Pathway & Network Analysis of Omics Data: Networks in Biology

Ali Shojaie
Department of Biostatistics
University of Washington
faculty.washington.edu/ashojaie

Summer Institute for Statistical Genetics – 2018

Why Study Networks?

- Components of biological systems (genes, proteins etc) interact with each other to carry out cell functions.
- Examples of such interactions include signaling, regulation and interactions between proteins.
- We cannot understand the function and behavior of biological systems by studying individual components (2 + 2 ≠ 4!).
- Networks provide an efficient representation of complex interactions in cells, and a basis for mathematical/statistical models to study these systems.
Central Dogma of Molecular Biology (Extended)

DNA → Genomics – 25,000 Genes

RNA → Transcriptomics – 100,000 Transcripts

Protein → Proteomics – 1,000,000 Proteins

Biochemicals (Metabolites) → Metabolomics ~2000 Compounds

Central Dogma of Molecular Biology (Extended)

Networks in Biology: Gene Regulatory Interactions

DNA → Target gene

RNA → mRNA

Protein → Cell functions

Biochemicals (Metabolites) → Metabolism

Networks in Biology: Gene Regulatory Interactions

A GENE REGULATORY NETWORK

INPUT

DNA

INPUT

mRNA

OUTPUT

Cell functions

OUTPUT

Protein

INPUT

Signal A

INPUT

Signal B

OUTPUT

mRNA

OUTPUT

Protein

© Ali Shojaie  
SISG: Pathway & Networks

3

© Ali Shojaie  
SISG: Pathway & Networks

4
Networks in Biology: Gene Regulatory Networks

Networks in Biology: Protein-Protein Interaction
Networks in Biology: Protein-Protein Interactions (PPI)

Networks in Biology: Metabolic Reactions

- DMA:PP
- IPP
- GPP
- Squuirone
- Lanosterol
But Do Networks Matter?

- They Do!
- Recent studies have linked changes in gene/protein networks with many human diseases.

**Gene Networks and microRNAs Implicated in Aggressive Prostate Cancer**


1Departments of Laboratory Medicine and Pathology and 2Health Sciences Research, Mayo Clinic, Rochester, Minnesota; and Departments of 3Laboratory Medicine and Pathology, 4Medicine, and Genetics, Cell Biology, and Development, University of Minnesota, Minneapolis, Minnesota
But Do Networks Matter?

Estrogen-Regulated Gene Networks in Human Breast Cancer Cells: Involvement of E2F1 in the Regulation of Cell Proliferation

Joshua D. Stender, Jonna Frasor, Barry Komm, Ken C. N. Chang, W. Lee Kraus, and Benita S. Katzenellenbogen

Departments of Biochemistry (J.D.S.) and Molecular and Integrative Physiology (J.F., B.S.K.), University of Illinois at Urbana-Champaign, Urbana, Illinois 61801-3704; Women's Health and Musculoskeletal Biology (B.K., K.C.N.C.), Wyeth Research, Collegeville, Pennsylvania 19426; and Department of Molecular Biology and Genetics (W.L.K.), Cornell University, Ithaca, New York 14853-4203

*Correspondence: Benita.S.Katzenellenbogen@uiuc.edu

Significance

A Transcriptional Signature and Common Gene Networks Link Cancer with Lipid Metabolism and Diverse Human Diseases

Heather A. Hirsch,1,3,4 Dimitrios Ililopoulos,1,3 Amita Joshi,1,3,4 Yong Zhang,2 Savina A. Jaeger,2 Martha Bulyk,3,4,5 Philip N. Tsichlis,6 X. Shirley Liu,7 and Kevin Struhl1,4

1Department of Biological Chemistry and Molecular Pharmacology, Harvard Medical School, Boston, MA 02115, USA
2Department of Biostatistics and Computational Biology, Dana Farber Cancer Institute, Harvard School of Public Health, Boston, MA 02115, USA
3Division of Genetics, Department of Medicine, Brigham and Women’s Hospital and Harvard Medical School, Boston, MA 02115, USA
4Department of Pathology, Brigham and Women’s Hospital and Harvard Medical School, Boston, MA 02115, USA
5Harvard/MIT Division of Health Sciences and Technology (HST), Harvard Medical School, Boston, MA 02115, USA
6Molecular Oncology Research Institute, Tufts Medical Center, Boston, MA 02111, USA
7These authors contributed equally to this work

*Correspondence: kevin@hms.harvard.edu
DOI 10.1016/j.ccr.2010.01.022
But Do Networks Matter?

And, incorporating the knowledge of networks improves our ability to find causes of complex diseases.
Networks: A Short Premier

- A network is a collection of nodes $V$ and edges $E$.
- We assume there are $p$ nodes in the network, and that the nodes correspond to random variables $X_1, \ldots, X_p$. 
Networks: A Short Premier

- A network is a collection of nodes $V$ and edges $E$.
- We assume there are $p$ nodes in the network, and that the nodes correspond to random variables $X_1, \ldots, X_p$.
- Edges can be directed $X \rightarrow Y$ or undirected $X - Y$.

In all these example, the node set is $V = \{1, 2, 3\}$. 
Networks: A Short Premier

- A network is a collection of nodes $V$ and edges $E$.
- We assume there are $p$ nodes in the network, and that the nodes correspond to random variables $X_1, \ldots, X_p$.
- Edges can be directed $X \to Y$ or undirected $X - Y$.

In all these example, the node set is $V = \{1, 2, 3\}$.

The edges are:

- $E_1 = \{1 \to 2, 2 \to 3\}$
- $E_2 = \{1 \to 3, 3 \to 2\}$
- $E_3 = \{1 \to 2, 1 \to 3\}$
Networks: A Short Premier

- A convenient way to represent the edges of the network is to use an adjacency matrix $A$.
Networks: A Short Premier

- A convenient way to represent the edges of the network is to use an adjacency matrix $A$
- A matrix is a rectangular array of data (similar to a table)
- Values in each entry are shown by indices of row and column

$$A = \begin{bmatrix} . & x & . \\ . & . & . \\ . & . & . \end{bmatrix}$$

Here, $x$ is in row 1 and column 2

- Adjacency matrix is a square matrix, which has a 1 if there is an edge from a node in one row to a node in another column, and 0 otherwise
Networks: A Short Premier

- A convenient way to represent the edges of the network is to use an adjacency matrix \( A \)
- A matrix is a rectangular array of data (similar to a table)
- Values in each entry are shown by indeces of row and column

\[
A = \begin{bmatrix}
\ldots & x & \ldots \\
\ldots & \ldots & \ldots \\
\ldots & \ldots & \ldots 
\end{bmatrix}
\]
Here, \( x \) is in row 1 and column 2

- Adjacency matrix is a square matrix, which has a \textbf{1 if there is an edge} from a node in one row to a node in another column, and \textbf{0 otherwise}
- For undirected edges, we add a \textbf{1} in both directions
Networks: A Short Premier

What Do Edges in Biological Networks Mean?

- In gene regulatory networks, an edge from gene $i$ to gene $j$ often means that $i$ affects the expression of $j$; i.e., as $i$’s expression changes, we expect that expression of $j$ to increase/decrease.
What Do Edges in Biological Networks Mean?

- In gene regulatory networks, an edge from gene $i$ to gene $j$ often means that $i$ affects the expression of $j$; i.e. as $i$’s expression changes, we expect that expression of $j$ to increase/decrease.

- In protein-protein interaction networks, an edge between proteins $i$ and $j$ often means that the two proteins bind together and form a protein complex. Therefore, we expect that these proteins are generated at similar rates.

- In metabolic networks, an edge between compound $i$ and $j$ often means that the two compounds are involved in the same reaction, meaning that they are generated at relative rates.
What Do Edges in Biological Networks Mean?

- In **gene regulatory networks**, an edge from gene $i$ to gene $j$ often means that $i$ affects the expression of $j$; i.e. as $i$’s expression changes, we expect that expression of $j$ to increase/decrease.

- In **protein-protein interaction networks**, an edge between proteins $i$ and $j$ often means that the two proteins bind together and form a protein complex. Therefore, we expect that these proteins are generated at similar rates.

- In **metabolic networks**, an edge between compound $i$ and $j$ often means that the two compounds are involved in the same reaction, meaning that they are generated at relative rates.

- Thus, edges represent some type of **association among genes, proteins or metabolites**, defined generally to include **linear or nonlinear** associations; more later....
Statistical Models for Biological Networks

- We use the framework of graphical models
- In this setting, nodes correspond to “random variables”
Statistical Models for Biological Networks

- We use the framework of graphical models
- In this setting, nodes correspond to “random variables”
- In other words, each node of the network represents one of the variables in the study
  - In gene regulatory networks, nodes \( \equiv \) genes
  - In PPI networks, nodes \( \equiv \) proteins
  - In metabolic networks, nodes \( \equiv \) metabolites

- In practice, we observe \( n \) measurements of each of the variables (genes/proteins/metabolites) for say different individuals, and want to determine which variables are connected, or use their connection for statistical analysis.
Statistical Analysis of Known Networks

Suppose we have the following information:

- the network
- activities of individual nodes (genes, proteins, etc)
Statistical Analysis of Known Networks

Suppose we have the following information:

- the network
- activities of individual nodes (genes, proteins, etc)

How can we identify the important nodes?

and what does this even mean?
How can we identify the important nodes?

- We can select the significant nodes based on p-values, after adjusting for multiple comparisons (FDR, etc)
Statistical Analysis of Known Networks

How can we identify the important nodes?

- We can select the significant nodes based on p-values, after adjusting for multiple comparisons (FDR, etc)
- But the signal is often weak for lots of tests

- If we believe the network is informative, it may make sense to use the network to guide our selection
Statistical Analysis of Known Networks

How can we identify the important nodes using the network?

