Efficient query-driven biclustering of gene expression data using Probabilistic Relational Models

Tim Van den Bulcke, Hui Zhao, Kristof Engelen, Tom Michoel, Bart De Moor, Kathleen Marchal*

August 5, 2008

Abstract

Biclustering is an increasingly popular technique to identify gene regulatory modules that are linked to biological processes. We describe a novel method, called *Pro*Bic, that was developed within the framework of Probabilistic Relational Models (PRMs). *Pro*Bic is an efficient biclustering algorithm that simultaneously identifies a set of potentially overlapping biclusters in a gene expression dataset and which can be used both in a query-driven and a global setting.

The model naturally deals with missing values. Robust sets of biclusters are obtained due to the explicit modeling of noise. The maximum likelihood solution is approximated using an Expectation-Maximization strategy. *Pro*Bic was applied to various synthetic gene expression datasets and the results for synthetic data confirmed that *Pro*Bic can successfully identify biclusters under various noise levels, overlap and missing values in both the query-driven and global setting. Additional expert knowledge can be introduced through a number of prior distribution parameters. Default settings were shown to be applicable for a wide range of different datasets.

Our results on synthetic data show that PRMs can be used to identify overlapping biclusters in an efficient and robust manner.

keywords: biclustering, probabilistic relational model, gene expression, regulatory module, expectation-maximization.

*corresponding author

1 Background

The reconstruction of the cellular signaling pathways is one of the foremost challenges in systems biology (Greenbaum et al., 2001). While a large amount of high throughput *'omics* data are available, the reconstruction of this signaling network still remains a highly underdetermined problem. Currently, insufficient data is available to uniquely identify all the interactions and their parameters in the model. However, regulatory networks exhibit a modular organization (Tanay et al., 2004), an aspect that is successfully exploited in a number of biclustering and module inference methods (Van den Bulcke et al., 2006). Module inference greatly reduces the number of required parameters, while preserving the essential characteristics of the network.

Biclustering was originally introduced by (Cheng and Church, 2000) in the context of gene expression data. Biclustering algorithms perform simultaneous clustering in both the gene and array dimensions and results in the simultaneous identification of gene sets that are coexpressed and the conditions under which they are coexpressed (Madeira and Oliveira, 2004). The drawback of most existing biclustering algorithms is that they do not easily allow for the incorporation of additional biological knowledge. Molecular biologists might be interested in knowing whether a certain subset of genes is indeed coexpressed or in knowing which other genes are clustering together with a predefined set of *query genes*.

Probabilistic Relational Models (PRMs) (Koller and Pfeffer, 1998; Friedman et al., 1999; Getoor et al., 2001b) were recently developed as an extension of Bayesian networks to the relational domain. PRMs have since been successfully applied in wide range of settings such as hypertext classification (Getoor et al., 2001a), collaborative filtering (Newton and Greiner, 2004), ecosystem analysis (D'Ambrosio et al., 2003) and genomics (Segal et al., 2003a,b, 2001, 2003c). As we will show in Section 3, PRMs are also well suited for both query-driven and global biclustering of gene expression data.

In this paper we present the *Pro*Bic model, an alternative approach to identify overlapping biclusters in gene expression data using Probabilistic Relational Models, that

is able to incorporate biological prior information in the form of a set of *query genes*. *Pro*Bic is an efficient biclustering method that simultaneously identifies a number of potentially overlapping biclusters and allows biological prior knowledge (by means of a list of *query genes*) to be incorporated. Moreover, *Pro*Bic was designed such that it is easily extensible towards additional data types.

2 Methods

2.1 The *Pro*Bic model

Figure 1 presents an overview of the *Pro*Bic (***Pro***babilistic ***Bic***lustering) model. The model contains three *classes*: *Gene*, *Array* and *Expression*. For each class, a number of specific gene, array and expression *objects* exist (denoted by g, a and e respectively). Analogously, uppercase letters G, A, E denote the complete set of genes, arrays and expression objects. Each object g and a in the *Gene* and *Array* classes respectively, have a number of binary *atrributes* B_b that indicate for each gene(array) object if that gene(array) is part of a bicluster b or not. The *Array* class has an additional attribute ID that uniquely identifies each individual array object a. Finally, each object e of the class *Expression* has a single numeric attribute $e.level$ that contains the individual expression level value for a specific gene and array. These gene and array objects are specified by two *reference slots* (comparable to *foreign keys* in a database) $e.gene$ and $e.array$ that point to these specific gene and array objects.

The gene-bicluster $g.B_b$ labels (over all biclusters b) and the array-bicluster labels $a.B_b$ are hidden variables in the model. The model is parameterized with a number of Normal distribution parameters: one set of parameters $(\mu_{a,b}, \sigma_{a,b})$ for each array-bicluster combination (a, b). The conditional probability distribution $P(e.level|\ldots)$ and the prior distributions $P(\mu_{a,b}, \sigma_{a,b})$, $P(a.B_b)$, $P(g.B_b)$ and $P(g.B)$ will be defined in Section 2.2.

When given a particular dataset instance, the *Pro*Bic model specifies a joint probability distribution (JPD) over this particular instance by implicitly defining a ground Bayesian network over this instance. Figure S1 (Suppl. Mat.) illustrates how the dataset instance and the *Pro*Bic model implicitly generate a ground Bayesian network (GBN). Each node in the GBN corresponds to the attribute of a specific object in the *Pro*Bic model. The joint probability distribution is similar to that of a Bayesian network: it is the product over all nodes $x.A$ in the ground Bayesian network of the conditional probability distribution (CPD) belonging to each node, given the values

