorithm OPSM by Ben-Dor *et al.* (2003), Samba by Tanay *et al.* (2002), Iterative Signature Algorithm ISA by Ihmels *et al.* (2004), and BCCA by Bhattacharya and De (2009).

4.1 Setting

The experiments were carried out for a gene expression dataset that was produced to map the main bio-synthetic pathways of *Arabidopsis thaliana* (*Ath*)[1].

We approximated the normalization part in Equation (3.27) using 10000 samples. Upper bound for seed deviation bound δ was found using the method described in Section 2.5. The chosen method for matrix randomization is discussed later. We found that $T = 3$ and $\delta = 25.9$ produced on average 100 biclusters. Table 4.1 lists the chosen parameter values and the number of found biclusters, along with basic characteristics of the data set. Notice that we chose $\delta = 20$ as we had sufficient computer power to handle more calculations induced by the choice. For BiMDL+RND, 10000 random seeds were generated. The maximum allowed overlap between biclusters was set to $\gamma = 0.1$ throughout the experiments. Figure 2.1 illustrates the expression levels of a bicluster obtained from *Ath*

[1] http://www.tik.ee.ethz.ch/~sop/bimax/SupplementaryMaterial/data sets/BiologicalValidation/data/arabidopsis/ath_MetabolicMap,NASC,734x69.txt

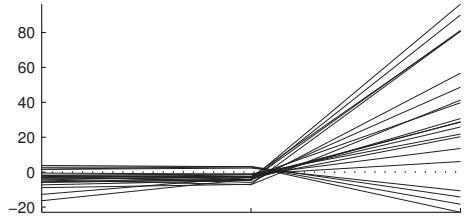


Figure 4.1. Example bicluster. Each line represents a gene and the three horizontal positions represent experimental conditions.

Table 4.1. Characteristics of the data set, chosen parameter values and number of seeds and biclusters found.

				BiMDL+DS		BiMDL+RND
# genes	# expr.	δ	T	# seeds	# bicl.	# bicl.
734	69	20	3	892	12	58

using BiMDL+DS.

For the other methods, we used the same parameters as used by (Prelić *et al.*, 2006) and (Bhattacharya and De, 2009).

4.2 Statistical significance

We used SWAPDISCRETIZED (Ojala *et al.*, 2009) to create random versions of the original data matrix. It is an efficient randomization method and suitable for gene expression data. The values were first grouped to three groups: significantly high, significantly low, and no significant expression. The grouping was based on the chosen δ value, with values at least δ grouped to significantly high, values at most $-\delta$ grouped to significantly low, and others to no significance. SWAPDISCRETIZED maintains the marginal distributions of groups on rows and on columns, while swapping the values in the matrices. In the end, each row and column will contain the original distribution of significantly high and low expressions, while the actual values are shuffled.

SWAPDISCRETIZED is based on small modifications of the data matrix, and the number of these modifications is a parameter. We used the method in Hanhijärvi *et al.* (2009) to find a value for this parameter, and chose the value of 4 times the number of cells in the data matrix. We produced 100 random matrices in such a way and used the method by Hanhijärvi (2011) to calculate the FWER adjusted p-values for the biclusters. For

BiMDL+DS, 6 out of the 12 biclusters were statistically significant with less than 5% FWER level, and correspondingly 51 out of 58 biclusters for the BiMDL+RND method. It is clear that both of the methods were able to locate statistically significant biclusters even with such a strict null distribution.

4.3 Extraction power

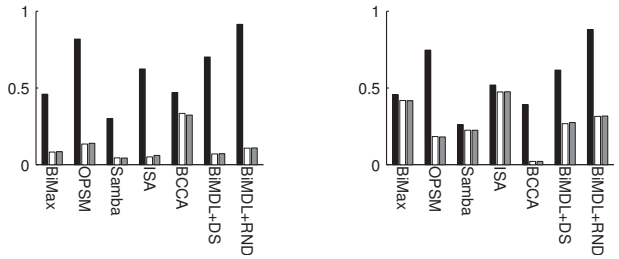
We measured the power of the methods to extract local correlation in the data set. The score is calculated as the mean of the square of cosine correlation coefficients between all rows, or columns of the bicluster,

$$g(D, R, C) = \frac{1}{|R|^2} \sum_{r_1, r_2 \in R} \frac{\left(\sum_{c \in C} D_{r_1 c} D_{r_2 c}\right)^2}{\sum_{c \in C} D_{r_1 c}^2 \sum_{c \in C} D_{r_2 c}^2}.$$

The square of the correlation coefficient is familiar from machine learning, and it is known as coefficient of determination, where in linear regression, it expresses the ability of a regressor to explain the variance of the output variable. The range of values is between $[0, 1]$ and a higher value is considered better as more variance is explained.