The simplest option is to limit our search/testing to the central nodes in the network:

I Nodes connected to many other nodes, aka hub nodes
I Nodes that are close to many other nodes (closeness)
I Nodes that are on many network paths (betweenness)

Alternatively, we can perform pathway enrichment analysis to identify sets of significant nodes (later).
Statistical Analysis of Known Networks

How can we identify the important nodes using the network?

The simplest option is to limit our search/testing to the central nodes in the network:
- Nodes connected to many other nodes, aka hub nodes
- Nodes that are close to many other nodes (closeness)
- Nodes that are on many network paths (betweenness)

Alternatively, we can perform pathway enrichment analysis to identify sets of significant nodes (later)
Example: Functional Relevance of Hub Nodes

- Inferred genetic interaction network of cancer-related pathway in prostate cancer (data from TCGA)
- Hubs defined as nodes whose degrees are at the 75th percentile of the degree distribution

Other Measures of Centrality

- **Closeness**: Total distance of each node to other nodes:
  \[
  cl_j = \left( \sum_{k \in V} d(j, k) \right)^{-1}
  \]
  where \(d(j, k)\) is the (shortest path) distance between \(j\) and \(k\).
- **Betweenness**: The number of paths that go through a node:
  \[
  bw_j = \sum_{i \neq j \neq k} \frac{\pi_{ik}(j)}{\pi_{ik}}
  \]
  where \(\pi_{ik}(j)\) is the number of paths between \(i\) and \(k\) that go through \(j\), and \(\pi_{ik}\) is the total number of paths between them.
So, we can use the network information to identify differentially expressed genes with important function.

But, what if the network is not known?
Signal Detection in Known Networks

So, we can use the network information to identify differentially expressed genes with important function.

But, what if the network is not known?  
We need to infer the network from data!

Why Do We Need Network Inference?

- Despite progress, our knowledge of interactions is limited.
- The entire genome is a vast landscape, and experiments for discovering networks are very expensive.
- From a statistical point of view, network estimation is related to estimation of covariance matrices, which has many independent applications in statistical inference and prediction (more about this later).
- Finally, and perhaps most importantly, gene and protein networks are dynamic and changes in these networks have been attributed to complex diseases.
Network Inference — An Overview

Two general classes of network inference methods:

- Methods based on **marginal measures of association**:
  - Co-expression Networks (uses linear measures of association)
  - Methods based on **mutual information** (can accommodate non-linear associations)

- Methods based on **conditional measures of association**:
  - Methods assuming (multivariate) normality (*glasso*, etc)
  - Generalizations to allow for nonlinear dependencies (*nonparanormal*, etc)
Our Plan

In the remainder of this module, we will cover the following topics:

▶ Methods for learning undirected networks
  ▶ Co-expression Networks (WGCNA)
  ▶ ARACNE
  ▶ Conditional Independence Graphs
    ▶ Gaussian Observations (glasso, etc)
    ▶ Non-Gaussian and Non-Linear Data (nonparanormal, etc)

▶ Methods for learning directed networks
  ▶ Bayesian Networks (basic concepts, reconstruction algorithm)
  ▶ Learning directed networks from time-course data (dynamic Bayesian networks)
  ▶ Learning directed networks from perturbation screens

▶ Topology-based pathway enrichment analysis
Pathway & Network Analysis of Omics Data: Undirected Graphical Models - I

Ali Shojaie
Department of Biostatistics
University of Washington
faculty.washington.edu/ashojaie

Summer Institute for Statistical Genetics – 2018

An Overview of Network Reconstruction Methods

Network reconstruction methods can be categorized into two general classes:
Network reconstruction methods can be categorized into two general classes:

- Methods based on **marginal measures of association**:
  - Co-expression Networks (uses linear measures of association)
  - Methods based on mutual information (can accommodate non-linear associations)

- Methods based on **conditional measures of association**:
  - Methods assuming multivariate normality/normality
  - Generalizations to allow for nonlinear dependencies
Co-Expression/Correlation Networks

- This is the simplest (and most-widely used!!) method for estimating networks; it assumes that edges correspond to large correlation magnitudes.
Co-Expression/Correlation Networks

- This is the simplest (and most-widely used!!) method for estimating networks; it assumes that edges correspond to large correlation magnitudes.
- Let $r(i,j)$ be correlation between $X_i$ and $X_j$; we claim an edge between $i$ and $j$ if $|r(i,j)| > \tau$.

Correlation is a simple measure of linear association between two random variables.

Here, $\tau$ is a user-specified threshold, and is the tuning parameter for this method.
Co-Expression/Correlation Networks

- This is the simplest (and most-widely used!!) method for estimating networks; it assumes that edges correspond to large correlation magnitudes.
- Let $r(i,j)$ be correlation between $X_i$ and $X_j$; we claim an edge between $i$ and $j$ if $|r(i,j)| > \tau$.
- Correlation is a simple measure of linear association between two random variables.
- Here, $\tau$ is a user-specified threshold, and is the tuning parameter for this method.
- By construction, this is an undirected network (correlation is symmetric).

Limitations of Co-Expression Networks

- The estimation is highly dependent on the choice of $\tau$.
- They may not correctly detect the edges in biological networks: two genes/proteins can have high correlations, even if they don’t interact with each other!
- Correlation is a measure of linear association, but many biological relationships are nonlinear.
Limitations of Co-Expression Networks

- The estimation is highly dependent on the choice of $\tau$.
- They may not correctly detect the edges in biological networks: two genes/proteins can have high correlations, even if they don’t interact with each other!
Limitations of Co-Expression Networks

- The estimation is highly dependent on the choice of \( \tau \).
- They may not correctly detect the edges in biological networks: two genes/proteins can have high correlations, even if they don’t interact with each other!
- Correlation is a measure of linear association, but many biological relationships are nonlinear.
Limitations of Co-Expression Networks

- The estimation is highly dependent on the choice of \( \tau \).
- We can instead test \( H_0 : r_{xy} = 0 \)

A commonly used test is given by the Fisher transformation

\[
Z = \frac{1}{2} \ln \left( \frac{1 + r}{1 - r} \right) = \text{artanh}(r) \sim_{H_0} N(0, \frac{1}{\sqrt{n - 3}})
\]
Limitations of Co-Expression Networks

- The estimation is highly dependent on the choice of $\tau$.
  - We can instead test $H_0: r_{xy} = 0$
  - A commonly used test is given by the Fisher transformation

$$Z = \frac{1}{2} \ln \left( \frac{1 + r}{1 - r} \right) = \tanh(r) \sim_{H_0} N(0, \frac{1}{\sqrt{n - 3}})$$

- Alternatively, we can work with “weighted” co-expression networks

---

Weighted Gene Co-expression Network Analysis

Weighted Gene Co-expression Network Analysis

- Measure concordance of gene expression using Pearson correlation
- Continuously transform the Pearson correlations into an (soft) adjacency function → weighted network
  - using the sigmoid adjacency function
    \[ A_{ij} = \frac{1}{1 + e^{-\alpha (r_{ij} - \tau_0)}} \]
  - using the power adjacency function
    \[ A_{ij} = |r_{ij}|^\beta \]

1 Zang and Horvath, A General Framework for Weighted Gene Co-Expression Network Analysis, Stat App in Gen and Mol Bio, 2005
**Weighted Gene Co-expression Network Analysis**

- Measure concordance of gene expression using **Pearson correlation**
- Continuously transform the Pearson correlations into an (soft) adjacency function → **weighted network**
  - using the sigmoid adjacency function
  \[
  A_{ij} = \frac{1}{1 + e^{-\alpha(r_{ij}-r_0)}}
  \]
  - using the power adjacency function
  \[
  A_{ij} = |r_{ij}|^\beta
  \]
- Perform downstream network analysis (clustering, etc) on weighted networks


---

**Choice of Parameters**

- By changing the tuning parameters, adjacency functions behave similar to hard thresholding
Choice of Parameters

- By changing the tuning parameters, adjacency functions behave similar to hard thresholding.

- Power and sigmoid adjacency functions lead to similar results if the parameters are chosen to achieve scale-free topology.
- We focus on power adjacency function.
Choice of Parameters

- Using $\beta \approx 6$ gives a scale free network

Software

- Implemented in the R-package WGCNA
  ```
  install.packages('WGCNA',lib=NULL,repos='http://cran.us.r-project.org')
  ```
- Main estimation function
  ```
  adjacency(datExpr,
  selectCols = NULL,
  type = "unsigned",
  power = if (type=="distance") 1 else 6,
  corFnc = "cor", corOptions = "use = 'p'",
  distFnc = "dist", distOptions = "method = 'euclidean'")
  ```
- To determine the power so that the network has scale-free distribution, need to search for multiple powers
Limitations of Co-Expression Networks

- Correlation is a measure of linear association, but many biological relationships are nonlinear.
Limitations of Co-Expression Networks

- Correlation is a measure of linear association, but many biological relationships are nonlinear
  - We can use other measures of association, for instance, Spearman correlation or Kendal’s $\tau$.
    - These methods define correlation between two variables, based on the ranking of observations, and not their exact values
    - They can better capture non-linear associations
Limitations of Co-Expression Networks

- Correlation is a measure of linear association, but many biological relationships are nonlinear.
  - We can use other measures of association, for instance, Spearman correlation or Kendall’s τ.
  - These methods define correlation between two variables, based on the ranking of observations, and not their exact values.
  - They can better capture non-linear associations.
- We can instead use mutual information; this has been used in many algorithm, including ARACNE.