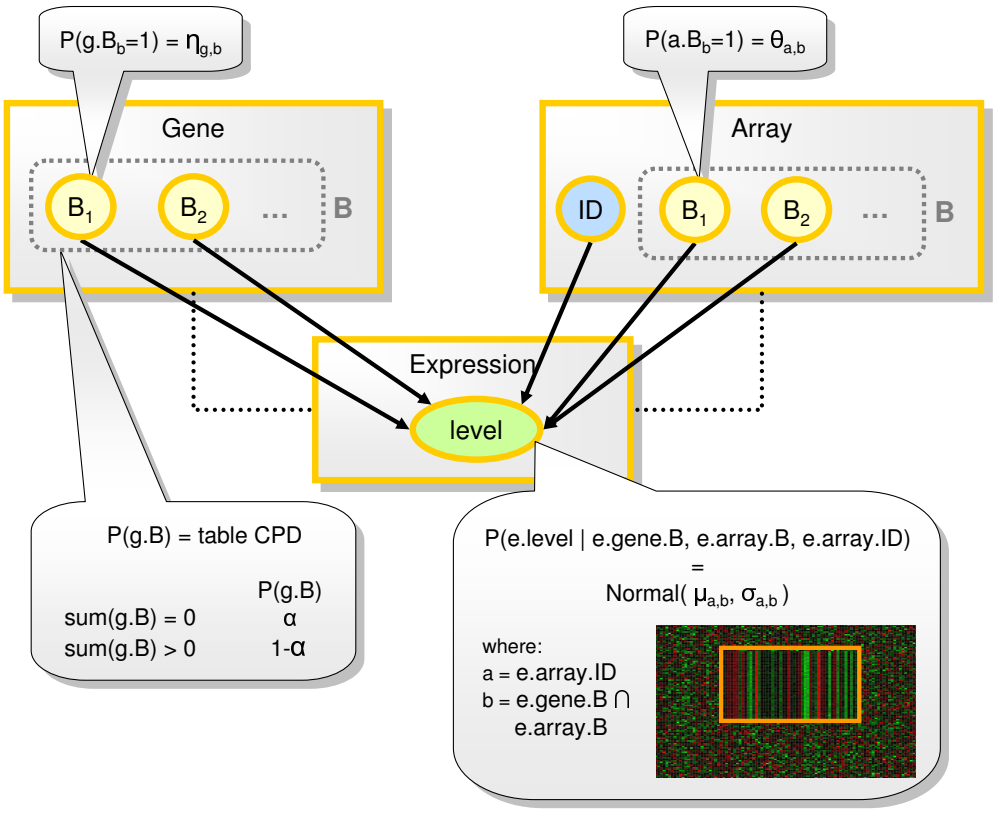


Figure 1: Schematic overview of the *Pro*Bic model and the conditional probability distributions of each of the attributes. The three classes are *Gene*, *Array* and *Expression*. *Expression* has one attribute *level* that contains the expression level and it contains two slots (similar to foreign keys in a relational database) *gene* and *array* that indicate the gene and the array for which the expression level is measured. *Gene* and *Array* each have a number of boolean attributes B_b that represent the presence or absence of a gene/array in a bicluster b. For notational convenience, these attributes are grouped in a vector $B = \{B_1, B_2, ..., B_k\}$. The class *Array* has an extra attribute ID that uniquely identifies each array. The conditional probability distribution $P(e.level|e.gene.B, e.array.B, e.array.ID)$ is modeled as a set of Normal distributions (one for each array-bicluster combination) each with their own mean $\mu_{a,b}$ and standard deviation $\sigma a, b$. A number of prior distributions $P(a.B_b)$, $P(g.B_b)$ and $P(g.B)$ allow biologists to guide the model towards biclusters of interest if necessary.

that were assigned to its parents.

After multiplication of the JPD with the different prior distributions (see Section 2.2 for a definition of the prior distributions), the following posterior is obtained for the *Pro*Bic model:

$$
\begin{aligned}
posterior \quad = \quad & \prod_{a \in A} \prod_{b \in B} P(\mu_{a,b}, \sigma_{a,b}) \cdot & (1) \\
& \prod_{e \in E} P(e.level | e.gene.B, e.array.B, e.array.ID) \cdot & (2) \\
& \prod_{b \in B} \prod_{a \in A} P(a.B_b) \cdot \prod_{b \in B} \prod_{g \in G} P(g.B_b) \cdot \prod_{g \in G} P(g.B) & (3)
\end{aligned}
$$

where the $\mu_{a,b}$ and $\sigma_{a,b}$ parameters are the model parameters and the binary bicluster assignment labels $a.B_b$ and $g.B_b$ are the hidden variables in the model. Rather than calculating the full posterior, which is computationally intractable, we are interested in finding the maximum likelihood of the posterior w.r.t. the model parameters and also in identifying the hidden variables (i.e. the gene-bicluster and array-bicluster assignments $g.B$ and $a.B$) that are associated with this solution. The exact calculation of this maximum likelihood solution is however still computationally intractable. Several approximating strategies can be applied to maximize this posterior, ranging from Monte Carlo approaches such as Gibbs sampling, over variational approaches to more general strategies such as genetic algorithms and simulated annealing.

We apply a *hard assignment* Expectation-Maximization (EM) approach (Dempster et al., 1977) to find this maximum likelihood solution. EM finds a local maximum of the posterior by starting with an initial bicluster and then iteratively maximizing the posterior over the model parameters and over the hidden variables until convergence. *Hard assignment* means that the assignment for the hidden variables during the iterations is not probabilistic, but that the value with the highest probability is assigned in each iteration.

2.2 The *Pro*Bic priors and the CPDs

In the following sections, each of the priors and the CPDs are formally defined and their effect on the model is discussed.

2.2.1 $P(e.level|e.gene.B, e.array.B, e.array.ID)$

This is the main CPD for the biclustering model and it consists of two individual factors:

$$\begin{aligned} P(e.level|e.gene.B, e.array.B, e.array.ID) &= P_1(e.level|e.gene.B, e.array.B, e.array.ID). \\ & \quad P_2(e.level|e.gene.B, e.array.B, e.array.ID) \end{aligned}$$

This first factor $P_1(e.level|\ldots)$ covers a number of separate cases. Let us first evaluate the case where an expression level is part of none of the biclusters ($g.B \bigcap a.B = \emptyset$). The probability $P_1(e.level|\ldots)$ is then described as a Normal distribution with a set of parameters $(\mu_a^{bgr}, \sigma_a^{bgr})$ per individual array a. The parameters of the background distribution are *fixed* and derived from the dataset using a robust estimation of the background distributions. The background estimation is performed using the smallest quartile range normalization (SQRN) method, as described in (Gu and Liu, 2008).

Next, in case there is no overlap between the different biclusters, every expression level is part of exactly one bicluster. The probability $P_1(e.level|\ldots)$ is again described as a Normal distribution with parameters $(\mu_{a,b}, \sigma_{a,b})$, where a indicates a particular array and b a particular bicluster. We now formally define the probability for an expression level that is part of a single bicluster b as:

$$\begin{aligned} P(e.level|e.gene.B, e.array.B, e.array.ID) &= P_1^*(e.level|iset_B(e) = \{b\}, e.array.ID = a) \\ &= P_1^*(e.level|\mu_{a,b}, \sigma_{a,b}) \\ &= \frac{1}{\sigma_{a,b}\sqrt{2\pi}} \exp[-\frac{(e.level - \mu_{a,b})^2}{2\sigma_{a,b}^2}] \end{aligned}$$

where the probability $P_1^*(e.level|\mu_{a,b}, \sigma_{a,b})$ is defined as the probability of an expression level belonging to a single bicluster b and where the following notation was introduced: the set $iset_B(e)$ is the set of biclusters of which the expression level e is part of or more formally, it is the *set* of bicluster indices for which the corresponding element

in the intersection of the binary gene-bicluster assignment vector $e.gene.B$ and the array-bicluster assignment vector $e.array.B$ is not 0.

Finally, overlap between two or more biclusters can in principle be defined totally independent of the conditional probabilities described in the previous case of no overlap. The overlap can for example be modeled as the sum, the product, the average, the minimum or maximum of the individual bicluster contributions. We will define one such overlap model here that reflects a particular biological meaning of 'overlap' and secondly, it leads to very efficient computations in the different EM steps. The biological and computational properties are further examined in Sections 3 and 4. The probability of an expression level in the overlap region is formally defined as the geometric mean of the separate bicluster probabilities:

$$P(e.level|e.gene.B, e.array.B, e.array.ID) = \prod_{\substack{b\in \\ iset_B(e)}} P_1^*(e.level|\mu_{a,b}, \sigma_{a,b})^{1/\#iset_B(e)}$$

where $\#iset_B(e)$ is the number of elements in the set $iset_B(e)$.

So the probability for an expression level to be in the overlap of two biclusters is the (geometric) mean of the individual probabilities of belonging to the single biclusters. This definition leads to at least one type of biologically interesting overlap structure in which the expression values are similar to the expression values in the corresponding individual biclusters (per array).

For notational convenience, we will now describe a general notation covering all three cases for the factor $P_1(e.level|\ldots)$:

$$P_1(e.level|e.gene.B, e.array.B, e.array.ID) = \prod_{\substack{b\in \\ iset_B(e)}} P_1^*(e.level|\mu_{a,b}, \sigma_{a,b})^{1/\#iset_B(e)}$$

For including the background in this notation, we introduce a bicluster with index -1 that describes the background as a special bicluster. The parameters ($\mu_{bgr} = \mu_{a,-1}, \sigma_{bgr} = \sigma_{a,-1}$) of this special 'bicluster' are fixed and this background bicluster can not overlap with other biclusters.