We also calculate the score along different sets of columns. We use the set of columns not included in the bicluster, $g(D, R, [M] \backslash C)$, and all columns, $g(D, R, [M])$. These two express how much correlation is outside of the bicluster, and overall between the rows along all columns. The same scores are calculated between columns of the bicluster, and the set of rows is varied, $g(D^\top, C, R)$, $g(D^\top, C, [N] \backslash R)$, and $g(D^\top, C, [N])$. These scores are averaged over all biclusters. The averages expresses the overall ability of a biclustering algorithm to locate areas of high correlation, and to capture it, finding each bicluster so that only little correlation is left outside of it. Figure 4.2 illustrates the average scores on both rows and columns for different biclustering methods.

On rows, OPSM and the BiMDL methods locate high correlation within the biclusters (black bars) and little correlation outside of it (white and gray bars). This means that the biclusters express local correlation structure within the matrix, and that the rows do not correlate much on the other columns. Such biclusters are very interesting, since they express deviant behavior. ISA and BiMax are not far behind, but Samba and BCCA did not perform that well. The results between the columns are similar, in that OPSM and BiMDL outperform the other methods, even more clearly



(a) rows, genes

(b) columns, experimental conditions

Figure 4.2. Average scores of mean squared cosine correlation for different biclustering methods. Black bars are calculated along the columns (rows) of the bicluster, white bars along columns (rows) not in the bicluster, and gray bars along all columns (rows).

than on rows.

4.4 Gene interaction network

The second comparison was to assess how known gene interaction networks correspond to the found biclusters. The gene interaction network was constructed from metabolic pathway information (Wille *et al.*, 2004) by creating an edge between genes if they are involved in the same pathway, which resulted in 98.4% cover of genes and 6.6% of all the gene pairs had an edge between them.

The intuition is that gene interactions should be reflected in the gene expression data. This was measured by calculating the number of disconnected gene pairs in a bicluster, as well as the average distance of the gene pairs that are connected. We calculated the empirical p-values for each score as the fraction of 10000 random gene groups of the same size with equal or lower score. The empirical p-value expresses the probability that a random gene group would have a similar score, and therefore, expresses how exceptional the result is. The p-values were thresholded with 10%, without multiplicity adjustment, since we only want to compare methods. The fraction of biclusters with scores less than the threshold for different methods is depicted in Figure 4.3. We also compared the recall ability of the methods, which was measured by the fraction of edges of the gene interaction network that were covered by all the biclusters.

The precisions are not very good for any of the methods. BiMDL+DS method had clearly the highest precision on disconnected gene pairs with

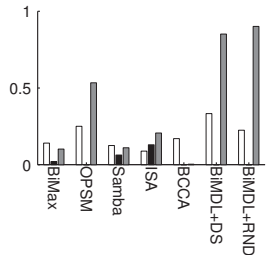


Figure 4.3. Precision of different methods are illustrated by the fraction of biclusters with statistically significant scores in fraction of disconnected gene pairs (white bar) and average distance of connected gene pairs (black bar). Recall is illustrated by the fraction of edges covered of the gene interaction network (gray bar). A higher bar is better.

OPSM and BiMDL+RND coming next. Other methods obtained more or less the expected value, which is 10% because the p-values were not adjusted for multiplicity. The precision of average distance is extremely low for all methods. The poor discrimination ability is likely caused by the low percentage (6.6%) of connected gene pairs in the pathway network. All methods have most likely found biclusters that are not explained by the pathway network.

Recall (gray bars) of the BiMDL methods were clearly the highest with close to perfect recall for both methods. OPSM obtained close to 50% recall, and the other methods had much smaller values. Overall, the BiMDL methods performed better than most of the other methods.

