Systems biology
Learning signalling pathways and regulatory networks from postgenomic data

possibly completely unknown
E.g.: Flow cytometry experiments
Machine Learning statistical methods

true network
extracted network
Is the extracted network a good prediction of the real relationships?
Reverse Engineering of Regulatory Networks

- Can we learn network structures from postgenomic data themselves?
- Are there statistical methods to distinguish between direct and indirect correlations?
- Is it worth applying time-consuming Bayesian network approaches although computationally cheaper methods are available?

Three widely applied methodologies:

- Relevance networks (RNs)
- Graphical Gaussian models (GGMs)
- Bayesian networks (BNs)

Relevance networks (RNs)

1. Choose a measure of association $A(.,.)$
2. Define a threshold value $t_A$
3. For all pairs of domain variables $(X,Y)$ compute their association $A(X,Y)$
4. Connect those variables $(X,Y)$ by an undirected edge whose association $A(X,Y)$ exceeds the predefined threshold value $t_A$
Measures of association (RNs)

E.g.: Pairwise Mutual Information (MI)
(for discrete data)

$$MI(x, y) = \sum_{i=1}^{r} \sum_{j=1}^{r} P(x = i, y = j) \log \frac{P(x = i, y = j)}{P(x = i)P(y = j)}$$

Problems of pairwise associations without taking the context of the system into consideration

E.g.: Correlation between A and C is disturbed (weakened) by the influence of B
May give misleading network predictions!

Correlation matrix i.e. Pearson correlation coefficients (as matrix)

$$\Sigma = \begin{pmatrix}
\sigma_{11} & \sigma_{12} & \sigma_{13} & \ldots & \sigma_{1n} \\
\sigma_{21} & \sigma_{22} & \sigma_{23} & \ldots & \sigma_{2n} \\
\sigma_{31} & \sigma_{32} & \sigma_{33} & \ldots & \sigma_{3n} \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
\sigma_{n1} & \sigma_{n2} & \sigma_{n3} & \ldots & \sigma_{nn}
\end{pmatrix}$$

Strong correlation $$\sigma_{12}$$

‘direct interaction’

‘common regulator’

‘indirect interaction’
Relevance networks (RNs)

Obviously relevance networks cannot distinguish between direct and indirect interactions. Already from a theoretical point of view it can be seen that lots of false positive edges will be extracted. And this is a fundamental disadvantage of Relevance network approaches.

Graphical Gaussian Models (GGMs)

Partial correlation, i.e. correlation conditional on all other domain variables

\[ \text{Corr}(X_1, X_2 | X_3, \ldots, X_n) \]

1. Linear regression models (without intercept):

\[ X_1 = \beta_3^1 \cdot X_3 + \ldots + \beta_n^1 \cdot X_n + \varepsilon^1 \]
\[ X_2 = \beta_3^2 \cdot X_3 + \ldots + \beta_n^2 \cdot X_n + \varepsilon^2 \]

Compute the least squares (LS) estimators
Graphical Gaussian Models (GGMs)

Partial correlation, i.e. correlation conditional on all other domain variables

Corr\( (X_1, X_2 | X_3, \ldots, X_n) \)

2. Compute the residuals:

\[
R_1 = X_1 - \hat{\beta}_3 \cdot X_3 - \cdots - \hat{\beta}_n \cdot X_n \\
R_2 = X_2 - \hat{\beta}_3 \cdot X_3 - \cdots - \hat{\beta}_n \cdot X_n
\]

3. The Pearson correlation \( \text{corr}(R_1, R_2) \) is the partial correlation of \( X_1 \) and \( X_2 \).

Graphical Gaussian Models (GGMs)

Alternative computation of partial correlation coefficients:

1. Compute/estimate the n-by-n covariance matrix of \((X_1, \ldots, X_n)^T\) from a data matrix \(D\)

\[
D = \begin{pmatrix}
D_{1,1} & \cdots & D_{1,m} \\
\vdots & \ddots & \vdots \\
D_{n,1} & \cdots & D_{n,m}
\end{pmatrix}
\]

\(n\) variables \(X_1, \ldots, X_n\)

\(m\) observations \(D_{1,1}, \ldots, D_{m}\)

Each column is a realisation of \((X_1, \ldots, X_n)^T\)
Graphical Gaussian Models (GGMs)

Alternative computation of partial correlation coefficients:

2. Invert the covariance matrix to obtain the precision matrix:

\[
\Sigma = \begin{pmatrix}
\hat{\Sigma}_{1,1} & \cdots & \hat{\Sigma}_{1,n} \\
\vdots & \ddots & \vdots \\
\hat{\Sigma}_{n,1} & \cdots & \hat{\Sigma}_{n,n}
\end{pmatrix}
\]

\[
\hat{\Pi} = \Sigma^{-1} = \begin{pmatrix}
\hat{\Pi}_{1,1} & \cdots & \hat{\Pi}_{1,n} \\
\vdots & \ddots & \vdots \\
\hat{\Pi}_{n,1} & \cdots & \hat{\Pi}_{n,n}
\end{pmatrix}
\]

covariance matrix precision matrix (matrix-inverse of the covariance matrix)

Partial correlation between \(X_i\) and \(X_j\)

\[
\text{Corr}(X_i, X_j | \{X_1, \ldots, X_n\} \backslash \{X_i, X_j\})
\]

All other variables without \(X_i\) and \(X_j\)

Graphical Gaussian Models (GGMs)

But there is usually one substantial problem:
There are often fewer observations than variables

Symbolically: \(m < n\)

\[
\hat{\Sigma} = \begin{pmatrix}
\hat{\Sigma}_{1,1} & \cdots & \hat{\Sigma}_{1,n} \\
\vdots & \ddots & \vdots \\
\hat{\Sigma}_{n,1} & \cdots & \hat{\Sigma}_{n,n}
\end{pmatrix}
\]

so that the covariance matrix is singular and cannot be inverted
Idea: Shrinkage estimation of the covariance matrix

\[ \hat{\Sigma}_S (\lambda_0 \mid T) = (1 - \lambda_0) \cdot \hat{\Sigma}_{ML} + \lambda_0 \cdot \hat{\Sigma}_{MLIT} \]

0<λ₀<1 estimated (optimal) shrinkage intensity

\[ \hat{\Sigma}_{MLIT} = \begin{pmatrix} \hat{\Sigma}_{11} & \cdots & 0 & 0 \\ 0 & \hat{\Sigma}_{22} & \cdots & 0 \\ \vdots & \ddots & \ddots & \vdots \\ 0 & \cdots & 0 & \hat{\Sigma}_{nn} \end{pmatrix} \]

with

\[ MSE(\hat{\Sigma}_S (\lambda_0 \mid T)) \leq MSE(\hat{\Sigma}_S (\lambda \mid T)) \]

where

\[ MSE(\hat{\Sigma}_S (\lambda \mid T)) = E[\|\hat{\Sigma}_S (\lambda \mid T) - \Sigma \|^2_F] \]

\[ \hat{\Sigma}_S (\lambda_0 \mid T) > 0 \quad \text{is guaranteed} \]

How to compute/estimate the shrinkage intensity λ₀?

\[ D = \begin{pmatrix} D_{1,1} & \cdots & D_{1,m} \\ \vdots & \ddots & \vdots \\ D_{n,1} & \cdots & D_{n,m} \end{pmatrix} \quad \text{data matrix} \]

Shrinkage intensity estimation

\[ \hat{\Sigma}_S (\lambda_0 \mid T) = (1 - \lambda_0) \cdot \hat{\Sigma}_{ML} + \lambda_0 \cdot \hat{\Sigma}_{MLIT} \]

0<λ₀<1 estimated (optimal) shrinkage intensity

\[ \hat{\Sigma}_{MLIT} = \begin{pmatrix} \hat{\Sigma}_{11} & \cdots & \hat{\Sigma}_{1,n} \\ \hat{\Sigma}_{2,1} & \cdots & \hat{\Sigma}_{2,n} \\ \vdots & \ddots & \vdots \\ \hat{\Sigma}_{n,1} & \cdots & \hat{\Sigma}_{n,n} \end{pmatrix} + \lambda_0 \cdot \begin{pmatrix} 0 & \cdots & 0 \\ 0 & \cdots & \cdots & \vdots \\ 0 & \cdots & 0 \end{pmatrix} \]

\[ \bar{D}_i = \frac{1}{m} \sum_{k=1}^{m} D_{i,k} \quad \text{empirical mean of variable } X_i \]

\[ s_{i,j}^2 = \frac{1}{m-1} \sum_{k=1}^{m} (D_{i,k} - \bar{D}_i) \cdot (D_{j,k} - \bar{D}_j) \]