The meaning of the product over all biclusters in the intersection for an expression level e also changes slightly: $\prod_{b \in iset_B(e)}$ now covers two cases:

- $iset_B(e) = \emptyset$: background distribution, so the product is over the set $b \in [-1]$
- $iset_B(e) \neq \emptyset$: bicluster distribution, the product is over the set of biclusters in the intersection (excluding b=-1).

The second factor $P_2(e.level|\ldots)$ is introduced to reduce the model complexity. The maximum likelihood solution explains as much of the variation in the expression data as possible by assigning expression values to biclusters and choosing the optimal model parameters. However, it is usually more likely for a set of expression values to be in a separate bicluster distribution rather than in the background (see also Figure 2), as this introduces additional degrees of freedom to the model (namely one additional parameter set $(\mu_{a,b}, \sigma_{a,b})$). Therefore, an additional penalty factor $P_2(e.level|\ldots)$ is defined, such that it effectively only allows a set of expression values to be in a bicluster if they are on average N times more likely to be in a bicluster distribution than in the background distribution. The factor P_2 decomposes similarly to P_1, leading to:

$$P_2(e.level|e.gene.B, e.array.B, e.array.ID) \quad = \quad \prod_{b \in iset_B(e)} P_2^*(e.level|b)^{\frac{1}{\#iset_B(e)}}$$

where $P_2^*(e.level|b) = \pi_{bgr}$ if the expression level is in the background ($b = -1$) and $P_2^*(e.level|b) = \pi_{bicl}$ if it is in a bicluster. A set of expression values $E_{s,1}$, as shown in Figure 2, will be assigned to bicluster 1 if the following equation holds:

$$\prod_{e \in E_{s,1}} P^*(e, bicl). \prod_{e \in E_{s,1}} \pi_{bicl} > \prod_{e \in E_{s,1}} P^*(e, bgr). \prod_{e \in E_{s,1}} \pi_{bgr} \Leftrightarrow \prod_{e \in E_{s,1}} \frac{\pi_{bicl}}{\pi_{bgr}} > \prod_{e \in E_{s,1}} \frac{P^*(e, bgr)}{P^*(e, bicl)}$$

This implies that the ratio $\frac{\pi_{bicl}}{\pi_{bgr}}$ indicates how many times more likely it must be on average that an expression value is part of the bicluster distribution compared to part of the background distribution before such a set E_s is actually added to the bicluster.

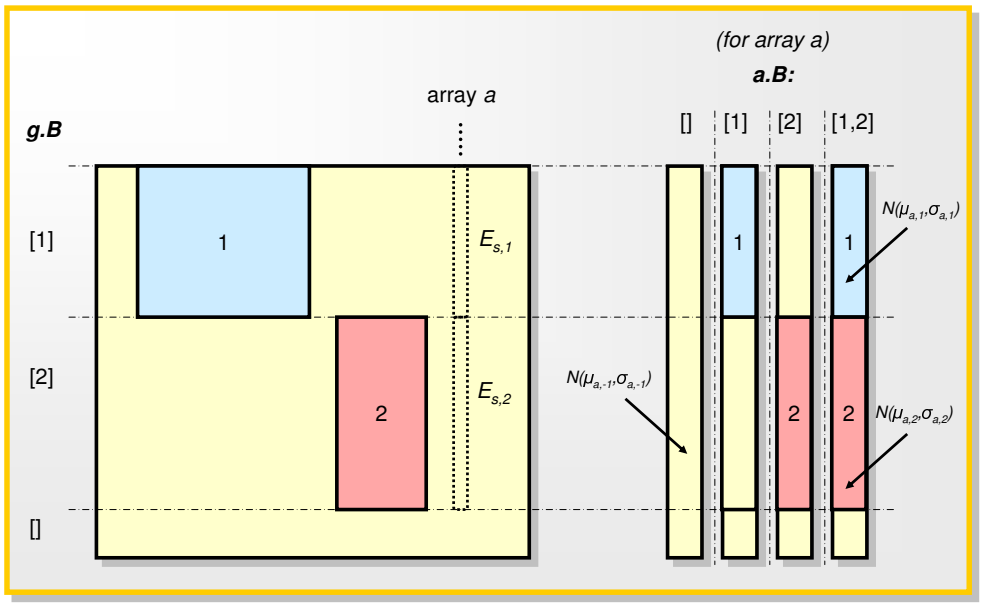


Figure 2: Illustration of the Expectation-AB step. For a particular array a, the log-posterior is calculated for all possible $a.B$ assignments. At the right hand side the situation is shown for array a for each of the $a.B$ assignments. The the log-posterior is the sum of a number of separate contributions from each of the expression levels in the colored blocks.

2.2.2 $P(g.B_b)$ and $P(a.B_b)$

The prior probability for assigning or not assigning a particular gene g to a bicluster b, $P(g.B_b = \{0, 1\})$, is defined as:

$$P(g.B_b = 1) = \eta_{g,b}$$

Note that this equation indirectly also defines $P(g.B_b = 0) = 1 - \eta_{g,b}$.

By specifying a prior for every gene-bicluster combination, the model is able to perform *query-driven* biclustering: a specific set of genes, called *query genes*, is considered to be more likely part of a bicluster based on prior expert knowledge and the model will then identify other genes with similar expression profiles and bicluster them together with these query genes.

Similarly, the prior probability $P(a.B_b)$ for assigning specific arrays to a bicluster is defined as:

$$P(a.B_b = 1) = \theta_{a,b}$$

The individual values for $\bar{\bar{\eta}}$ and $\bar{\bar{\theta}}$ are considered as prior knowledge to the *Pro*Bic model.

2.2.3 $P(g.B)$

Whereas $P(g.B_b)$ defines the *individual* probability that a gene is part of a *specific* bicluster, the distribution $P(g.B)$ determines the *overall* probability that a particular gene g is in *any* bicluster. More formally:

$$P(g.B) = \begin{cases} \alpha, & \text{if } g.B = \mathbf{0} \\ \beta = 1 - \alpha, & \text{else} \end{cases}$$

This prior definition allows another type of biologically interesting prior knowledge: biologists are often interested in specific pathways with a relatively small amount of genes. By increasing the ratio $\frac{\beta}{\alpha}$, biologists can in fact *zoom in* on a bicluster of interest and obtain biclusters with a decreasing amount of genes. α represents the prior probability that a gene is part of no bicluster, and β represents the probability that a gene is part of *one or more* biclusters.

The prior per individual gene $P(g.B_b)$ cannot express such prior probabilities, as it would be counted multiple times for genes in the overlap between biclusters. One of the undesired side-effects would then be that overlap between biclusters is avoided in models with low prior values for some $P(g.B_b)$.

2.3 Normal prior distributions $P(\bar{\bar{\mu}}, \bar{\bar{\sigma}})$

A prior distribution is defined on each of the Normal distributions with similar decomposition as the expression level probabilities $P_1(e.level|\ldots)$. This approach leads to a decomposition of the posterior in the EM algorithm into independent factors, thereby greatly reducing the computational complexity for the maximization step in the EM

algorithm (see Section 3.1).

$$\prod_{a \in A} \prod_{b \in B} P(\mu_{a,b}, \sigma_{a,b})$$

Note that the product $\prod_{b \in B}$ ranges over all the biclusters *including* the background distribution.