5. Discussion

We presented a biclustering algorithm that allows the detection of interesting regions of correlation, *i.e.*, sets of rows and columns that have high cosine correlation. The algorithm locates small regions of deviant expression values and extends these areas to biclusters of strong cosine correlation. This is done using a parameter-free model. Finally, possible overlap between biclusters, which in caused by extending the small regions to similar biclusters, in removed.

While the method is motivated and constructed for gene expression data, the model is likely to be suitable for other types of biological array measurement data.

Assessing the significance of biclusters has been previously overlooked, even though the search space of potential biclusters is large. We used a method that is capable of obtaining p-values for each bicluster and correcting them for multiple hypothesis testing.

We conducted experiments with the proposed biclustering method and compared it with existing methods. The first experiment tested the ability of the biclustering algorithms to extract biclusters that explain most of the local correlation within data matrix. The second test measured the precision and recall of the methods against known results, and the results, although not very decisive, indicated that the proposed BiMDL methods perform better than the existing methods. Furthermore, the proposed methods were able to capture most of the correlation within local neighborhood of the biclusters. The other methods were not able to do that as clearly. Therefore, as a conclusion from the experiments, the proposed biclustering algorithm BiMDL+DS is an improvement in finding local correlation structures in the data matrix.

Acknowledgments

We thank Heikki Mannila for help with the seed generation method and commenting the article. Jussi Taipale and Teemu Kivioja for helping to construct a biologically relevant model. We thank Teemu Roos for advice in Minimum Description Length.

Bibliography

Ben-Dor, A., Chor, B., Karp, R., and Yakhini, Z. (2003). Discovering local structure in gene expression data: The order-preserving submatrix problem. *Journal of Computational Biology*, **10**(3–4), 373–384.

Bhattacharya, A. and De, R. K. (2009). Bi-correlation clustering algorithm for determining a set of co-regulated genes. *Bioinformatics*, **25**(21), 2795–2801.

Forbes, C., Evans, M., Hastings, N., and Peacock, B. (2010). *Statistical Distributions*. John Wiley & Sons, Inc., 4 edition.

Freeman, W. T., Perona, P., Perona, P., and Freeman, W. (1999). A factorization approach to grouping. In *in European Conference on Computer Vision*, pages 655–670.

Grünwald, P. (2007). *The Minimum Description Length Principle*. MIT Press.

Hanhijärvi, S. (2011). Multiple hypothesis testing in pattern discovery. *Lecture Notes in Computer Science*, **6926/2011**, 122–134.

Hanhijärvi, S., Ojala, M., Vuokko, N., Puolamäki, K., Tatti, N., and Mannila, H. (2009). Tell me something I don't know: randomization strategies for iterative data mining. In *KDD '09: Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 379–388, New York, NY, USA. ACM.

Huber, G. (1982). Gamma function derivation of n-Sphere volumes. *The American Mathematical Monthly*, **89**(5), 301—302.

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

Knuth, D. E. (1997). *The art of computer programming. Volume 1, Fundamentals algorithms*. Addison-Wesley, Reading, Mass.

Madeira, S. C. and Oliveira, A. L. (2004). Biclustering algorithms for biological data analysis: A survey. *IEEE Transactions on Computational Biology and Bioinformatics*, **1**(1), 24–45.

Marsaglia, G. (1972). Choosing a point from the surface of a sphere. *The Annals of Mathematical Statistics*, **43**, 645–646.

Ojala, M., Vuokko, N., Kallio, A., Haiminen, N., and Mannila, H. (2009). Assessing data analysis results on real-valued matrices. *Statistical Analysis and Data Mining*, **2**, 209–230.

Prelić, A., Bleuer, S., Zimmerman, 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**(3), 1122–1129.

Puolamäki, K., Hanhijärvi, S., and Garriga, G. C. (2008). An approximation ratio for biclustering. *Information Processing Letters*, **108**(2), 45–49.

Rissanen, J. (1989). *Stochastic Complexity in Statistical Inquiry*. World Scientific.

Tanay, A., Sharan, R., and Shamir, R. (2002). Discovering statistically significant biclusters in gene expression data. *Bioinformatics*, **1**(1), 1–9.

Wille, A., Zimmermann, P., Vranová, E., Fürholz, A., Laule, O., Bleuler, S., Hennig, L., 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).