Identification of overlapping biclusters using Probabilistic Relational Models

Tim Van den Bulcke, Hui Zhao, Kristof Engelen, Bart De Moor, Kathleen Marchal
Overview

• Biclustering and biology
• Probabilistic Relational Models
• ProBic biclustering model
• Algorithm
• Results
• Conclusion
Overview

• Biclustering and biology
  – What is biclustering?
  – Why biclustering?

• Probabilistic Relational Models
• ProBic biclustering model
  • Algorithm
  • Results
  • Conclusion
What is biclustering?

- **Definition in the context of gene expression data:**
  A **bicluster** is a *subset* of genes which show a similar expression profile under a *subset* of conditions.
Biclustering and biology

Why bi-clustering?*

• Only a small set of the genes participates in a cellular process.
• A cellular process is active only in a subset of the conditions.
• A single gene may participate in multiple pathways that may or may not be coactive under all conditions.

Overview

• Biclustering and biology
• **Probabilistic Relational Models**
• *ProBic* biclustering model
• Algorithm
• Results
• Conclusion
Probabilistic Relational Models (PRMs)

Virus strain

Patient

Contact

Treatment

*Image: free interpretation from Segal et al. Rich probabilistic models*

Machine Learning in Systems Biology, Sep 25th
Probabilistic Relational Models (PRMs)

• Traditional approaches “flatten” relational data
  – Causes bias
  – Centered around one view of the data
  – Loose relational structure

• PRM models
  – Extension of Bayesian networks
  – Combine advantages of probabilistic reasoning with relational logic
Overview

• Biclustering and biology
• Probabilistic Relational Models
• ProBic biclustering model
• Algorithm
• Results
• Conclusion
ProBic biclustering model: notation

- **g**: gene
- **c**: condition
- **e**: expression
- **g.B_k**: gene-bicluster assignment for gene $g$ to bicluster $k$
  (unknown, 0 or 1)
- **c.B_k**: condition-bicluster assignment for condition $c$ to bicluster $k$
  (unknown, 0 or 1)
- **e.Level**: expression level value (known, continuous value)
**ProBic biclustering model**

- **Dataset instance**

<table>
<thead>
<tr>
<th>Gene</th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>ID</strong></td>
<td><strong>B₁</strong></td>
</tr>
<tr>
<td>g₁</td>
<td>? (0 or 1)</td>
</tr>
<tr>
<td>g₂</td>
<td>? (0 or 1)</td>
</tr>
</tbody>
</table>

**Expression**

<table>
<thead>
<tr>
<th>g.ID</th>
<th>c.ID</th>
<th>level</th>
</tr>
</thead>
<tbody>
<tr>
<td>g₁</td>
<td>c₁</td>
<td>-2.4</td>
</tr>
<tr>
<td>g₁</td>
<td>c₂</td>
<td>(missing value)</td>
</tr>
<tr>
<td>g₂</td>
<td>c₁</td>
<td>1.6</td>
</tr>
<tr>
<td>g₂</td>
<td>c₂</td>
<td>0.5</td>
</tr>
</tbody>
</table>

Machine Learning in Systems Biology, Sep 25th
**ProBic biclustering model**

- Relational schema and PRM model

Notation:
- \( g \): gene
- \( c \): condition
- \( e \): expression
- \( g.B_k \): gene-bicluster assignment for gene \( g \) to bicluster \( k \) (0 or 1, unknown)
- \( c.B_k \): condition-bicluster assignment for condition \( c \) to bicluster \( k \) (0 or 1, unknown)
- \( e.\text{Level} \): expression level value (continuous, known)
ProBic biclustering model

Database instance

<table>
<thead>
<tr>
<th>Gene</th>
<th></th>
<th>Condition</th>
</tr>
</thead>
<tbody>
<tr>
<td>ID</td>
<td>B1</td>
<td>B2</td>
</tr>
<tr>
<td>g1</td>
<td>0 or 1</td>
<td>0 or 1</td>
</tr>
<tr>
<td>g2</td>
<td>0 or 1</td>
<td>0 or 1</td>
</tr>
</tbody>
</table>

Expression

<table>
<thead>
<tr>
<th>Gene</th>
<th>Condition</th>
<th>level</th>
</tr>
</thead>
<tbody>
<tr>
<td>g1</td>
<td>c1</td>
<td>-5.4</td>
</tr>
<tr>
<td>g1</td>
<td>c2</td>
<td>1.6</td>
</tr>
<tr>
<td>g2</td>
<td>c1</td>
<td>5.5</td>
</tr>
<tr>
<td>g2</td>
<td>c2</td>
<td>-2.4</td>
</tr>
</tbody>
</table>