---

ARACNE: Algorithm for the Reconstruction of Accurate Cellular NEtworks\(^2\)

ARACNE: Algorithm for the Reconstruction of Accurate Cellular Networks\textsuperscript{2}

1. Identifies statistically significant gene-gene co-regulation based on mutual information

2. It then eliminates indirect relationships in which two genes are co-regulated through one or more intermediates

\textsuperscript{2}ARACNE: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context, Margolin et al, BMC Bioinfo, 2006
**Marginal Association Networks**  
**Conditional Independence Networks**  
**WGCNA**  
**ARACNE**

**ARACNE**

Data Processing Inequality (DPI)

\[ A \rightarrow B \rightarrow C \]
Data Processing Inequality (DPI)

\[ I(A, C) \leq \min[I(A, B), I(B, C)] \]

where

\[ I(g_i, g_j) = \log \frac{P(g_i, g_j)}{P(g_i)P(g_j)} \]

- Look at every triplet and remove the weakest link
- Need to estimate marginal and joint (pairwise) probabilities
  (using Gaussian Kernel)
Algorithm Details

- Starts with a network where each triplet of genes is connected by an edge.
Algorithm Details

- Starts with a network where each triplet of genes is connected by an edge.
- The algorithm then examines each gene triplet for which all pairwise MIs are greater than a cut-off and removes the edge with the smallest value based on DPI.
  - Each triplet is analyzed irrespectively of whether its edges have been selected for removal by prior DPI applications to different triplets.
  - The least of the three MIs can come from indirect interactions only, and checking against the DPI may identify gene pairs that are not independent but still do not interact.
Rationale and Guarantees

- If MIs can be estimated with no errors, then ARACNE reconstructs the underlying interaction network exactly, provided this network is a tree and has only pairwise interactions.
- The maximum MI spanning tree is a subnetwork of the network built by ARACNE.

**Theorem.** Let $\pi_{ik}$ be the set of nodes forming the shortest path in the network between nodes $i$ and $k$. Then, if MIs can be estimated without errors, ARACNE reconstructs an interaction network without false positives edges, provided: (a) the network consists only of pairwise interactions, (b) for each $j \in \pi_{ik}$, $I_{ij} \geq I_{ik}$. Further, ARACNE does not produce any false negatives, and the network reconstruction is exact iff (c) for each directly connected pair $ij$ and for any other node $k$, we have $I_{ij} > \min[I_{ik}, I_{jk}]$. 
Performance on Synthetic Data

Application: B-lymphocytes Expression Data
Application: B-lymphocytes Expression Data

- MYC (proto-oncogene) subnetwork (2063 genes)
- 29 of the 56 (51.8%) predicted first neighbors biochemically validated as targets of the MYC transcription factor.
- New candidate targets were identified, 12 experimentally validated.
  - 11 proved to be true targets.
- The candidate targets that have not been validated are possibly also correct.

Software

- Implemented in the R-package minet:
  ```r
source("http://bioconductor.org/biocLite.R")
biocLite("minet")
```
- Main estimation function `aracne(mim, eps=0)`
  - `mim`: mutual information matrix
    ```r
    mim <- build.mim(syn.data, estimator="spearman")
    ```
  - `eps`: threshold for setting an edge to zero, prior to searching over triplets
Limitations of Co-Expression Networks

- The estimation is highly dependent on the choice of $\tau$.
- They may not correctly detect the edges in biological networks: two genes/proteins can have high correlations, even if they don’t interact with each other!
Limitations of Co-Expression Networks

- The estimation is highly dependent on the choice of $\tau$
- They may not correctly detect the edges in biological networks: two genes/proteins can have high correlations, even if they don’t interact with each other!
Limitations of Co-Expression Networks

- The estimation is highly dependent on the choice of \( \tau \)
- They may not correctly detect the edges in biological networks: two genes/proteins can have high correlations, even if they don’t interact with each other!

Partial Correlation
Partial Correlation

- Partial correlation measures the correlation between \( i \) and \( j \) after the effect of the other variables are removed.

- In our example, this means that we would be taking into account that the "information" was passed through mutual friends, and not directly.

Mathematically, the partial correlation between \( X_i \) and \( X_j \) given \( X_k \) is given by:

\[
\rho_{ij|k} = \frac{\rho_{ij} - \rho_{ik} \cdot \rho_{jk}}{\sqrt{1 - \rho_{ik}^2} \cdot \sqrt{1 - \rho_{jk}^2}}
\]
Partial Correlation

- Partial correlation measures the correlation between $i$ and $j$ after the effect of the other variables are removed.
- In our example, this means that we would be taking into account that the “information” was passed through mutual friends, and not directly.
- This gives a more direct connection to biological networks; in PPI networks: if protein $A$ binds with $B$ and $C$, but $B$ and $C$ don’t bind, then the correlation between $B$ and $C$ will be removed once conditioned on $A$.

Mathematically, the partial correlation between $X_i$ and $X_j$ given $X_k$ is given by:

$$
\rho_{ij:k} = \frac{\rho_{ij} - \rho_{ik}\rho_{jk}}{\sqrt{1 - \rho_{ik}^2}\sqrt{1 - \rho_{jk}^2}}.
$$
Partial Correlation

- Partial correlation is also symmetric.
- Partial correlation is also a number between -1 and 1.
- In partial correlation networks, we draw an edge between $X$ and $Y$, if the partial correlation between them is large.
- Calculation of partial correlation is more difficult.
- Again, we can determine this using testing, however, we need a larger sample size.
- New statistical methods have been proposed in the past couple of years to make this possible... (active area of research)

A simple example

$$\text{Correlation} = \begin{bmatrix} 1 & -0.8 & 0.7 \\ -0.8 & 1 & -0.8 \\ 0.7 & -0.8 & 1 \end{bmatrix} \quad \text{PartialCorr} = \begin{bmatrix} 1 & 0.6 & 0 \\ 0.6 & 1 & 0.6 \\ 0 & 0.6 & 1 \end{bmatrix}$$
A simple example

\[
\text{Correlation} = \begin{bmatrix}
1 & -0.8 & 0.7 \\
-0.8 & 1 & -0.8 \\
0.7 & -0.8 & 1
\end{bmatrix}
\quad \text{PartialCorr} = \begin{bmatrix}
1 & 0.6 & 0 \\
0.6 & 1 & 0.6 \\
0 & 0.6 & 1
\end{bmatrix}
\]

A larger example
A larger example

- A network with 10 nodes and 20 edges
- \( n = 100 \) observations
A larger example

- A network with 10 nodes and 20 edges
- \( n = 100 \) observations
- Estimation using correlation & partial correlation (20 edges)
Partial Correlation for Gaussian Random Variables

- It turns out, we can calculate the partial correlation between $X_i$ and $X_j$ given all other variables, by calculating the inverse of the empirical covariance matrix $S$. 
Partial Correlation for Gaussian Random Variables

- It turns out, we can calculate the partial correlation between $X_i$ and $X_j$ given all other variables, by calculating the inverse of the empirical covariance matrix $S$.
- In other words, the $(i,j)$ entry in $\Sigma^{-1}$ gives the partial correlation between $X_i$ and $X_j$ given all other variables $X_{\setminus i,j}$.
- Now suppose the variables are connected by a graph $G$, then if $X \sim N(0, \Sigma)$, the nonzero entries in the inverse covariance matrix correspond to the edges of $G$: $(i,j) \in E$ iff $\Sigma_{ij}^{-1} \neq 0$. 
Partial Correlation for Gaussian Random Variables

\begin{align*}
\begin{pmatrix}
-1 & x & 0 \\
x & -1 & x \\
0 & x & -1 \\
\end{pmatrix}
& \quad \begin{pmatrix}
-1 & x & x & 0 \\
x & -1 & x & 0 \\
x & x & -1 & 0 \\
0 & 0 & 0 & -1 \\
\end{pmatrix} \\
\begin{pmatrix}
-1 & x & 0 & x \\
x & -1 & x & 0 \\
0 & x & -1 & x \\
x & 0 & x & -1 \\
\end{pmatrix}
& \quad \begin{pmatrix}
-1 & 0 & 0 & x \\
0 & -1 & x & 0 \\
0 & x & -1 & x \\
x & 0 & x & -1 \\
\end{pmatrix}
\end{align*}


Estimation

Therefore, to estimate the edges in the graph $G$,

- First, calculate the empirical covariance matrix of the observations $S = 1/(n - 1)X^TX$ (remember $X$ is $n \times p$).  

Estimation

Therefore, to estimate the edges in the graph $G$,

- First, calculate the empirical covariance matrix of the observations $S = 1/(n-1)X^TX$ (remember $X$ is $n \times p$).
- Then, find the inverse of $S$. Non-zero values of this matrix determine where there are edges in the network.

This seems pretty simple, however, in practice this may not work that well, even if the sample size is very large!!
**Estimation**

Therefore, to estimate the edges in the graph $G$,

- First, calculate the **empirical covariance matrix** of the observations $S = 1/(n - 1)X^TX$ (remember $X$ is $n \times p$).
- Then, **find the inverse of $S$**. Non-zero values of this matrix determine where there are edges in the network.
- This seems pretty simple, however, in practice this may not work that well, even if the sample size is very large!!

![True Graph vs. Est Graph](image_url)

**Difficulties in HD**

- A number of problems arise in high dimensional settings, especially when $p > n$.
- If $p < n$, but $p \approx n$, we may get poor estimates, i.e., lots of false positives and false negatives.
Difficulties in HD

- A number of problems arise in high dimensional settings, especially when $p \gg n$.

- First, $S$ is not invertible if $p > n$!
Difficulties in HD

- A number of problems arise in high dimensional settings, especially when \( p \gg n \).
- First, \( S \) is not invertible if \( p > n \! \).
- Even if \( p < n \), but \( p \approx n \), we may get poor estimates, i.e., lots of false positives and false negatives.
A number of methods have been proposed for estimation of conditional independence graphs from Gaussian observations in high dimensions.
A number of methods have been proposed for estimation of conditional independence graphs from Gaussian observations in high dimensions.

The main idea in most of these methods is to use a regularization penalty, like the lasso.