Different types of distributions can be used for parameterizing this prior. Two particularly interesting choices of prior are for example a Normal-Inverse-χ^2 distribution and a Normal distribution with a pseudocount. Throughout this paper, the Normal-Inverse-χ^2 distribution will be used as prior distribution. The Normal-Inverse χ^2 distribution (Gelman et al., 2003) is distributed as follows:

$$\mu|\sigma^2 \sim N(\mu_0, \sigma^2/\kappa_0)$$
$$\sigma^2 \sim Inv - \chi^2(\nu_0, \sigma_0^2)$$

and is parameterized by parameters $(\mu_0, \sigma_0, \kappa_0, \nu_0)$. This type of prior is particularly flexible for the purpose of biclustering (see also Section 4). It can be shown that the posterior mean is a weighted average of the prior mean and the sample mean with weights proportional to κ_0 and the sample size respectively.

3 Algorithm

The core algorithm is a hard assignment Expectation-Maximization (EM) algorithm. In each iteration, a maximization step and an expectation step are executed until the algorithm converges:

- Maximization step: maximize over all the model parameters $(\bar{\bar{\mu}}, \bar{\bar{\sigma}})$, keeping the hidden variables (i.e. the gene- and array-bicluster assignments) fixed.
- Expectation step: find the expected values for the hidden variables, keeping the current model parameters fixed.

In the Expectation step, all gene-bicluster labels $g.B$ and all array-bicluster labels $a.B$ need to be reassigned and this step is still computationally intractable. However, the expectation step can be split into two substeps. In each of these steps, either only the gene-bicluster labels $g.B$ are reassigned or only the array-bicluster labels $a.B$. This approach uses the relational structure in the model and leads to a decomposition of the posterior in factors which can be optimized independently either per individual gene or array, depending on the substep.

- E-step 1: maximize posterior independently per array a w.r.t. the array-bicluster labels $a.B$, keeping both the model parameters $(\bar{\bar{\mu}}, \bar{\bar{\sigma}})$ and $g.B$ fixed for each gene g.
- E-step 2: maximize posterior independently per gene g w.r.t. the gene-bicluster labels $g.B$, keeping both the model parameters and the array-bicluster labels $a.B$ fixed for all arrays a.

The core algorithm can be applied in two different modes. In *mode A*, the EM algorithm optimizes all biclusters simultaneously. In *mode B*, the EM algorithm is executed multiple times and optimized an individual bicluster in each run according to the following scheme: when optimizing bicluster i, all previously identified biclusters $(1 \,..\, (i-1))$are kept fixed. In *mode B*, assuming that there are I iterations until convergence and that each of the E-steps is followed by an M-step, the time complexity

of the complete algorithm is: $\mathcal{O}(I.A.G.B^2)$ where G represents the number of genes, A the number of arrays and B the number of biclusters. In other words, the average time complexity is linear in the number of genes and arrays and quadratic in the number of biclusters. For all presented examples in Section 4, both modes lead to the same results, indicating that the approximation of mode B does not seem to affect the quality of the maximum likelihood solution.

In the query-driven mode, a single initial expectation substep 1 (reassigning array-bicluster labels) is executed prior to the full EM algorithm, in order to filter out the arrays that are not differentially expressed for the set of query genes. The global bicluster identification starts with a phase where only expectation step 2 (reassigning genes to biclusters) and the maximization step are executed until convergence, prior to running the full EM algorithm. This approach avoids 'race conditions' between removing genes and removing arrays from the initial global bicluster and by consequence leads to more robust bicluster identifications.

Algorithm 1 **[pseudocode* Pro*Bic algorithm]*
for b from 1 to maxNumberOfBiclusters:

- *initialize bicluster b (either as the query genes over all conditions or as the complete dataset)*
- *while not converged:*
  - *maximization: maximize the posterior w.r.t.* $(\mu_{a,b}, \sigma_{a,b})$ *over all arrays a*
  - *expectation:*
    - *E-step 1: maximize posterior w.r.t.* $a.B_b$ *label for bicluster b (for each array a)*
    - *E-step 2: maximize posterior w.r.t.* $g.B_b$ *label for bicluster b (for each gene g)*

3.1 Maximization step

In the maximization step, all hidden variables (the $g.B$ and $a.B$ labels) are known and fixed. The posterior is maximized over all model parameters $(\mu_{a,b}, \sigma_{a,b})$. Note that due to our particular choice of overlap model, there are no separate parameters for the overlap regions between biclusters: the parameters $(\mu_{a,b}, \sigma_{a,b})$ implicitly define the overlap. Maximizing the $log(posterior)$ leads to the following derivation:

$$\begin{aligned}
argmax_{\bar{\mu},\bar{\sigma}}\ log(posterior) &= argmax_{\bar{\mu},\bar{\sigma}} \sum_{a\in A}\sum_{b\in B} logP(\mu_{a,b},\sigma_{a,b}) + \\
&\quad \sum_{e\in E} logP(e.level|e.gene.B, e.array.B, e.array.ID) + C_1 \\
&= argmax_{\bar{\mu},\bar{\sigma}} \sum_{a\in A}\sum_{b\in B} logP(\mu_{a,b},\sigma_{a,b}) + \\
&\quad \sum_{a\in A}\sum_{e\in E_{\cdot a}} \sum_{\substack{b\in \\ iset_B(e)}} \frac{1}{\#iset_B(e)}[logP_1^*(e.level|\mu_{a,b},\sigma_{a,b}) + C_2] \\
&= argmax_{\bar{\mu},\bar{\sigma}} \sum_{a\in A}\sum_{b\in B} logP(\mu_{a,b},\sigma_{a,b}) + \\
&\quad \sum_{a\in A}\sum_{b\in B} \sum_{\substack{e\in E_{\cdot a}: \\ b\in iset_B(e)}} \frac{1}{\#iset_B(e)}[logP_1^*(e.level|\mu_{a,b},\sigma_{a,b})]
\end{aligned}$$

where $E_{\cdot a}$ is the subset of expression values for array a and C_1 and C_2 are constant factors from Equation 1 that do not depend on the parameters $(\mu_{a,b}, \sigma_{a,b})$.

Given that the parameters $(\mu_{a,b}, \sigma_{a,b})$ are independent per array and per bicluster, the above expression can be maximized independently per array and per bicluster. This leads to the following optimization problem per array a and per bicluster b:

$$argmax_{\mu_{a,b},\sigma_{a,b}} logP(\mu_{a,b},\sigma_{a,b}) + \sum_{\substack{e\in E_{cdota}: \\ b\in iset_B(e)}} \frac{1}{\#iset_B(e)} logP_1^*(e.level|\mu_{a,b},\sigma_{a,b})$$

An analytical solution can be derived for the above equation for both the cases where the prior distribution is a Gaussian pseudocount and a Normal-Inverse-χ^2 distribution.

This leads to very efficient computations in the maximization step.

3.2 Expectation-step 1

In this step, the model parameters $(\bar{\bar{\mu}}, \bar{\bar{\sigma}})$ and the gene-bicluster labels $g.B$ are fixed and the posterior is maximized for all possible $a.B$ assignments:

$$
\begin{aligned}
argmax_{A.B} log(posterior) &= argmax_{A.B} \sum_{a\in A}\sum_{b\in B} logP(\mu_{a,b},\sigma_{a,b}) + \\
&\quad \sum_{e\in E}\sum_{b\in iset_B(e)} \frac{[logP_1^*(e.level|a,b) + logP_2^*(e.level|a,b)]}{\#iset_B(e)} + \\
&\quad \sum_{b\in B}\sum_{a\in A} logP(a.B_b) + \sum_{b\in B}\sum_{g\in G} logP(g.B_b) + \sum_{g\in G} logP(g.B) \\
&= argmax_{A.B} \\
&\quad \sum_{a\in A}\sum_{e\in E_a}\sum_{b\in iset_B(e)} \frac{[logP_1^*(e.level|a,b) + logP_2^*(e.level|a,b)]}{\#iset_B(e)} + \\
&\quad \sum_{a\in A}\sum_{b\in B} logP(a.B_b)
\end{aligned}
$$

where capital A stands for the set of all arrays a.