= ML estimator of \( \hat{\Sigma}_{i,j} \)
Shrinkage intensity estimation

\[
\hat{\Sigma}_S(\lambda_0 | T) = (1 - \lambda_0) \cdot \begin{pmatrix}
\hat{\Sigma}_{01} & \hat{\Sigma}_{02} & \ldots & \hat{\Sigma}_{0n} \\
\hat{\Sigma}_{11} & \hat{\Sigma}_{12} & \ldots & \hat{\Sigma}_{1n} \\
\vdots & \vdots & \ddots & \vdots \\
\hat{\Sigma}_{n1} & \hat{\Sigma}_{n2} & \ldots & \hat{\Sigma}_{nn}
\end{pmatrix} + \lambda_0 \cdot \begin{pmatrix}
\hat{\Sigma}_{01} & \hat{\Sigma}_{02} & \ldots & \hat{\Sigma}_{0n} \\
0 & \hat{\Sigma}_{22} & \ldots & \vdots \\
\vdots & \vdots & \ddots & \vdots \\
0 & \ldots & \ldots & \hat{\Sigma}_{nn}
\end{pmatrix}
\]

\[
r_{i,j}^2 = \frac{s_{i,j}^2}{\sqrt{s_{i,i}^2 \cdot s_{j,j}^2}} \quad \text{Correlation of } X_i \text{ and } X_j
\]

Shrinkage covariance estimator:

\[
(\hat{\Sigma}_S(\lambda_0 | T))_{i,j} = \begin{cases}
s_{i,j}^2, & i = j \\
n_{i,j} \cdot \min\{1, (\max\{0, 1 - \hat{\lambda}_0\})\}, & i \neq j
\end{cases}
\]

Idea: Shrinkage estimation of the covariance matrix

\[
\hat{\Sigma}_S(\lambda_0 | T) = (1 - \lambda_0) \cdot \hat{\Sigma}_{ML} + \lambda_0 \cdot \hat{\Sigma}_{MLIT}
\]

\[
\hat{\Sigma}_S(\lambda_0 | T) = \begin{pmatrix}
\hat{\Sigma}_{11} & \hat{\Sigma}_{12} & \ldots & \hat{\Sigma}_{1n} \\
\hat{\Sigma}_{21} & \hat{\Sigma}_{22} & \ldots & \hat{\Sigma}_{2n} \\
\vdots & \vdots & \ddots & \vdots \\
\hat{\Sigma}_{n1} & \hat{\Sigma}_{n2} & \ldots & \hat{\Sigma}_{nn}
\end{pmatrix}
\]

\[
\hat{\lambda}_0 = \frac{\sum_{i=1}^{n-2} \sum_{j=i+1}^{n} \text{Var}_{ML}(r_{i,j}^2)}{\left(\sum_{i=1}^{n-2} \sum_{j=i+1}^{n} (r_{i,j}^*)^2\right)^2}
\]

\[
\text{Var}_{ML}(r_{i,j}^2) = \frac{m}{(m-1)^3} \sum_{k=1}^{m} (w_{k,i,j} - \overline{w}_{i,j})^2
\]

\[
w_{k,i,j} = \left(\sqrt{s_{i,i}^2 \cdot s_{j,j}^2}\right)^{-1} \cdot (D_{k,i} - \overline{D}_i) \cdot (D_{k,j} - \overline{D}_j)
\]

\[
\overline{w}_{i,j} = \frac{1}{m} \sum_{k=1}^{m} w_{k,i,j}
\]

Summary: The shrinkage covariance estimator can be computed in closed-form. And it is guaranteed that the estimator is positive definite so that the matrix inverse (precision matrix) can be computed.
Remark: Other target matrices
\[
\hat{\Sigma}_{MLT} = \begin{pmatrix}
\hat{\Sigma}_{11} & \cdots & 0 & 0 \\
0 & \hat{\Sigma}_{22} & \cdots & 0 \\
\vdots & \ddots & \ddots & \vdots \\
0 & \cdots & 0 & \hat{\Sigma}_{nn}
\end{pmatrix}
\]
Here, we considered this "target matrix", and we computed the shrinkage estimator for this target matrix only.

Other target matrices can be used. E.g.:
\[
\hat{\Sigma}_{MLT} = \begin{pmatrix}
\hat{\Sigma} & \cdots & 0 & 0 \\
0 & \hat{\Sigma} & \cdots & 0 \\
\vdots & \ddots & \ddots & \vdots \\
0 & \cdots & 0 & \hat{\Sigma}
\end{pmatrix}
\]

The shrinkage estimator depends on the target matrix!

Further drawbacks

- Relevance networks and Graphical Gaussian models can extract undirected edges only.
- Bayesian networks promise to extract at least some directed edges.

But can we trust in these edge directions?!
It may be better to learn undirected edges than learning directed edges with false orientations?!