PRM model

Gene

Condition

ID

Expression

level

P(e.level | g1.B1, g1.B2, c1.ID, g2.B1, g2.B2, c2.ID) = Normal(μ, σ, c1.ID, c2.ID)

ground Bayesian network

Machine Learning in Systems Biology, Sep 25th
**ProBic biclustering model**

- Joint Probability Distribution is defined as a product over each of the node in Bayesian Network:
- **ProBic posterior** (~ likelihood x prior):

\[
\text{posterior} \propto \prod_{c \in \text{set}(c.ID)} \left\{ \prod_{(gb, cb) \in \text{set}(G.B, C.B)} P(\mu_{gb, cb, c}, \sigma_{gb, cb, c}) \right\} \prod_{e \in E: \begin{array}{l} e.gene.B = gb, \\
e.cond.B = cb, \\
e.cond.ID = e \end{array}} P(e.L | g.B_1, g.B_2, c.B_1, c.B_2, c.ID) \right]
\]

Prior condition to bicluster assignments

Expression level prior \((\mu, \sigma)\)'s

Expression level conditional probabilities

Machine Learning in Systems Biology, Sep 25th
Overview

• Biclustering and biology
• Probabilistic Relational Models
• ProBic biclustering model
• Algorithm
• Results
• Conclusion
Algorithm: choices

• Different approaches possible
• Only approximative algorithms are tractable:
  – MCMC methods (e.g. Gibbs sampling)
  – Expectation-Maximization (soft, hard assignment)
  – Variational approaches
  – simulated annealing, genetic algorithms, …
• We chose a hard assignment **Expectation-Maximization algorithm (E.-M.)**
  – Natural decomposition of the model in E.-M. steps
  – Efficient
  – Extensible
  – Relatively good convergence properties for this model
Algorithm: Expectation-Maximization

• **Maximization step:**
  – Maximize posterior w.r.t. $\mu$, $\sigma$ values (model parameters), given the current gene-bicluster and condition-bicluster assignments (=the hidden variables)

• **Expectation step:**
  – Maximize posterior w.r.t. gene-bicluster and condition-bicluster assignments, given the current model parameters
  – Two-step approach:
    • **Step 1:** max. posterior w.r.t. C.B, given G.B and $\mu$, $\sigma$ values
    • **Step 2:** max. posterior w.r.t. G.B, given C.B and $\mu$, $\sigma$ values
Algorithm: Simple Example
Overview

• Biclustering and biology
• Probabilistic Relational Models
• ProBic biclustering model
• Algorithm
• Results
  – Noise sensitivity
  – Bicluster shape
  – Overlap
  – Missing values
• Conclusion
Results: noise sensitivity

• Setup:
  – Simulated dataset: 500 genes x 200 conditions
  – Background distribution: Normal(0,1)
  – Bicluster distributions: Normal( \text{rnd}(N(0,1)), \sigma ), varying sigma
  – Shapes: three 50x50 biclusters
Results: noise sensitivity

Precision (genes) vs. Recall (genes)

Precision (conditions) vs. Recall (conditions)

Precision = TP / (TP+FP)       Recall = TP / (TP+FN)

Machine Learning in Systems Biology, Sep 25th
Results: bicluster shape independence

• Setup:
  – Dataset: 500 genes x 200 conditions
  – Background distribution: N(0,1)
  – Bicluster distributions: N( rnd(N(0,1)), 0.2 )
  – Shapes: 80x10, 10x80, 20x20
Results: bicluster shape independence
Results: Overlap examples

- Two biclusters (50 genes, 50 conditions)
- Overlap: 25 genes, 25 conditions
- Two biclusters (10 genes, 80 conditions)
- Overlap: 2 genes, 40 conditions
Results: Missing values

• *ProBic* model has no concept of ‘missing values’

\[
posterior \propto \prod_{c \in \text{set}(c,ID)} \left\{ \prod_{(gb,cb) \in \text{set}(G,B,C,B)} P(\mu_{gb,cb,c}, \sigma_{gb,cb,c}) \prod_{e \in E:\ e.gene=B=gb, e.cond.B=cb, e.cond.ID=c} P(e,L|g.B_1, g.B_2, c.B_1, c.B_2, c.ID)) \right\} \cdot \prod_{k} \prod_{c} P(c.B_k) \cdot \prod_{k} \prod_{g} P(g.B_k)
\]

→ No prior missing value estimations which could bias the result
Results: Missing values – one example