**The Lasso**

The lasso involves finding $\beta$ that minimizes

$$\left\| y - \sum_{k=1}^{p} X_k \beta_k \right\|^2 + \lambda \sum_j |\beta_k|.$$
The Lasso

- The lasso involves finding $\beta$ that minimizes
  \[ \left\| y - \sum_{k=1}^{p} X_k \beta_k \right\|^2 + \lambda \sum_j |\beta_j|. \]

- Here $\lambda$ is a tuning parameter
  - When $\lambda = 0$, we get least squares!
  - When $\lambda$ is very large, we get $\hat{\beta} = 0$.

- Equivalently, find $\beta$ that minimizes
  \[ \left\| y - \sum_{k=1}^{p} X_k \beta_k \right\|^2 \]
  subject to the constraint that
  \[ \sum_{k=1}^{p} |\beta_k| \leq s. \]
A Geometric Interpretation

Lasso As $\lambda$ Varies
The idea in the first method, called neighborhood selection, is to estimate the graph by fitting a penalized regression of each variable on all other variables.
Estimation in High Dimensions – Method 1

- The idea in the first method, called *neighborhood selection*, is to estimate the graph by fitting a *penalized regression of each variable on all other variables*.
- In other words, we solve, for \( j = 1, \ldots, p \)
  \[
  \| X_j - \sum_{k \neq j} X_k \beta_k \|^2 + \lambda \sum_{k \neq j} |\beta_k | 
  \]

- The final estimate of the graph is obtained by getting all of the edges found from these individual regression problems.
In the second approach, called graphical lasso, we directly estimate the inverse covariance matrix by maximizing the $\ell_1$ penalized log likelihood.
Estimation in High Dimensions – Method 2

- In the second approach, called graphical lasso, we directly estimate the inverse covariance matrix by maximizing the $\ell_1$ penalized log likelihood.
- Turns out, the log likelihood function of (mean 0) Gaussian random variables can be written as

$$\log \det(\Theta) - \text{tr}(S\Theta),$$

where $\Theta$ is the $p \times p$ inverse covariance (aka precision) matrix.
Comparing the Two Approaches

- It turns out that the neighborhood selection approach is an approximation to the graphical lasso problem:
Comparing the Two Approaches

- It turns out that the neighborhood selection approach is an approximation to the graphical lasso problem:
  - Consider regression of $X_j$ on $X_k, j \neq k$
  - Then the regression coefficient for neighborhood selection is related to the $j, k$ element of $\Theta$:
    \[
    \beta_k = \frac{-\Theta_{jk}}{\Theta_{jj}}
    \]

- A main difficulty with the neighborhood selection approach is that the resulting graph is not necessarily symmetric.
Comparing the Two Approaches

- It turns out that the neighborhood selection approach is an approximation to the graphical lasso problem:
  - Consider regression of $X_j$ on $X_k, j \neq k$
  - Then the regression coefficient for neighborhood selection is related to the $j, k$ element of $\Theta$:
    \[
    \beta_k = -\frac{\Theta_{jk}}{\Theta_{jj}}
    \]

- A main difficulty with the neighborhood selection approach is that the resulting graph is not necessarily symmetric.

- To deal with this, we can take the union or intersection of edges from regressing $X_k$ on $X_k$ and $X_j$ on $X_k$; however, this is an ad hoc solution.

- On the other hand, neighborhood selection is computationally more efficient, and may gives better estimates.
A Real Example

- Flow cytometry allows us to obtain measurements of proteins in individual cells, and hence facilitates obtaining datasets with large sample sizes.
A Real Example

- **Flow cytometry** allows us to obtain measurements of proteins in individual cells, and hence facilitates obtaining datasets with large sample sizes.
- Sachs et al (2003) conducted an experiment and gathered data on \( p = 11 \) proteins measured on \( n = 7466 \) cells.
Choice of tuning parameter

Unlike supervised learning, choosing the right \( \lambda \) is very difficult in this case.
Choice of tuning parameter

- Unlike supervised learning, choosing the right $\lambda$ is very difficult in this case.
- As the previous example shows, as $\lambda$ gets larger, we get sparser graphs.

However, there is no systematic way of choosing the right $\lambda$. A number of methods have been proposed, based on the idea of trying to control the false positives, but this is still the topic of ongoing research.

One option for choosing $\lambda$ controls the probability of falsely connecting disconnected components at level $\lambda$ (Banerjee et al, 2008). When variables are standardized, this gives:

$$\lambda = t_{n-2d} \left( \frac{\lambda}{2n} \right),$$

where $t_{n-2d}$ is the (100 $\%$ quantile of $t$-distribution with $n-2d$ degrees of freedom.
Choice of tuning parameter

- Unlike supervised learning, choosing the right $\lambda$ is very difficult in this case.
- As the previous example shows, as $\lambda$ gets larger, we get sparser graphs.
- However, there is no systematic way of choosing the right $\lambda$.
- A number of methods have been proposed, based on the idea of trying to control the false positives, but this is still the topic of ongoing research.

One option for choosing $\lambda$ controls the probability of falsely connecting disconnected components at level $\alpha$ (Banerjee et al, 2008). When variables are standardized, this gives:

$$
\lambda(\alpha) = \frac{t_{n-2}(\alpha/2p^2)}{\sqrt{n - 2 + t_{n-2}(\alpha/2p^2)}},
$$

where $t_{n-2}(\alpha)$ is the (100 $-\alpha$)% quantile of $t$-distribution with $n - 2$ d.f.
The penalized estimation methods discussed above allow estimation of graphical models in the $p \gg n$ settings, e.g. when $p$ is in 1000’s and $n$ is in 100’s.
Some Comments

► The penalized estimation methods discussed above allow estimation of graphical models in the $p \gg n$ settings, e.g. when $p$ is in 1000’s and $n$ is in 100's.

► However, both of these methods, and most other methods for estimation of conditional independence networks, work when the network is sparse.

► Sparsity means that there are not many edges in the network, and the network is far from fully connected.
Some Comments

- The penalized estimation methods discussed above allow estimation of graphical models in the $p \gg n$ settings, e.g. when $p$ is in 1000's and $n$ is in 100's.
- However, both of these methods, and most other methods for estimation of conditional independence networks, work when the network is sparse.
- Sparsity means that there are not many edges in the network, and the network is far from fully connected.
- Good news is that biological networks are believed to be “sparse”. However, all of these concepts are theoretical and it is difficult to assess how things work on real networks.

Computation

- As we saw previously, the neighborhood selection problem is an approximation to the graphical lasso problem.
Computation

- As we saw previously, the neighborhood selection problem is an approximation to the graphical lasso problem.
- It turns out that this relationship can be used for solving the graphical lasso problem efficiently.

The idea is to iterate over $p$ regression problems, one for each column of the precision matrix.

The algorithm, as well as the approximation for the neighborhood selection problem, is implemented in the R-package *glasso*. In practice, often better to use the empirical correlation matrix.
Computation

- As we saw previously, the neighborhood selection problem is an approximation to the graphical lasso problem.
- It turns out that this relationship can be used for solving the graphical lasso problem efficiently.
- The idea is to iterate over $p$ regression problems, one for each column of the precision matrix.
- This results in a very efficient algorithm for solving this problem, and in practice, we can solve problems with $p$ in 1000’s and $n$ in 100’s in a few minutes.

The algorithm, as well as the approximation for the neighborhood selection problem, is implemented in the R-package glasso.
Computation

- As we saw previously, the neighborhood selection problem is an approximation to the graphical lasso problem.
- It turns out that this relationship can be used for solving the graphical lasso problem efficiently.
- The idea is to iterate over $p$ regression problems, one for each column of the precision matrix.
- This results in a very efficient algorithm for solving this problem, and in practice, we can solve problems with $p$ in 1000’s and $n$ in 100’s in a few minutes.
- The algorithm, as well as the approximation for the neighborhood selection problem, is implemented in the R-package glasso.
- In practice, often better to use the empirical correlation matrix

An Example in R

- Install the R-package glasso

```r
library(glasso)

##Read the covariance matrix
sachs <- as.matrix(read.table("sachscov.txt"))
dim(sachs)

##glasso
est.1 <- glasso(s=sachs, rho=5, approx=FALSE, penalize.diagonal=FALSE)

##neighborhood selection
est.2 <- glasso(s=sachs, rho=5, approx=TRUE, penalize.diagonal=FALSE)
```
**Exercise**

- Estimate the graph from the previous example with different values of tuning parameter (Note: this is \( \rho \) in the code).
- Try the estimation with and without setting \texttt{penalize.diagonal=FALSE}. What do you see?
- Try the estimation with the empirical correlation matrix instead (you may find \texttt{cov2cor()} useful). What do you see?

---

**Marginal vs Conditional Associations**

- Partial correlations provide a better representation of edges in biological networks.
- Computationally, estimating the conditional independence graph is almost as costly as estimating the co-expression network (neighborhood selection gives a good approximation at similar computational cost).
- Estimation and inference using marginal associations can be done with much smaller samples.
- The most important difference, however, is the idea of conditioning! Partial correlation works if we condition on the right set of variables. Marginal associations on the other hand, is independent of conditioning.
Marginal vs Conditional Associations

- Partial correlations provide a better representation of edges in biological networks.

- Computationally, estimating the conditional independence graph is almost as costly as estimating the co-expression network (neighborhood selection gives a good approximation at similar computational cost).
Marginal vs Conditional Associations

- Partial correlations provide a better representation of edges in biological networks.
- Computationally, estimating the conditional independence graph is almost as costly as estimating the co-expression network (neighborhood selection gives a good approximation at similar computational cost).
- Estimation and inference using marginal associations can be done with much smaller samples.