The above expression can be optimized independently per array. For every array a, this leads in principle to the evaluation of 2^B possible $a.B$ assignments (when there are B biclusters). Unfortunately, the expression cannot be optimized independently per bicluster, as $iset_B(e)$ indirectly depends on the $a.B$ assignment (remember that $iset_B(e)$ is the intersection set between the $array.B$ and $gene.B$ vectors of the expression object e).

However, an approximation can be used to avoid a complete evaluation of all possible $a.B$ assignments. In case of no overlap (illustrated in Figure 2), we see on the right side all the different $a.B$ assignments. In case of no overlap, the bicluster contributions are independent and we can therefore deduce the score for $a.B = [1,2]$ from the scores of $a.B = []$, $a.B = [1]$ and $a.B[2]$: $score([1,2]) = score([1]) - score([]) + score([2]) - score([])$. This reasoning is still approximately valid when the amount of overlap be-

tween biclusters is relatively small compared to the bicluster sizes. We generalize this approach to the overlap case by applying an *Apriori*-like algorithm (Agrawal et al., 1993): starting at level 1, all single bicluster assignments for $a.B$ and their δ_{score}'s are evaluated. Iteratively, all next levels n are investigated where the n-bicluster assignments are evaluated: if all $\delta_{score}(set_{n-1})$'s are positive at level $n-1$, then $\delta_{score}(set_n)$ is evaluated at level n. The procedure stops when no more evaluations can be performed. For cases with large overlap between biclusters, the approximation might not hold anymore and thus the evaluation of a level n assignment may not be performed even if the score of that assignment would be higher than the level $n-1$ assignments. While the worse case computational complexity of the Apriori algorithm is $\mathcal{O}(2^B)$, it is in practice relatively low: in case of no overlap between the biclusters, the time complexity of this procedure is linear in the number of biclusters.

A significant additional increase in computation speed is achieved by carefully grouping genes and calculating certain sums upfront:

$$
\begin{aligned}
& argmax_{a.B} \sum_{\substack{gb \in \\ unq(G.B)}} \sum_{\substack{e \in E_a: \\ e.gene \in gb}} \sum_{\substack{b \in \\ iset_B(gb,a.B)}} \frac{logP^*(e.level|\mu_{a,b},\sigma_{a,b})}{\#iset_B(gb, e.array.B)} + \sum_{b \in B} logP(a.B_b) \\
= \; & argmax_{a.B} \sum_{\substack{gb \in \\ unq(G.B)}} \sum_{\substack{b \in \\ iset_B(gb,a.B)}} \frac{\sum_{\substack{e \in E_a: \\ e.gene \in gb}} logP^*(e.level|\mu_{a,b},\sigma_{a,b})}{\#iset_B(gb, a.B)} + \sum_{b \in B} logP(a.B_b) \\
= \; & argmax_{a.B} \sum_{\substack{gb \in \\ unq(G.B)}} \sum_{\substack{b \in \\ iset_B(gb,a.B)}} \frac{f_a(gb,a,b)}{\#iset_B(gb, a.B)} + \sum_{b \in B} logP(a.B_b)
\end{aligned}
$$

where the function $f_a(gb, a, b)$ represents the set of precalculated values and where $unq(G.B)$ is the set of all unique $g.B$ vectors *for the current model* (e.g. for the model in Figure 2, $unq(G.B) = [1], [2], []$). A single evaluation for $a.B = ab$ has now become a simple (weighted) sum of some precalculated values.

3.3 Expectation-step 2

In this step, the model parameters $(\bar{\bar{\mu}}, \bar{\bar{\sigma}})$ and the array-bicluster labels $a.B$ are fixed and the posterior is maximized for all possible $a.B$ assignments. By applying a completely analogous derivation as for step 1, a decomposition per gene can be achieved of the optimization problem. A similar Apriori approach and upfront calculation strategy can be applied to increase computational efficiency, leading to the following set of optimization problems per gene g:

$$argmax_{g.B} \sum_{\substack{ab \in \\ unq(A.B)}} \sum_{\substack{b \in \\ iset_B(g.B,ab)}} \frac{f_g(g, ab, b)}{\#iset_B(g.B, ab)} + \sum_{b \in B} logP(g.B_b) + logP(g.B)$$

4 Results

An extensive evaluation of the *Pro*Bic algorithm was performed on synthetic data to investigate the behavior of the algorithm under various parameter settings and input data. We tested 1) the robustness of the algorithm w.r.t. noise; 2) the algorithm's ability to handle missing values; 3) the algorithm's ability to incorporate expert knowledge in the form of query genes; 4) the extent to which overlapping biclusters can be identified; 5) the possibility to automatically determine the number of biclusters from the data; 6) the possibility to derive the optimal parameter settings for a specific dataset; 7) the speed of the algorithm.

4.1 Synthetic expression data

The synthetic expression data used in the following analyses is modeled according to the *Pro*Bic model assumptions. Background values were sampled from a Normal distribution $N(\mu_{bgr} = 0, \sigma_{bgr} = 1)$ for all conditions. The expression values e_{ga} that are part of a single bicluster b were sampled from the distributions $N(\mu_{a,b}, \sigma_{a,b})$ (a separate distribution was used per condition a and per bicluster b). The expression values in the overlap region between multiple biclusters, were modeled as the average of the values sampled from the distributions $N(\mu_{a,b}, \sigma_{a,b})$ over all overlapping biclusters b. The values for $\mu_{a,b}$ were chosen from a Normal distribution $N(0, \sigma_{\mu})$ (excluding the interval $[-\sigma_{\mu_{bgr}}/2, \sigma_{\mu_{bgr}}/2]$) where $\sigma_{\mu} = 1$ unless otherwise specified. The value $\sigma_{a,b}$ representing the level of 'noise' in the bicluster compared to the background and was defined separately for each of the analyses.

The synthetic expression data that was used in these analyses, is modeled similar to the *Pro*Bic model assumptions. Background values were sampled from a Normal distribution $N(\mu_{bgr} = 0, \sigma_{bgr} = 1)$ for all conditions. The expression values e_{ga} that are part of a single bicluster b were sampled from the distributions $N(\mu_{a,b}, \sigma_{a,b})$ (a separate distribution per condition a and per bicluster b). In case of overlap between multiple biclusters, the expression values were modeled as the average of all sampled values from

the distributions $N(\mu_{a,b}, \sigma_{a,b})$ over all overlapping biclusters b. The values for $\mu_{a,b}$ are chosen from a Normal distribution $N(0, \sigma_\mu)$ (excluding the interval $[-\sigma_{\mu_{bgr}}/2, \sigma_{\mu_{bgr}}/2]$). The value $\sigma_{a,b}$ represents the amount of 'bicluster noise' compared to the background noise $\sigma_{bgr} = 1$ and is defined separately for each of the following experiments.

4.1.1 Robustness against noise

A 500x200 (genes x arrays) synthetic dataset was constructed (as described in Section 4.1) with three 50x50 non-overlapping biclusters. The bicluster noise level was varied between 0 and 1 and the effect on the precision and recall was measured for both the genes and the arrays separately. The results are shown in Figure S2 (Suppl. Mat.).