- 500 genes x 200 conditions
- Noise std: bicluster 0.2, background 1.0
- Missing value: 70%
- One bicluster 50x50
Results: Missing values

Precision (genes)  Recall (genes)

Precision (conditions)  Recall (conditions)

% missing values

Machine Learning in Systems Biology, Sep 25th
Overview

• Biclustering and biology
• Probabilistic Relational Models
• ProBic biclustering model
• Algorithm
• Results
• Conclusion
Conclusion

- Noise robustness
- Naturally deals with missing values
- Relatively independent of bicluster shape
- Simultaneous identification of multiple overlapping biclusters
- Can be used query-driven
- Extensible
Acknowledgements

KULeuven:

• whole Biol group, ESAT-SCD
  – Tim Van den Bulcke
  – Thomas Dhollander

• whole CMPG group
  (Centre of Microbial and Plant Genetics)
  – Kristof Engelen
  – Kathleen Marchal

UGent:

• whole Bioinformatics & Evolutionary Genomics group
  – Tom Michoel
Thank you for your attention!
Near future

- Automated definition of algorithm parameter settings
- Application biological datasets
  - Dataset normalization
- Extend model with different overlap models
- Model extension from biclusters to regulatory modules
  include motif + ChIP-chip data
Results: Missing values

- **ProBic model has no concept of ‘missing values’**
Algorithm: example
Algorithm properties

• **Speed:**
  - 500 genes, 200 conditions, 2 biclusters: 2 min.
  - **Scaling:**
    • \( \sim \text{#genes} \cdot \text{#conditions} \cdot 2^{\#\text{biclusters}} \) (worse case)
    • \( \sim \text{#genes} \cdot \text{#conditions} \cdot (\#\text{biclusters})^p \) (in practice), \( p=1..3 \)
Overview

- Biclustering and biology
- Probabilistic Relational Models
- *ProBic* biclustering model
- Algorithm
- Results
- Discussion
- Conclusion
Discussion: Expectation-Maximization

• Initialization:
  – initialization with (almost) all genes and condition: convergence to good local optimum
  – multiple random initializations: many initializations required (speed!)

• E.-M. steps:
  – limit changes in gene/condition-bicluster assignments in both E-steps results in higher stability (at cost of slower convergence)
Probabilistic Relational Models (PRMs)

- **Relational extension to Bayesian Networks (BNs):**
  - BNs ~ a single flat table
  - PRMs ~ relational data structure

- A relational scheme (implicitly) defines a constrained Bayesian network

- In PRMs, probability distributions are shared among all objects of the same class

- **Likelihood function:** (very similar to chain-rule in Bayesian networks)
  \[
  P(I \mid \sigma, S, \Theta) = \prod_{x \in \sigma} \prod_{x.A} P(x.A \mid \text{parents}_{s,\sigma}(x.A))
  \]

- Learning PRM model e.g. using maximum likelihood principle
Algorithm: user-defined parameters

- \( P(\mu_{g,B,c}, \sigma_{g,B,c}) \): prior distributions for \( \mu, \sigma \)
  - Conjugate:
    - Normal-Inverse-\( \chi^2 \) distribution or
    - Normal distribution with pseudocount
  - Makes extreme distributions less likely

- \( P(a.B_k) \): prior probability that a condition is in bicluster \( k \)
  - Prevents background conditions to be in biclusters
  - If no prior distribution \( P(\mu, \sigma) \): conditions are always more likely to be in a bicluster due to statistical variations.

- \( P(g.B_k) \): prior probability that a gene is in bicluster \( k \)
  - Initialize biclusters with seed genes: query-driven biclustering

- \( P(a.B_k) \) and \( P(g.B_k) \):
  - Both have impact on the preference for certain bicluster shapes
Algorithm: Expectation-Maximization

• **Expectation step 2:**
  - \( \text{argmax}_{G,B} \log(\text{posterior}) \)

\[
\text{posterior} \propto \prod_{c \in \text{set}(c,ID)} \left\{ \prod_{(gb,cb) \in \text{set}(G.B,C.B)} P(\mu_{gb,cb,c}, \sigma_{gb,cb,c}) \right\} \prod_{e \in E: \text{e.gene}.B=gb, \text{e.cond}.B=cb, \text{e.cond}.ID=c} P(e.L|g.B_1,g.B_2,c.B_1,c.B_2,c.ID)]
\]

\[
\cdot \prod_{k \in c} \prod_{g} P(c.B_k) \cdot \prod_{k} \prod_{g} P(g.B_k)
\]

constant

without prior: independent per gene

\( \rightarrow \)