The most important difference, however, is the idea of conditioning! Partial correlation works if we condition on the right set of variables. Marginal associations on the other hand, is independent of conditioning.
Final Thoughts

- Estimation of graphical models is an important but challenging problem.
Final Thoughts

- Estimation of graphical models is an important but challenging problem.
- The appropriate method depends on the design of experiment, available data and sample size.
Final Thoughts

- Estimation of graphical models is an important but challenging problem.
- The appropriate method depends on the design of experiment, available data and sample size.
- Choosing the tuning parameter is a challenging problem in both cases.
- Often difficult to validate the estimates; however, in biological networks, can compare our findings with known interactions from literature (sort of!).
Non-linear associations
Non-linear associations

Recall that correlation is a measure of linear dependence, this is also true about partial correlation.

However, many real-world associations are non-linear.

Therefore, (partial) correlation may miss non-linear associations among variables.
Non-linear associations

- Recall that correlation is a measure of linear dependence, this is also true about partial correlation.
- However, many real-world associations are non-linear
- Therefore, (partial) correlation may miss non-linear associations among variables
- Mutual information-based methods (ARACNE etc) try to address this issue
  - calculating conditional mutual information is computationally expensive
  - ARACNE’s solution for removing indirect associations is ad-hoc

Linearity and Normality

---

Linearity and Normality

- Need methods for estimation of graphical models with non-linear associations

- Interestingly, assuming linear associations is closely related to multivariate normality (MVN):
  - MVN ⇒ linear relationships
  - linear dependencies (+ extra mild assumptions) ⇒ MVN

---

Linearity and Normality

- Need methods for estimation of graphical models with non-linear associations
- Interestingly, assuming linear associations is closely related to multivariate normality (MVN):
  - MVN ⇒ linear relationships
  - linear dependencies (+ extra mild assumptions) ⇒ MVN\(^1\)
- Both of these are strong assumptions and may not hold in real-world applications!

\(^1\)Khatri & Rao (1976) & Fisk (1970)

Our Plan
Our Plan

- We will start by discussing the general notion of conditional independence graphs (aka Markov Random Fields)
- We will then discuss three classes of models:
  - Transformation-based and robust methods for handling non-Gaussianity
  - Parametric graphical models with non-Gaussian variables
  - Semi- and non-parametric approaches for flexible estimation of graphical models
In case of Gaussian variables, $\Theta_{jk} = 0$ implies that $X_j$ and $X_k$ are conditionally independent.
Conditional Independence Graphs

- In case of Gaussian variables, $\Theta_{jk} = 0$ implies that $X_j$ and $X_k$ are conditionally independent.
- Conditional dependence is a general notion that defines the class of conditional independent graphs (CIG). In CIG,
  - $X \perp Y \mid Z$ iff $P(X = x, Y = y \mid Z = z) = P(X = x \mid Z = z)P(Y = y \mid Z = z)$
  - If $X$ and $Y$ are neighbors $(X \rightarrow Y)$, they are conditionally dependent
  - $X$ is conditionally independent of all other nodes, given $\text{neighbors}(X)$: $Z \notin \text{neighbors}(X)$, then $X \perp Z \mid \text{neighbors}(X)$
Suppose $X \sim N(0, \Sigma)$, but there exists monotone functions $f_j, j = 1, \ldots, p$ such that $[f_1(X_1), \ldots, f_p(X_p)] \sim N(0, \Sigma)$. We say that $X$ has a nonparanormal distribution $X \sim NPN_p(f, \Sigma)$. If $f$ and $\Sigma$ are parameters of the distribution, and need to be estimated from data. For continuous distributions, the nonparanormal family is equivalent to the Gaussian copula family. To estimate the nonparanormal network:

1. transform the data $[f_1(X_1), \ldots, f_p(X_p)]$
2. estimate the network of the transformed data (e.g. calculate the empirical covariance matrix of the transformed data, and apply glasso or neighborhood selection).
Nonparanormal (Gaussian Copula) Models

- Suppose $X \sim N(0, \Sigma)$, but there exists monotone functions $f_j, j = 1, \ldots, p$ such that $[f_1(X_1), \ldots, f_p(X_p)] \sim N(0, \Sigma)$
  - We say that $X$ has a nonparanormal distribution $X \sim NPN_p(f, \Sigma)$.
  - $f$ and $\Sigma$ are parameters of the distribution, and need to be estimated from data.
  - For continuous distributions, the nonparanormal family is equivalent to the Gaussian copula family.

- To estimate the nonparanormal network:
  i) transform the data: $[f_1(X_1), \ldots, f_p(X_p)]$
  ii) estimate the network of the transformed data (e.g. calculate the empirical covariance matrix of the transformed data, and apply glasso or neighborhood selection)
A Related Procedure

- Liu et al (2012) and Xue & Zou (2012) proposed a closely related idea using rank-based correlation
  - Let $r_i^j$ be the rank of $x_i^j$ among $x_1^j, \ldots, x_n^j$ and $\bar{r}_j = (n + 1)/2$ be the average rank
  - Calculate Spearman’s $\rho$ or Kendall’s $\tau$

\[
\hat{\rho}_{jk} = \frac{\sum_{i=1}^{n} (r_i^j - \bar{r}_j)(r_k^j - \bar{r}_k)}{\sqrt{\sum_{i=1}^{n} (r_i^j - \bar{r}_j)^2 \sum_{i=1}^{n} (r_k^i - \bar{r}_k)^2}}
\]

\[
\hat{\tau}_{jk} = \frac{2}{n(n-1)} \sum_{1 \leq i < i' \leq n} \text{sign} \left( (x_i^j - x_{i'}^j)(x_k^i - x_{k'}^i) \right)
\]
A Related Procedure

- Liu et al (2012) and Xue & Zou (2012) proposed a closely related idea using rank-based correlation
  - Let $r_j^i$ be the rank of $x_j^i$ among $x_1^j, \ldots, x_n^j$ and $\bar{r}_j = (n + 1)/2$ be the average rank
  - Calculate Spearman’s $\rho$ or Kendall’s $\tau$

$$
\hat{\rho}_{jk} = \frac{\sum_{i=1}^{n} (r_j^i - \bar{r}_j)(r_k^i - \bar{r}_k)}{\sqrt{\sum_{i=1}^{n} (r_j^i - \bar{r}_j)^2 \sum_{i=1}^{n} (r_k^i - \bar{r}_k)^2}} \\
\hat{\tau}_{jk} = \frac{2}{n(n-1)} \sum_{1 \leq i < i' \leq n} \text{sign} \left( (x_j^i - x_j^{i'}) (x_k^i - x_k^{i'}) \right)
$$

- If $X \sim NPN_p(f, \Sigma)$, then $\Sigma_{jk} = 2\sin(\rho_{jk}\pi/6) = \sin(\tau_{jk}\pi/2)$

- Therefore, we can estimate $\Sigma^{-1}$ by plugging in rank-based correlations into graphical lasso (R-package huge)
A Real Data Example

- Protein cytometry data for cell signaling (Sachs et al, 2005)
- Transform the data using Gaussian copula (Liu et al, 2009), giving marginal normality

- Pairwise relationships seem non-linear

- Shapiro-Wilk test rejects multivariate normality: 
  \[ p < 2 \times 10^{-16} \]
Graphical Models for Discrete Random Variables

- In many cases, biological data are not Gaussian: SNPs, RNAseq, etc
- Need to estimate CIG for other distributions: binomial, poisson, etc
Graphical Models for Discrete Random Variables

- In many cases, biological data are not Gaussian: SNPs, RNAseq, etc
- Need to estimate CIG for other distributions: binomial, poisson, etc
- Unfortunately, for these distribution, the problem does not have a closed-form!
- A special case, which is computationally more tractable, is the class of pairwise MRFs

Pairwise Markov Random Fields
Pairwise Markov Random Fields

- The idea of pairwise MRFs is to “assume” that only two-way interactions among variables exist.
- The pairwise MRF associated with the graph $G$ over the random vector $X$ is the family of probability distributions $P(X)$ that can be written as
  \[ P(X) \propto \exp \sum_{(j,k) \in E} \phi_{jk}(x_j, x_k) \]
- For each edge $(j, k) \in E$, $\phi_{jk}$ is called the edge potential function.
- For discrete random variables, any MRF can be transformed to an MRF with pairwise interactions by introducing additional variables (Wainwright & Jordan, 2008).

Graphical Models for Binary Random Variables
Graphical Models for Binary Random Variables

- Suppose $X_1, \ldots, X_p$ are binary random variables, corresponding to e.g. SNPs, or DNA methylation.

- A special case of discrete graphical models is the Ising model for binary random variables.

$$P_\theta(x) = \frac{1}{Z(\theta)} \exp \left\{ \sum_{(j,k) \in E} \theta_{jk} x_j x_k \right\}$$
Graphical Models for Binary Random Variables

- Suppose $X_1, \ldots, X_p$ are binary random variables, corresponding to e.g. SNPs, or DNA methylation
- A special case of discrete graphical models is the Ising model for binary random variables

$$P_\theta(x) = \frac{1}{Z(\theta)} \exp \left\{ \sum_{(j,k) \in E} \theta_{jk}x_jx_k \right\}$$

- A pairwise MRF for binary data, with $\phi_{jk}(x_j, x_k) = \theta_{jk}x_jx_k$
- $x^i \in \{-1, +1\}^p$
- The partition function $Z(\theta)$ ensures that distribution sums to 1
- $(j, k) \in E$ iff $\theta_{jk} \neq 0!$

---

\(^2\text{Ravikumar et al (2010)}\)
Graphical Models for Binary Random Variables

- We can consider a neighborhood selection\(^2\) approach with an \(\ell_1\) penalty to find the neighborhood of each node

\[ N(j) = \{ k \in V : (j, k) \in E \} \]

\(^2\)Ravikumar et al (2010)

Graphical Models for Binary Random Variables

- We can consider a neighborhood selection\(^2\) approach with an \(\ell_1\) penalty to find the neighborhood of each node 
\(N(j) = \{ k \in V : (j, k) \in E \} \)