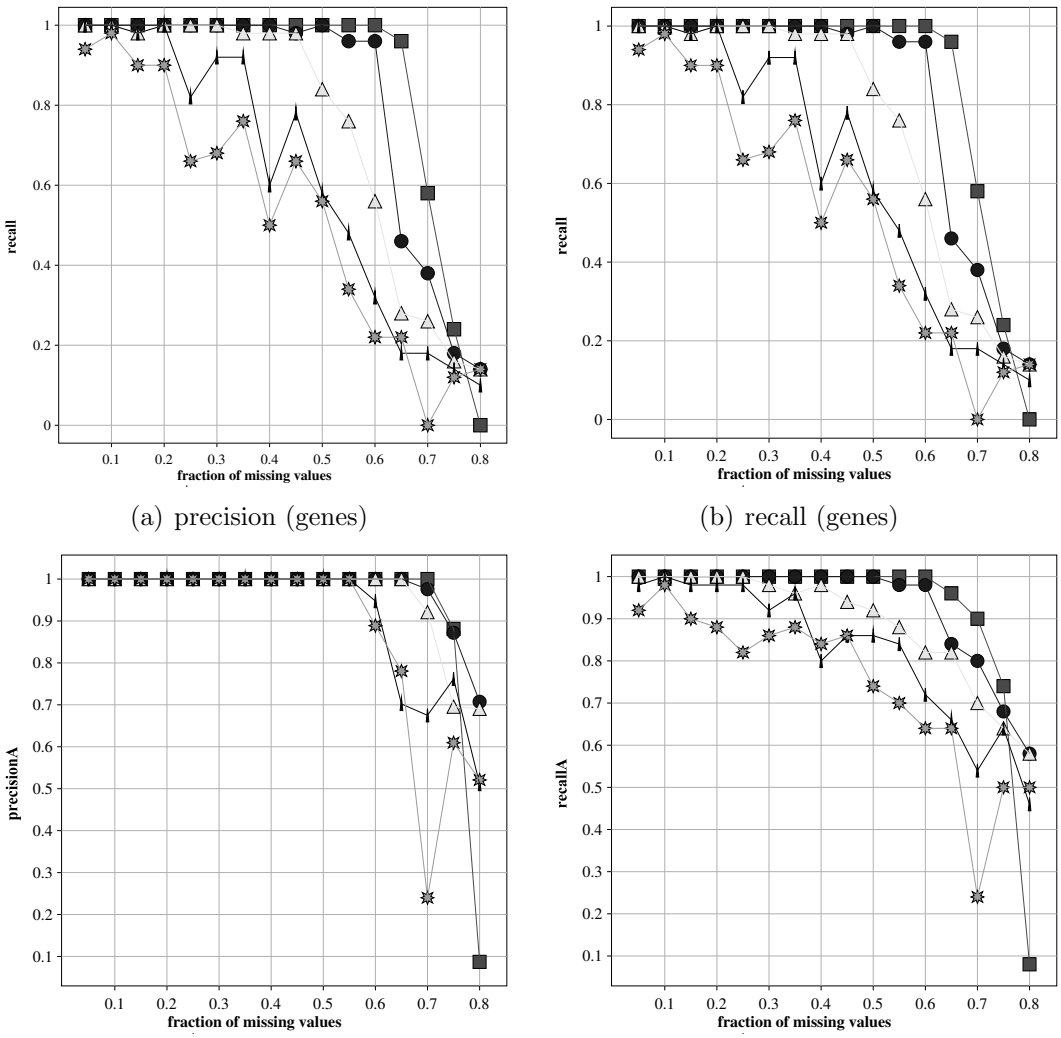
Figure S2(b) and (c) (Suppl. Mat.) show that perfect bicluster identification was achieved w.r.t. the genes for noise levels up to 70% of the background noise. When further increasing the noise levels, about 12% of the genes were no longer identified. The arrays which constitute a bicluster were perfectly recovered for noise levels up to 40% of the background noise (see Figure S2(d) and (e) (Suppl. Mat.)). With increasingly higher noise levels, the model gradually has more difficulties in distinguishing the bicluster distributions from the background distribution and the precision and recall in identifying the true arrays drop significantly.

4.1.2 Robustness against missing values

Also for this test we used a 500x200 synthetic dataset with three non-overlapping 50x50 biclusters, containing varying levels of missing values. The bicluster noise level was varied between 0 and 1. The effect of different levels of missing values on the precision and recall of finding the true bicluster genes and arrays was assessed for different levels of noise in the data . The results are shown in Figure 3.

Figure 3a) and c) show that for lower bicluster noise levels (< 0.5), the presence of up to 60% of missing values in the dataset does not interfer with the detection of the true bicluster genes and arrays (a precision and recall of about 100% is obtained for

both the genes and arrays). As can expected the model sensitivity to missing values increases with the noise level, but even in the presence of high noise levels (1.0), the presence of up to 20% missing values does not considerably deteriorate the algorithms recall and precision (recall levels of 0.9 and 0.88 for the genes and arrays respectively).



(a) precision (genes)

(b) recall (genes)

(c) precision (arrays)

(d) recall (arrays)

Figure 3: *Pro*Bic performance on a synthetic dataset with a varying percentage of missing values. The X-axes show the percentage of missing values (between 0% and 100%) in the dataset and the plots indicate the average precision (=TP/(TP+FP)) and recall (=TP/(TP+FN)) for the genes and arrays respectively of the three biclusters for different levels of bicluster noise. Noise levels: square: 0.2; circle: 0.4; triangle: 0.6; small triangle: 0.8; star: 1.0.

4.2 Overlapping biclusters

We also investigated to what extent *Pro*Bic is able to correctly identify biclusters that are overlapping. We tested both the query-driven and the global biclustering approach. For these experiments, 500x200 synthetic datasets with varying noise levels were constructed, each of which contained two 20x80 overlapping biclusters, overlapping for 2 genes and 20 arrays.

In the query-driven setting, the set of query genes are a subset of 5 genes for each of the biclusters. The model is initialized with a bicluster containing the set of query genes and all conditions. A prior $\eta_{g,b}$ is defined for each gene and each bicluster: for the query genes $\eta_{g,b} = \gamma$ and for the other genes $\eta_{g,b} = 0.5$ (note that the actual value of 0.5 has no effect on the result, as only the *ratio* between the query and non-query η's is important).

For all examined noise levels, the global approach identifies both overlapping biclusters correctly for lower noise levels (< 0.7). For increased noise levels, the global setting is no longer able to correctly identify the true overlapping biclusters, but the query-driven setting still does.

4.3 Query-driven biclustering

In the following experiments, the effect of defining a set of query genes is investigated. Synthetic 500x200 datasets were generated for varying noise levels containing two non-overlapping biclusters: a large 50x50 bicluster and a smaller 20x80 bicluster.

We examine the effect of setting the query genes: if no set of query genes is specified, the algorithm converges to the 50x50 bicluster as expected. However, if a biologist is interested in the smaller bicluster and specifies a subset (5 genes) of these as the set of query genes, the algorithm converges to the bicluster of interest to the biologist.

Next, the effect of a *noisy* query is investigated: what happens if one or more genes are selected as query genes, while these are not part of the real bicluster? This is a frequently occurring case in practice when a biologist selects genes, since the set of

query genes are usually *expected* (but not absolutely sure) to be in a bicluster. We simulated this situation in the following cases: a) a set of five bicluster genes and three non-bicluster genes are selected and b) five bicluster genes and ten non-bicluster genes are selected as query genes. In both these cases, the correct bicluster is still identified. Note that in the second case, the number of real bicluster genes is even smaller than the number of noisy query genes and the algorithm still correctly identifies the bicluster.