Approximation based on previous iteration

*quasi independence* if small changes in assignment
ProBic biclustering model

PRM model

\[
P(e.\text{level} | g.B_1, g.B_2, c.B_1, c.B_2, c.ID) = \text{Normal}(\mu_{g.B,c.B,c.ID}, \sigma_{g.B,c.B,c.ID})
\]

Machine Learning in Systems Biology, Sep 25th
**ProBic biclustering model**

- ground Bayesian network
ProBic biclustering model

- **Likelihood function:**
  (\(\sim\) chain-rule in Bayesian networks)

\[
P(I \mid \sigma, S, \Theta) = \prod_{x \in \sigma} \prod_{x.A} P(x.A \mid \text{parents}_{S,\sigma}(x.A))
\]

- **In PRMs, probability distributions are shared among all objects of the same class**
Algorithm: Expectation-Maximization

- Maximization step:
  \[ \text{argmax}_{\mu, \sigma} \log(\text{posterior}) \]

\[
\text{posterior} \propto \prod_{c \in \text{set}(c.ID)} \left\{ \prod_{(gb, cb) \in \text{set}(G.B, C.B)} \right\} \left( \prod_{e \in F: e.\text{gene}=gb, e.\text{cond}=cb, e.\text{cond}.ID=c} P(e.L | g.B_1, g.B_2, c.B_1, c.B_2, c.ID) \right) \\
\cdot \prod_{i \in c} P(c.B_i) \cdot \prod_{k \in g} P(g.B_k)
\]

constant

independent per condition

+ analytic solution based on sufficient statistics
Algorithm: Expectation-Maximization

- **Expectation step 1: assign conditions**
  - \( \text{argmax}_{c,B} \log(\text{posterior}) \)

\[
\text{posterior} \propto \prod_{c \in \text{set}(c,ID)} \left\{ \prod_{(gb,cb) \in \text{set}(G,B,C,B)} P(\mu_{gb,cb,c}, \sigma_{gb,cb,c}) \right\} \prod_{e \in E: e.gene.B=gb, e.cond.B=cb, e.cond.ID=c} P(e.L|g.B_1, g.B_2, c.B_1, c.B_2, c.ID) \]

\[
\times \prod_{k} \prod_{c} P(c.B_k) \prod_{g} P(g.B_k)
\]

constant

independent per condition
**Algorithm: Expectation-Maximization**

- **Expectation step 1:**
  - Evaluate function for every condition and for every bicluster assignment
    e.g. 200 conditions, 30 biclusters: $200 \times 2^{30} = 200$ billion ~ a lot
  - But can be performed very efficiently:
    - Partial solutions can be reused among different bicluster assignments
Algorithm: Expectation-Maximization

- **Expectation step 1:**
  - Evaluate function for every condition and for every bicluster assignment
    e.g. 200 conditions, 30 biclusters: $200 \times 2^{30} = 200$ billion ~ a lot
  - But can be performed very efficiently:
    - Partial solutions can be reused among different bicluster assignments
    - Only evaluate potential good solutions: use *Apriori*-like approach.
Algorithm: Expectation-Maximization

- **Expectation step 1:**
  - Evaluate function for every condition and for every bicluster assignment
    e.g. 200 conditions, 30 biclusters: $200 \times 2^{30} = 200$ billion ~ a lot
  - But can be performed very efficiently:
    - Partial solutions can be reused among different bicluster assignments
    - Only evaluate potential good solutions: use *Apriori*-like approach.
    - Avoid background evaluations
Algorithm: Expectation-Maximization

• **Expectation step 2:**
  – Analogous approach as in step 1
Acknowledgements

**KULeuven:**
- whole Biol group, ESAT-SCD
  - Tim Van den Bulcke
  - Thomas Dhillander
- whole CMPG group (Centre of Microbial and Plant Genetics)
  - Kristof Engelen
  - Kathleen Marchal

**UGent:**
- whole Bioinformatics & Evolutionary Genomics group
  - Tom Michoel
Algorithm: iteration strategy

• Many implementation choices in generalized E.-M.:
  – Single E-step per M-step
  – Iterate E-steps until convergence per M-step
  – Iteratively perform E-step 1, M-step, E-step 2, M-step
  – …

• Our choice:
  – Iteratively perform E-step 1, M-step, E-step 2, M-step
  – fast and good convergence properties
Results: 9 bicluster dataset (15 genes x 80 conditions)
Results: Missing values

- 500 genes, 300 conditions
- Noise stdev: 0.2 (bicluster), 1.0 background
- 1 50x50 bicluster