- For \(j = 1, \ldots, p\), need to solve (after some algebra)

\[
\min_\theta \left\{ n^{-1} \sum_{i=1}^{n} \left[ f(\theta; x^i) - \sum_{k \neq j} \theta_{jk} x^i_k x^i_k + \lambda \| \theta_{-j} \|_1 \right] \right\}
\]

- \( f(\theta; x) = \log \left\{ \exp \left( \sum_{k \in -j} \theta_{jk} x_k \right) + \exp \left( - \sum_{k \neq j} \theta_{jk} x_k \right) \right\} \)

- It turns out this is equivalent to solving \(p\) penalized logistic regression problems, which is rather easy (R-package g1mnnet)

\(^2\)Ravikumar et al (2010)

Other Non-Gaussian Distributions

- Assume a pairwise graphical model

\[
P(X) \propto \exp \left\{ \sum_{j \in V} \theta_j \phi_j(X_j) + \sum_{(j,k) \in E} \theta_{jk} \phi_{jk}(X_j, X_k) \right\}
\]

- Then, similar to the Ising model, graphical models can be learned for other members of the exponential family
  - Poisson graphical models (for e.g. RNAseq), Multinomial graphical models, etc
  - All of these can be learned using a neighborhood selection approach, using the g1mnnet package\(^3\)
  - We can even learn networks with multiple types of nodes (gene expression, SNPs, and CNVs)\(^4\)

\(^3\)Yang et al (2012)

A General Approach for Estimation of Graphical Models

- Consider $n$ iid observations from a $p$-dimensional random vector $x = (X_1, \ldots, X_p) \sim \mathcal{P}$
- Consider the (undirected) graph $G = (V, E)$ with vertices $V = \{1, \ldots, p\}$
- Want to estimate edges $E \subseteq V \times V$ that satisfy
  \[ p_j(X_j \mid \{X_k : k \neq j\}) = p_j(X_j \mid \{X_k : k \in N(j)\}) = p_j(X_j \mid \{X_k : (k, j) \in E\}) \]
- $N(j)$ is the minimal set of variables on which the conditional densities depend

Estimating Conditional Independencies

Question: how to condition?
Estimating Conditional Independencies

Question: how to condition?

- **Approach 1**: Estimate the joint density \( f(X_1, \ldots, X_p) \); then get the conditionals \( f_j(X_j \mid X_{-j}) \)
  - Efficient, coherent
  - Computationally challenging
  - Restrictive: how many joint distributions do you know?
  - Hard to check if assumptions hold!
Estimating Conditional Independencies

Question: how to condition?

- **Approach 1**: Estimate the joint density $f(X_1, \ldots, X_p)$; then get the conditionals $f_j(X_j | X_{-j})$
  - Efficient, coherent
  - Computationally challenging
  - Restrictive: how many joint distributions do you know?
  - Hard to check if assumptions hold!

- **Approach 2**: Estimate the conditionals directly $f_j(X_j | X_{-j})$
  - Computationally easy
  - Leads to easy & flexible models (regression)!
  - May not be efficient or coherent
A Semi-parametric Approach

- Consider additive non-linear relationships (additive model):

\[ X_j \mid X_{-j} = \sum_{k \neq j} f_{jk}(X_k) + \varepsilon \]
A Semi-parametric Approach

- Consider additive non-linear relationships (additive model):

\[ X_j \mid X_{-j} = \sum_{k \neq j} f_{jk}(X_k) + \varepsilon \]

- Then if \( f_{jk}(X_k) = f_{kj}(X_j) = 0 \), we conclude that \( X_j \) and \( X_k \) are conditionally independent, given the other variables.

- In other words, we assume that conditional distributions and conditional means depend on the same set of variables.
A Semi-parametric Approach

- Consider additive non-linear relationships (additive model):

$$X_j | X_{-j} = \sum_{k \neq j} f_{jk}(X_k) + \varepsilon$$

- Then if $f_{jk}(X_k) = f_{kj}(X_j) = 0$, we conclude that $X_j$ and $X_k$ are conditionally independent, given the other variables.

- In other words, we assume that conditional distributions and conditional means depend on the same set of variables.

- We then use a semi-parametric approach for estimating the conditional dependencies.

SpaCE JAM\textsuperscript{5}

- Sparse Conditional Estimation with Jointly Additive Models (SpaCE JAM)

$$\min_{f_k \in \mathbb{R}} \frac{1}{2n} \sum_{j=1}^{p} \|X_j - \sum_{k \neq j} f_{jk}(X_k)\|^2_2 + \lambda \sum_{k> j} \left( \|f_{jk}(X_k)\|^2_2 + \|f_{kj}(X_j)\|^2_2 \right)^{1/2}$$

- $f_{jk}(X_k) = \psi_{jk} \beta_{jk}$
- $\psi_{jk}$ is a $n \times r$ matrix of basis functions for $f_{jk}$
- $\beta_{jk}$ is an $r$-vector of coefficients
- The standardized group lasso penalty for functions $\|f_k\|_2$
- This is a convex problem, and block coordinate descent converges to the global minimum

\textsuperscript{5}Voorman et al (2014) Biometrika, R-package spacejam
Other Flexible Procedures

- **Forest density estimation** (Liu et al, 2011) assumes that the underlying graph is a forest, and estimates the bivariate densities non-parametrically.
- **Graphical random forests** (Fellinghauer et al, 2013) uses random forests to flexibly model conditional means.
  - Considers conditional dependencies through conditional mean.
  - Allows for general random variables, discrete or continuous.
  - Uses a random forest to estimate $E[X_j | X_{\setminus j}]$ non-parametrically.

Comparison on Simulated Data

non-linear relationships ($p = 100$, $n = 50$)

![Graphical Models for Non-Gaussian Distributions](image-url)
Comparison on Simulated Data

linear relationships \((p = 100, n = 50)\)

Estimation of Cell Signaling Network

Sachs et al (2005)
Summary - I

- Multivariate normality & linear conditional relationships are strong assumptions that may not hold in practice
Summary - I

- Multivariate normality & linear conditional relationships are strong assumptions that may not hold in practice
- Marginal transformations (and rank-based methods) also assume linear relationships in the transformed scale
- Estimation of graphical models for general non-Gaussian distributions is a difficult problem, and often requires additional assumptions (pairwise interactions, dependency via conditional means etc)
Summary - II

- Assuming pairwise interactions, graphical models for members of the exponential family can be estimated efficiently
  - This idea can also be extended to graphs with multiple node types, however, the pairwise graphical model becomes restrictive in that setting
Summary - II

- Assuming pairwise interactions, graphical models for members of the exponential family can be estimated efficiently
  - This idea can also be extended to graphs with multiple node types, however, the pairwise graphical model becomes restrictive in that setting
- Considering conditional means and additive models is a tractable alternative with good empirical and theoretical properties
  - GraFo uses random forests to solve this problem
  - SpaCE JAM applies a standardized group lasso penalty to enforce “symmetry” in terms of edge selection
Probability Distribution over DAGs

Conditional Independence in DAGs

Bayesian Networks

I Bayesian networks are a special class of graphical models defined on directed acyclic graphs.

I Directed acyclic graphs (DAGs) are defined as graphs that:
- only have directed edges, i.e., if $A_{ij} \neq 0$, $A_{ji} = 0$;
- there are no cycles in the network.

Bayesian networks are widely used to model causal relationships between variables.

Note that correlation ≠ causation!

Therefore, we (usually) cannot estimate Bayesian networks from (partial) correlations.
Bayesian Networks

- Bayesian networks are a special class of graphical models defined on directed acyclic graphs.
- Directed acyclic graphs (DAGs) are defined as graphs that:
  i) only have directed edges, i.e. if $A_{ij} \neq 0$, $A_{ji} = 0$;
  ii) there are no cycles in the network.
Bayesian Networks

- Bayesian networks are a special class of graphical models defined on directed acyclic graphs.
- Directed acyclic graphs (DAGs) are defined as graphs that:
  i) only have directed edges, i.e. if $A_{ij} \neq 0$, $A_{ji} = 0$;
  ii) there are no cycles in the network.
- Bayesian networks are widely used to model causal relationships between variables.

Note that correlation $\neq$ causation!
Bayesian Networks

- Bayesian networks are a special class of graphical models defined on directed acyclic graphs.
- Directed acyclic graphs (DAGs) are defined as graphs that:
  i) only have directed edges, i.e. if $A_{ij} \neq 0$, $A_{ji} = 0$;
  ii) there are no cycles in the network.
- Bayesian networks are widely used to model causal relationships between variables.
- Note that correlation $\neq$ causation!
- Therefore, we (usually) cannot estimate Bayesian networks from (partial) correlations

Why Bayesian Networks?
Why Bayesian Networks?
Many biological networks include directed edges:

- In gene regulatory networks, protein products of transcription factors can alter the expression of target genes, but the target genes (usually) don’t have a direct effect on the expression of transcription factors.
Why Bayesian Networks?

Many biological networks include directed edges:

- In cell signaling networks, the signal from the cell’s environment is transducted into the cell, and results e.g. in (global) changes in gene expression, but gene expression may not affect the environmental factors.

- Biochemical reactions in metabolic networks, may not be reversible, and in that case, one metabolite may affect the other, but the relationship is not reciprocated.
Why Bayesian Networks?

However, biological networks may not be DAGs:

- Gene regulatory networks, signaling networks and metabolic networks, may all contain feedback loops (positive/negative)

which make estimation even more difficult!
What's the Difference?

- Bayesian networks are widely used to model causal relationships between variables.
What's the Difference?

- Bayesian networks are widely used to model causal relationships between variables.
- Undirected networks (e.g. GGM) provide information about associations among variables; while this greatly helps in the study of biological systems, in some cases, they are not enough (e.g. drug development).