Bayesian networks (see lectures V2-V9)

- Relevance networks (RNs)
- Graphical Gaussian models (GGMs)
- Bayesian networks (BNs)

- Marriage between graph theory and probability theory.
- Directed acyclic graph (DAG) represents conditional independence relations.
- Markov assumption leads to a factorization of the joint probability distribution:

\[
P(A, B, C, D, E, F) = P(A) \cdot P(B \mid A) \cdot P(C \mid A) \cdot P(D \mid B, C) \cdot P(E \mid D) \cdot P(F \mid C, D)
\]
Edge relation features

We can convert the DAG sample into a CPDAG sample $G_{J+1}, \ldots, G_T$ by employing the DAG-to-CPDAG algorithm. We obtain a sample of CPDAGs: $G_j \rightarrow \text{CPDAG}(G_j)$ ($j=J+1, \ldots, T$)

We can interpret undirected edges in CPDAGs as superposition of two directed edges pointing in opposite direction.

Evaluation of Performance

- Relevance networks and Graphical Gaussian models extract undirected edges (edge scores = (partial) correlations)
- Bayesian networks extract undirected as well as directed edges (edge scores = posterior probabilities of edges)
- Undirected edges can be interpreted as superposition of two directed edges with opposite direction.
- How to cross-compare the learning performances when the true regulatory network is known?
- Distinguish between DGE (directed graph evaluation) and UGE (undirected graph evaluation)

Probabilistic inference - UGE

Utilise the DAG (CPDAG) sample $G_{J+1}, \ldots, G_T$ for estimating the posterior probabilities of edge relation features:

$$\hat{P}(A \rightarrow B) = \frac{1}{(T-J)} \sum_{i=J+1}^{T} I(G_i)$$

where $I(G_i)$ is 1 if the CPDAG of $G_i$ contains the directed edge $A \rightarrow B$, and 0 otherwise.
Probabilistic inference - UGE

Probabilistic inference

Probabilistic inference

Probabilistic inference

Probabilistic inference

How can we get a network prediction? For each edge we have an undirected edge score.

We have to impose a threshold on the edge scores to obtain a concrete network prediction! Only edges having scores higher than the threshold will be extracted.

There are three possible edge connections: 2 edges and 1 non-edge.
Probabilistic inference

The predicted network depends on the threshold. In the example below a high threshold gives 1 true positive (TP) edge finding and 0 false positive (FP) edge findings. A low threshold gives 2 true positive (TP) edge findings but 1 false positive (FP) edge finding.

<table>
<thead>
<tr>
<th>high</th>
<th>Thresholding</th>
<th>low</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>concrete network (skeleton) predictions</td>
<td></td>
</tr>
<tr>
<td>TP:1</td>
<td>FP:0</td>
<td>TP:2</td>
</tr>
<tr>
<td>FP:0</td>
<td></td>
<td>FP:1</td>
</tr>
</tbody>
</table>

**TN = true negatives = 1**

**TP = true positives = 1**

**FN = false negatives = 1**

**FP = false positives = 0**

**TP+FN = number of network edges = 2**

**TN+FP = number of network non-edges = 1**

**Probabilistic inference**

**TN = true negatives = 0**

**TP = true positives = 2**

**FN = false negatives = 0**

**FP = false positives = 1**

**TP+FN = number of network edges = 2**

**TN+FP = number of network non-edges = 1**
Probabilistic inference

Sensitivity = $\frac{TP}{TP + FN}$
Specificity = $\frac{TN}{TN + FP}$

$1 - \text{Specificity} = \frac{FP}{TN + FP}$

Which prediction is better? What is more important? The sensitivity or the specificity?

Probabilistic inference - DGE

Sensitivity = $\frac{TP}{TP + FN}$
Specificity = $\frac{TN}{TN + FP}$

Instead of imposing one single threshold one may compute the sensitivity and specificity for different thresholds. Afterwards the resulting sensitivities (y-axis) can be plotted against the inverse specificities (x-axis). This gives the Receiver-Operator-Characteristic (ROC) curve.
Evaluation 1: AUC scores

Area under Receiver Operator Characteristic (ROC) curve

AUC = 0.5
AUC = 1
0.5 ≤ AUC ≤ 1

Evaluation 2: TP scores

We set the threshold such that we obtain 5 spurious edges (5 FPs) and count the corresponding number of true edges (TP count).

“How many true positive edges can we get when accepting 5 FP edge findings?”

Evaluation

Two ways of interpreting edges

- **UGE**: undirected graph evaluation
- **DGE**: directed graph evaluation

Two evaluation procedures:

- **AUC**: Area under the ROC curve, with larger areas indicating overall, a better performance.
- **TP count**: True positive number of edges for the same false positive count of FP = 5 across all methods.
Thank you for your attention!

Any questions?