4.4 Determining the number of biclusters in the dataset

Most biclustering algorithms either identify only one bicluster (Bergmann et al., 2003; Dhollander et al., 2007) or they require the user to define the number of biclusters in the dataset upfront (Lazzeroni and Owen, 1999; Segal et al., 2003a). *Pro*Bic uses a data driven approach to select the optimal number of biclusters. The initial model is constructed with a large number of (empty) B_{max} biclusters. Iteratively, biclusters are added to the model, using the EM approach as described in Section 3. After the optimization of the Nth bicluster, the optimization only leads to additional empty biclusters from this point on. This point indicates that the optimal number of biclusters is reached.

4.5 Optimal model parameter settings

The *Pro*Bic biclustering model has several parameters that potentially affect the biclustering. While it is important for real life applications that parameters can be tuned, it is however also required that sensible default values can be defined that are broadly applicable on a wide range of datasets.

In this section we will investigate optimal parameter settings for each of the parameters of the *Pro*Bic model. Some of the parameter values can be derived from the data while for other parameters values can be defined where stable results are obtained across a wide range of datasets.

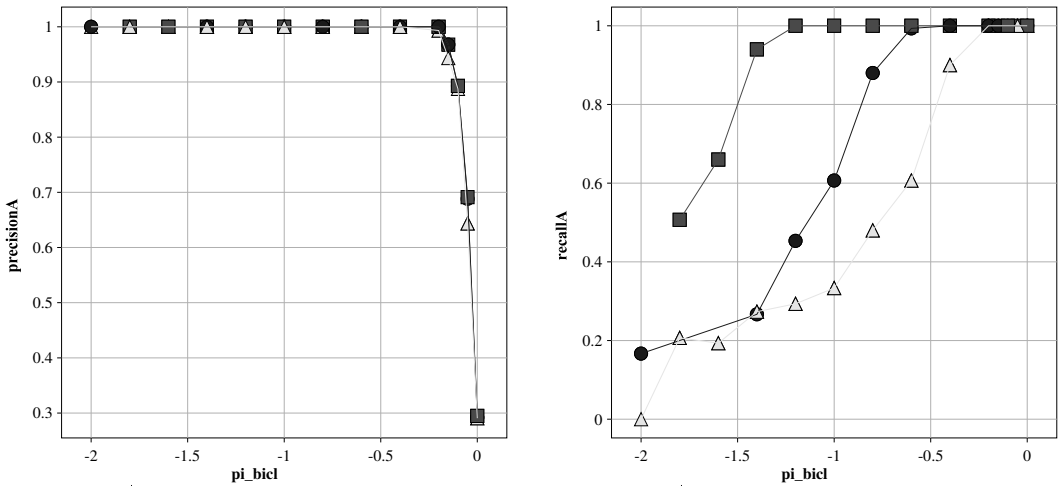
First, how to determine the optimal ratio ratio $\frac{\pi_{bicl}}{\pi_{bgr}}$, namely how many times more

likely should an expression value be part of a bicluster distribution compared to the background distribution before it is actually added to the bicluster. When assuming there exists one known bicluster in the dataset, the optimal ratio we can relatively easy be determined using the the following procedure: for every array a, the difference in score (called δ's in the remainder of the section) is calculated between the situations where either that array is part of the bicluster or where it is part of the background. We then solve a classification problem where the optimal threshold is determined for the δ's, minimizing the global error rate (= the product of the false positive rate and the false negative rate) for classifying the array as part of the background or of the bicluster. This procedure can easily be extended for more than one known bicluster.

In case there is a set of genes that are known to be coexpressed for some arrays (e.g. a set of operon genes) but the arrays in which they are coexpressed are unknown, a different procedure can be applied. Again the δ's are calculated for all arrays a. Since there are no known background/bicluster assignment labels for the arrays, we perform either a visual inspection of the sorted δ's or a statistical test for differential expression. A clear cutoff point between arrays with high and low δ's can often be determined visually, as can be seen in Figure S3 (Suppl. Mat.).

Next, the effect of the π_{bicl} (π_{bgr} ratio is investigated. Figure 4 shows how precision and recall for the arrays varies with respect to the parameter π_{bicl} (π_{bgr} is kept at a fixed level since only the ratio matters) for a synthetic 500x200 dataset with three non-overlapping 50x50 biclusters for a number of different noise levels. As π_{bicl} decreases, the precision is very high (precision=1) as only real bicluster arrays are kept in the identified bicluster. However at the expense of a lower recall for decreasing π_{bicl} values. In the range [-0.1;-0.2], both precision and recall remain high under various noise conditions.

The overall probability that a gene is in any bicluster (β) affects the size (in number of genes) of the biclusters. Figure 5 illustrates the effect of varying β on the bicluster size for a 500x200 dataset with one 100x100 bicluster that is composed out of three parts: expression values for 20 genes are sampled from $N(\mu_a, 0.2)$ distributions, 30



(a) precision (arrays) (b) recall (arrays)

Figure 4: The effect of π_{bicl} on the precision and recall for the arrays on a 500x200 synthetic dataset with 3 nonoverlapping 50x50 biclusters for varying levels of noise (square: 0.2; circle: 0.5; triangle: 1.0).

genes from $N(\mu_a, 0.5)$ and 50 genes from $N(\mu_a, 1.0)$. We expect that lower β values lead to a smaller bicluster (keeping only the most correlated genes) and this is confirmed in Figure 5. The number of genes in the bicluster decreases successively from 100 to 50 to 20 for decreasing β. We confirmed that the identified bicluster indeed corresponded to the original 100x100, 50x100 and 20x100 biclusters respectively.

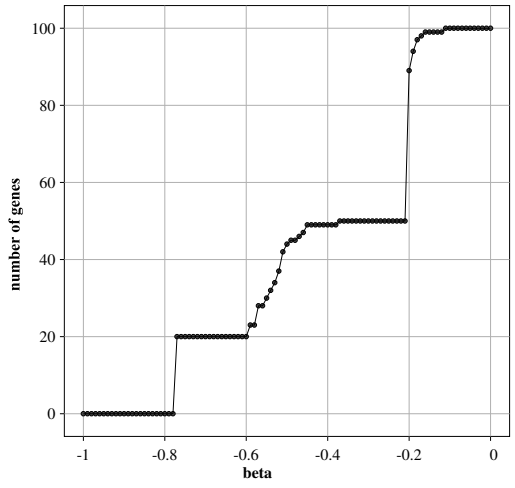


Figure 5: The effect of β on the number of genes for a 500x200 synthetic dataset with one 100x100 bicluster that is composed out of three parts: expression values for 20 genes are sampled from $N(\mu_a, 0.2)$ distributions, 30 genes from $N(\mu_a, 0.5)$ and 50 genes from $N(\mu_a, 1.0)$.

Sensitivity analyses were also performed for the parameters of the normal distribution priors. The Normal-Inverse-χ^2 priors have 4 parameters: μ, σ, κ, ν per array and per bicluster. The priors on the μ's should be uninformative in settings where there is no prior knowledge about the bicluster distributions (centered around 0 with standard deviation $+\inf$), leading to $\mu_{bcl} = 0$ and $\kappa_{bcl} = 0$.

The parameter ν_{bcl} is the relative weight of the prior compared to the data. For example, in certain cases it is desirable to create very strong prior distributions (high ν_{bcl}) with mean and standard deviation around a specific (desired) value. In normal cases however, where little is known about the bicluster, a proper value for ν_{bcl} is around the expected number of genes of the biclusters of interest. That choice gives equal weights to the prior and the data. Finally, the parameter σ_{bcl} is the desired standard deviation of the bicluster distributions. A good choice here is a value that is slightly smaller than the background distribution, e.g. $\sigma_{bcl} = 0.8\sigma_{bgr}$.