- The main difference is the direction of the edges; however, it turns out that there are also some differences in terms of structure/skeleton of the network (more on this later).
What’s the Difference?

- Bayesian networks are widely used to model causal relationships between variables.
- Undirected networks (e.g., GGM) provide information about associations among variables; while this greatly helps in the study of biological systems, in some cases, they are not enough (e.g., drug development).
- The main difference is the direction of the edges; however, it turns out that there are also some differences in terms of structure/skeleton of the network (more on this later).
- We can estimate undirected networks from observational data, i.e., steady-state gene expression data, but usually they are not enough for estimation of directed networks.
- Finally, estimating directed networks is (much) more difficult.

Why is estimation more difficult?

- Estimation of Bayesian networks requires estimating both the skeleton of the network (i.e., whether there is an edge between \( i \) and \( j \)) and also the direction of the edges.
Why is estimation more difficult?

- Estimation of Bayesian networks requires estimating both the skeleton of the network (i.e. whether there is an edge between $i$ and $j$) and also the direction of the edges.
- While estimation of skeleton is possible, direction of edges cannot be in general learned from observational data, no matter how many samples we have (this is referred to as observational equivalence). Consider this simple graph:

\[ X_1 \rightarrow X_2 \]

- Then, no matter what $n$ is, we cannot distinguish between $X_1 \rightarrow X_2$ and $X_2 \rightarrow X_1$, so basically what we see is:

\[ X_1 \rightarrow X_2 \]
Introduction
Probability Distribution over DAGs
Conditional Independence in DAGs

Outline

- Basics of Bayesian networks, including
Outline

- **Basics** of Bayesian networks, including
  - directed acyclic graphs (DAGs)
  - conditional independence in DAGs, d-separation, and moral graphs
  - probability distributions over DAGs
  - structural equation models (SEM)
  - additional topics (faithfulness, Markov equivalence, ...)

- **Estimation** of Bayesian networks from observational data
Outline

- **Basics** of Bayesian networks, including
  - directed acyclic graphs (DAGs)
  - conditional independence in DAGs, d-separation, and moral graphs
  - probability distributions over DAGs
  - structural equation models (SEM)
  - additional topics (faithfulness, Markov equivalence, ...)
- **Estimation** of Bayesian networks from observational data
- **Estimation** of Bayesian networks from perturbation and time-course data

Directed Graphs: Some Terminology

- **nodes** in directed networks represent random variables; we denote the set of nodes by $V$
- **edges** are directed, and represent causal relationships among variables; we denote the set of edges by $E$
Directed Graphs: Some Terminology

- **nodes** in directed networks represent random variables; we denote the set of nodes by $V$.
- **edges** are directed, and represent causal relationships among variables; we denote the set of edges by $E$.
- The **parents** of node $j$ are $\{ k : k \to j \}$, we denote this by $pa_j$ or $pa(j)$.
- The **children** of node $j$ are $\{ k : j \to k \}$.
Directed Graphs: Some Terminology

- nodes in directed networks represent random variables; we denote the set of nodes by $V$
- edges are directed, and represent causal relationships among variables; we denote the set of edges by $E$
- The parents of node $j$ are $\{k : k \rightarrow j\}$, we denote this by $pa_j$ or $\text{pa}(j)$
- The children of node $j$ are $\{k : j \rightarrow k\}$
- Two vertices connected by an edge are called adjacent
Directed Graphs: Some Terminology

- $\text{pa}(1) = \emptyset$, $\text{pa}(2) = 1$, $\text{pa}(3) = \text{pa}(4) = \{2\}$, $\text{pa}(5) = \{3, 4\}$

- What are children of $\{1, \ldots, 5\}$?
Directed Graphs: Some Terminology

- A path between two nodes $i$ and $j$ is a sequence of distinct adjacent nodes:
  - e.g. $i \leftarrow k_1 \rightarrow k_2 \rightarrow k_3 \leftarrow j$
  - In a DAG with $p$ nodes, there cannot be a path longer than $p - 1$ (why?)
  - There can be multiple paths between two nodes
Directed Graphs: Some Terminology

- A path between two nodes $i$ and $j$ is a sequence of distinct adjacent nodes:
  - e.g. $i \leftarrow k_1 \rightarrow k_2 \rightarrow k_3 \leftarrow j$
  - In a DAG with $p$ nodes, there cannot be a path longer than $p - 1$ (why?)
  - There can be multiple paths between two nodes
- $i$ is an ancestor of $j$ if there is a directed path of length $\geq 1$ from $i$ to $j$: $i \rightarrow \cdots \rightarrow j$ (or if $i = j$)
- If $i$ is an ancestor of $j$, then $j$ is said to be a descendant of $i$
Directed Graphs: Some Terminology

- What are paths between 1&4, 3&4, 2&6?

- What are ancestors of \{1, ..., 5\}?
Introduction
Probability Distribution over DAGs
Conditional Independence in DAGs

Directed Graphs: Some Terminology

1
2
3
4
5
6

- What are paths between 1&4, 3&4, 2&6?
- What are ancestors of \{1, \ldots, 5\}? 

An important concept in DAGs is that of colliders (aka “inverted forks”):
Directed Graphs: Some Terminology

An important concept in DAGs is that of **colliders** (aka “inverted forks”):

- \( k \) is a **collider on a path** between \( i \) and \( j \) if it is a not an end-point of the path, and the path is of the form

\[
i \rightarrow k \leftarrow \ldots j
\]

- \( k \) is an **non-collider** if it is not an end-point, and is not a collider on a path:
  - \( i \leftarrow k \leftarrow \ldots j \)
  - \( i \rightarrow k \rightarrow \ldots j \)
  - \( i \leftarrow k \rightarrow \ldots j \)
Directed Graphs: Some Terminology

An important concept in DAGs is that of colliders (aka “inverted forks”):

- *k* is a **collider on a path** between *i* and *j* if it is a not an end-point of the path, and the path is of the form

  \[ i \ldots \rightarrow k \leftarrow \ldots j \]

- *k* is an **non-collider** if it is not an end-point, and is not a collider on a path:
  - \[ i \ldots \leftarrow k \leftarrow \ldots j \]
  - \[ i \ldots \rightarrow k \rightarrow \ldots j \]
  - \[ i \ldots \leftarrow k \rightarrow \ldots j \]

- **Note:** colliders and non-colliders are defined w.r.t. paths; a collider in one path can be a non-collider in another!
Directed Graphs: Some Terminology

- What are the colliders on paths between 1&4, 3&4, 2&6?
- What are the non-colliders on paths between 1&4, 3&4, 2&6?
Factorization of Probability Distributions over DAGs

First, note that for any set of random variables, not necessarily on a DAG, we can write:

\[
P(X_1, X_2, X_3) = P(X_1 \mid X_2, X_3) P(X_2 \mid X_3) P(X_3) \\
= P(X_3 \mid X_1, X_2) P(X_2 \mid X_1) P(X_1) \\
= \ldots
\]
Factorization of Probability Distributions over DAGs

- First, note that for any set of random variables, not necessarily on a DAG, we can write:

\[
P(X_1, X_2, X_3) = P(X_1 \mid X_2, X_3) P(X_2 \mid X_3) P(X_3) \\
= P(X_3 \mid X_1, X_2) P(X_2 \mid X_1) P(X_1) \\
= \cdots
\]

- Now, consider this simple DAG

\[
\begin{array}{c}
\text{\(X_1\)} \\
\xrightarrow{\rho_{12}} \\
\text{\(X_2\)} \\
\xrightarrow{\rho_{23}} \\
\text{\(X_3\)}
\end{array}
\]

- Then, the probability distribution can be factorized as

\[
P(X_1, X_2, X_3) = P(X_3 \mid X_2) P(X_2 \mid X_1) P(X_1)
\]
Factorization of Probability Distributions over DAGs

- In general, for any set of random variables on a DAG $G = (V, E)$, and for any probability distribution $P$ (Markov relative to $G$) we have

$$P(V) = \prod_{j \in V} P(X_j | \text{pa}_j)$$

- Compare this with the general probability decomposition

$$P(V) = \prod_{j \in V} P(X_j | X_1, \ldots, X_{j-1})$$
Factorization of Probability Distributions over DAGs

- In general, for any set of random variables on a DAG $G = (V, E)$, and for any probability distribution $P$ (Markov relative to $G$) we have

$$P(V) = \prod_{j \in V} P(X_j | \text{pa}_j)$$

- Compare this with the general probability decomposition

$$P(V) = \prod_{j \in V} P(X_j | X_1, \ldots, X_{j-1})$$

- This means that on DAGs we have

$$P(X_j | X_1, \ldots, X_{j-1}) = P(X_j | \text{pa}_j)$$

- In other words, the probability distribution for each variable depends only on its parents
Independence (unconditional)

- Recall the following (equivalent) characterizations of independence, $X \perp Y$:
  - $P(X = x, Y = y) = P(X = x)P(Y = y)$
  - $P(X = x | Y = y) = P(X = x) \text{ (is symmetric)}$
Independence (unconditional)

- Recall the following (equivalent) characterizations of independence, $X \perp Y$:
  - $P(X = x, Y = y) = P(X = x)P(Y = y)$
  - $P(X = x | Y = y) = P(X = x)$ (is symmetric)
- Intuitively, if $X \perp Y$ then knowledge of $X$ provides no information about $Y$.

- These can be generalized for vectors.

If $X$ and $Y$ are jointly Gaussian $X \perp Y \iff \text{Corr}(X, Y) = 0$.

If $X$ and $Y$ are binary, $X \perp Y \iff \log \text{OR}(X, Y) = 0$. 