4.6 Algorithm running times

The identification of a single bicluster in the synthetic dataset (500 genes, 200 arrays) takes about 1 minute on a dual Opteron 250 server with 2 GB RAM (using only one core). A rough approximation of the running time on similar hardware for varying sizes of the dataset is: $0.00069 \cdot G \cdot A \cdot B$ where G, A and B are the number of genes, arrays and biclusters respectively.

5 Discussion

In this study we designed a PRM based framework for biclustering. The combination of the model design with a hard assignment EM algorithm greatly reduces the computational cost: it allows the decomposition of the posterior into a number of independent factors for each expectation-substep of the EM algorithm which can each be optimized independently. This decomposition turns finding the maximum likelihood into a tractable calculation.

The specific design of the model in combination with the EM algorithm turns *Pro*Bic into a flexible model that can easily be extended with additional data sources (a well designed extension of the model will allow most of the EM steps to remain unchanged).

The *Pro*Bic model uses a hybrid query-driven and model-based approach. This allows researchers to both identify biclusters using a global approach and to incorporate prior knowledge by performing directed queries around genes of interest. The global approach is interesting to study which biclusters and pathways are most prominent in a specific dataset. The advantage of the query based approach on the hand allows biologists to also look for the more subtle biclusters in the dataset that might actually contain their genes of interest and which are often overshadowed by the more prominent signals. Although some other algorithms also allow for query-driven searches such as for instance, the iterative signature algorithm (Bergmann et al., 2003), Gene Expression Mining Server (Wu and Kasif, 2005), Gene Recommender (Owen et al., 2003) and Query based Biclustering(Dhollander et al., 2007), they do not combine the advantages of the query-driven search with a model based approach for identifying overlapping biclusters.

Results on artificial data indicated the convergence behavior of our algorithm. The quality of the solutions (for identifying true overlapping biclusters) is good both in the query-driven and in the global mode. The algorithm is relatively robust against noise and can cope with missing values. In the query driven mode the performance is robust against noisy queries making the algorithm usable for real life situations.

The *Pro*Bic model offers a limited number of biologically interpretable parameters and priors that allow biologists tuning the algorithm towards solving specific biological questions. The results show that default settings are applicable for a wide range of datasets.

6 Conclusion

*Pro*Bic is a PRM based model that identifies overlapping biclusters in a gene expression dataset. It can be used in both a query-based and a global mode. Experiments on synthetic data showed that *Pro*Bic can successfully identify biclusters under various noise levels, with different numbers of missing values and with different levels of overlap between the biclusters. Expert knowledge can be introduced through a number of prior distribution parameters, while the default settings for these parameters are applicable for a wide range of different datasets.

*Pro*Bic is an efficient biclustering algorithm that simultaneously identifies a set of overlapping biclusters in a gene expression dataset. It can be used in both a query-based and a global mode. Experiments on synthetic data showed that *Pro*Bic can successfully identify biclusters under various noise levels, with different numbers of missing values and with potential overlap between the biclusters. Expert knowledge can be introduced through a number of prior distribution parameters, while the default settings for these parameters are applicable for a wide range of different datasets.

7 Author Disclosure Statement

No competing financial interests exist.

References

Agrawal, R., Imielinski, T., and Swami, A.N., 1993, Mining association rules between sets of items in large databases, *in Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, P. Buneman, and S. Jajodia, eds., 207–216, Washington, D.C.

Bergmann, S., Ihmels, J., and Barkai, N., 2003, Iterative signature algorithm for the analysis of large-scale gene expression data., *Physical review. E, Statistical, nonlinear, and soft matter physics* 67.

Cheng, Y., and Church, G.M., 2000, Biclustering of expression data., *Proc Int Conf Intell Syst Mol Biol* 8, 93–103.

D'Ambrosio, B., Jorgensen, J., and Altendorf, E., 2003, Ecosystem analysis using probabilistic relational modeling, *in Proceedings of the IJCAI-2003 Workshop on Learning Statistical Models from Relational Data*, Acapulco, Mexico.

Dempster, A.P., Laird, N.M., and Rubin, D.B., 1977, Maximum likelihood from incomplete data via the em algorithm, *Journal of the Royal Statistical Society* .

Dhollander, T., Sheng, Q., Lemmens, K., De Moor, B., Marchal, K., and Moreau, Y., 2007, Query-driven module discovery in microarray data, *Bioinformatics* 23, 2573–2580.

Friedman, N., Getoor, L., Koller, D., and Pfeffer, A., 1999, Learning probabilistic relational models, 1300–1309.

Gelman, A., Carlin, J.B., Stern, H.S., and Rubin, D.B., 2003, *Bayesian Data Analysis, Second Edition*, Chapman & Hall/CRC.

Getoor, L., Segal, E., Taskar, B., and Koller, D., 2001a, Probabilistic models of text and link structure for hypertext classification.

Getoor, L., Friedman, N., Koller, D., and Taskar, B., 2001b, Learning probabilistic models of relational structure, 170–177, Morgan Kaufmann, San Francisco, CA.

Greenbaum, D., Luscombe, N.M., Jansen, R., Qian, J., and Gerstein, M., 2001, Interrelating different types of genomic data, from proteome to secretome: 'oming in on function., *Genome research* 11, 1463–1468.

Gu, J., and Liu, J., 2008, Bayesian biclustering of gene expression data, *BMC Genomics* 9.

Koller, D., and Pfeffer, A., 1998, Probabilistic frame-based systems, Proc. AAAI, 580–587, Madison.

Lazzeroni, L., and Owen, A., 1999, Plaid models for gene expression data, *Statistica Sinica* 12, 61–86.

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

Newton, J., and Greiner, R., 2004, Hierarchical probabilistic relational models for collaborative filtering, *in Workshop on Statistical Relational Learning*, Banff, Canada.

Owen, A.B., Stuart, J., Mach, K., Villeneuve, A.M., and Kim, S., 2003, A gene recommender algorithm to identify coexpressed genes in c. elegans, *Genome Res.* 13, 1828–1837.

Segal, E., Battle, A., and Koller, D., 2003a, Decomposing gene expression into cellular processes, *Pac. Symp. Biocomput.* 89–100.

Segal, E., Shapira, M., Regev, A., Pe'er, D., Botstein, D., Koller, D., and Friedman, N., 2003b, Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data, *Nat. Genet.* 34, 166–176.

Segal, E., Taskar, B., Gasch, A., Friedman, N., and Koller, D., 2001, Rich probabilistic models for gene expression, *Bioinformatics* 17, S243–S252.

Segal, E., Yelensky, R., and Koller, D., 2003c, Genome-wide discovery of transcriptional modules from dna sequence and gene expression, *Bioinformatics* 19 Suppl 1, I273–I282.

Tanay, A., Sharan, R., Kupiec, M., and Shamir, R., 2004, Revealing modularity and organization in the yeast molecular network by integrated analysis of highly heterogeneous genomewide data, *Proceedings of the National Academy of Sciences U.S.A* 101, 2981–2986.

Van den Bulcke, T., Lemmens, K., Van de Peer, Y., and Marchal, K., 2006, Inferring transcriptional networks by mining 'omics' data, *Current bioinformatics* 1.

Wu, C.J., and Kasif, S., 2005, Gems: a web server for biclustering analysis of expression data., *Nucleic Acids Res* 33.