Recall the following (equivalent) characterizations of independence, $X \perp Y$:
Independence (unconditional)

- Recall the following (equivalent) characterizations of independence, $X \independent Y$:
  - $P(X = x, Y = y) = P(X = x)P(Y = y)$
  - $P(X = x | Y = y) = P(X = x)$ (is symmetric)
- Intuitively, if $X \independent Y$ then knowledge of $X$ provides no information about $Y$.
- These can be generalized for vectors.
- If $X$ and $Y$ are jointly Gaussian $X \independent Y$ iff $\text{Corr}(X, Y) = 0$.
- If $X$ and $Y$ are binary, $X \independent Y$ iff $\text{logOR}(X, Y) = 0$. 
Conditional Independence

- Conditional independence $X \perp Y \mid Z$ has similar characterizations:

  i) $P(X = x, Y = y \mid Z = z) = P(X = x \mid Z = z)P(Y = y \mid Z = z)$

  ii) $P(X = x \mid Y = y, Z = z) = P(X = x \mid Z = z)$ (is symmetric)
Conditional Independence

- Conditional independence $X \perp Y \mid Z$ has similar characterizations:
  i) $P(X = x, Y = y \mid Z = z) = P(X = x \mid Z = z)P(Y = y \mid Z = z)$
  ii) $P(X = x \mid Y = y, Z = z) = P(X = x \mid Z = z)$ (is symmetric)

- We also have,

$$P(X = x, Y = y, Z = z) = \frac{P(X = x, Z = z)P(Y = y, Z = z)}{P(Z = z)}.$$

Intuitively, if $X \perp Y$ then if $Z$ is known, knowledge of $X$ provides no information about $Y$.

These can be generalized for vectors.
Conditional Independence

- Conditional independence $X \perp Y \mid Z$ has similar characterizations:
  \[ P(X = x, Y = y \mid Z = z) = P(X = x \mid Z = z)P(Y = y \mid Z = z) \]
  \[ \text{ii) } P(X = x \mid Y = y, Z = z) = P(X = x \mid Z = z) \text{ (is symmetric)} \]
- We also have,
  \[ P(X = x, Y = y, Z = z) = \frac{P(X = x, Z = z)P(Y = y, Z = z)}{P(Z = z)}. \]
- Intuitively, if $X \perp Y$ then if $Z$ is known, knowledge of $X$ provides no information about $Y$.
- These can be generalized for vectors.

- If $X$ & $Y$ are binary, $X \perp Y \mid Z$ iff $\logOR(X, Y \mid Z) = 0$
  - This is the coefficient in logistic regression of (say) $Y$ on $X, Z$.  

© Ali Shojaie SISG: Pathway & Networks
Conditional Independence

- If $X$ & $Y$ are binary, $X \perp Y \mid Z$ iff $\log OR(X, Y \mid Z) = 0$
  - This is the coefficient in logistic regression of (say) $Y$ on $X, Z$.

- If $X$ & $Y$ are jointly Gaussian, $X \perp Y \mid Z$ iff $\text{Corr}(X, Y \mid Z) = 0$.
  - This is the coefficient in linear regression of (say) $Y$ on $X, Z$.

The Toy Example, Revisited

![Toy Example Diagram]
The Toy Example, Revisited

Recall that $P(X_1, X_2, X_3) = P(X_3|X_2)P(X_2|X_1)P(X_1)$

This implies that $X_3 \perp X_1 | X_2$ (by (i))
The Toy Example, Revisited

Recall that $P(X_1, X_2, X_3) = P(X_3|X_2)P(X_2|X_1)P(X_1)$

This implies that $X_3 \perp X_1 | X_2$ (by (i))

However, this is not always the case on DAGs!

How can we read conditional independence relations from the graph?

© Ali Shojaie  
SISG: Pathway & Networks  
21
The Toy Example, Revisited

Recall that \( P(X_1, X_2, X_3) = P(X_3 | X_2) P(X_2 | X_1) P(X_1) \)

This implies that \( X_3 \perp X_1 | X_2 \) (by (i))

However, this is not always the case on DAGs!

How can we read conditional independence relations from the graph?

We can do this using a concept called d-separation?

An example from genetics

Consider an example from population genetics:

We have genetic information for Mother, Father, Daughter and Son in form of dominant/recessive genotype (A/a) for a single gene

Then each individual can have one of three states: AA, aa, Aa
An example from genetics

Consider an example from population genetics:

Now, it is natural to assume that given the parents’ genetic information, the genotypes of Son and Daughter are independent \( S \perp D \mid \{M, F\} \)

Also, one can assume independence among genotypes of \( M \) and \( F \) \( \Rightarrow M \perp F \)

However, if we know that e.g. Son has Aa, and Mother has aa, then Father should have Aa or AA \( \Rightarrow M \perp F \mid S \)
A path $\pi$ is said to be d-separated (or blocked) by a set of nodes $Z$, iff

1. $\pi$ includes a chain $i \to m \to j$ or a fork $i \leftarrow m \to j$ such that the middle note is in $Z$, or
2. $\pi$ contains a collider (or inverted fork) $i \to m \leftarrow j$ such that neither the middle node $m$ nor its descendants are NOT in $Z$.

How is this used?
d-separation

A path \( \pi \) is said to be d-separated (or blocked) by a set of nodes \( Z \), iff

1. \( \pi \) includes a chain \( i \to m \to j \) or a fork \( i \leftarrow m \to j \) such that
   the middle note is in \( Z \), or
2. \( \pi \) contains a collider (or inverted fork) \( i \to m \leftarrow j \) such that
   neither the middle node \( m \) nor its descendants are NOT in \( Z \).

How is this used?

- If \( i \) and \( j \) are d-separated given \( Z \), then \( X_i \perp X_j | Z \) for any
  probability distribution \( P \) factorizing according to \( G \)
- If \( i \) and \( j \) are d-separated given \( \emptyset \), then \( X_i \perp X_j \) for any
  probability distribution \( P \) factorizing according to \( G \)
Genetics example, revisited

Consider an example from population genetics:

\[
\begin{align*}
M & \rightarrow S \\
F & \rightarrow S \\
M & \rightarrow D \\
F & \rightarrow D
\end{align*}
\]

\[
\begin{align*}
\{M, F\} & \text{ block all paths from } S \text{ to } D \Rightarrow D \perp S \mid \{M, F\}
\end{align*}
\]
Genetics example, revisited

Consider an example from population genetics:

- $\{M, F\}$ block all paths from $S$ to $D$ \Rightarrow $D \perp S \mid \{M, F\}$
- Is $M \perp F$?
Moral Graphs

- Reading conditional independence relations from DAGs can be difficult
- An alternative approach is to use a modified version of the network, called the moral graph of DAG
- To get the moral graph $\tilde{G}$ of $G$
  - join (“marry”) common parents of each node
  - remove all the directions
- Then, $X_i \perp X_j | Z$ iff $Z$ separates $i$ and $j$ in $\tilde{G}$

Genetics example, revisited (again)

Consider an example from population genetics:

```
M \rightarrow S \rightarrow F \rightarrow D
M \rightarrow F \rightarrow S \rightarrow D
M \rightarrow S \rightarrow D
M \rightarrow F \rightarrow S
```

Consider an example from population genetics:
Genetics example, revisited (again)

Consider an example from population genetics:

\[ \begin{align*}
\text{M} & \quad \text{F} \\
\downarrow & \\
\text{S} & \quad \text{D}
\end{align*} \]

- Is \( S \perp D \mid \{M, F\} \)?
- Is \( M \perp F \)?
- Is \( M \perp F \mid \{S, D\}, \mid S, \mid D \)?
Genetics example, revisited (again)

Consider an example from population genetics:

\[ M \rightarrow F \]
\[ S \rightarrow D \]

- Is \( S \perp D \mid \{M, F\} \)?
- Is \( M \perp F \)?
- Is \( M \perp F \mid \{S, D\}, \mid S, \mid D \)?

A More Complex Example

What are conditional independence relations in this graph?
A More Complex Example

What are conditional independence relations in this graph?

1  2  3  4  5  6

Structural Equation Models
Structural Equation Models

- A popular way to represent causal relationships on DAGs is via structural equation models

\[ X_j = f_j(p_{aj}, \gamma_j), \quad j = 1, \ldots, p \]

- \( f_j \) can be in general any function relating \( j \) to its parents

\[ X_j = \alpha_j + \sum_{i \in pa_j} \gamma_{ij} X_i \]
Structural Equation Models

- A popular way to represent causal relationships on DAGs is via structural equation models

\[ X_j = f_j(pa_j, \gamma_j), \quad j = 1, \ldots, p \]

- \( f_j \) can be in general any function relating \( j \) to its parents
- \( \gamma_j \)'s represent the independent component of \( j \)th variable (i.e. the part that doesn't depend on \( pa_j \))
- For Gaussian random variables, \( f_i \) is linear

\[ X_j = \sum_{j' \in pa_j} \rho_{jj'} X_{j'} + \gamma_j, \quad j = 1, \ldots, p \]
Structural Equation Models

- A popular way to represent causal relationships on DAGs is via structural equation models

\[ X_j = f_j(pa_j, \gamma_j), \quad j = 1, \ldots, p \]

- \( f_j \) can be in general any function relating \( j \) to its parents
- \( \gamma_j \)'s represent the independent component of \( j \)th variable (i.e. the part that doesn't depend on \( pa_j \))
- For Gaussian random variables, \( f_j \) is linear

\[ X_j = \sum_{j' \in pa_j} \rho_{jj'} X_{j'} + \gamma_j, \quad j = 1, \ldots, p \]

- here, \( \rho_{jj'} \) denotes the magnitude of effect of \( j' \) on \( j \), or their partial correlation

A Toy Example

![Toy Example Diagram](image_url)
A Toy Example

Assuming normality we can write:

\[
\begin{align*}
X_1 & = \gamma_1 \\
X_2 & = \rho_{12}X_1 + \gamma_2 = \rho_{12}\gamma_1 + \gamma_2 \\
X_3 & = \rho_{23}X_2 + \gamma_3 = \rho_{23}\rho_{12}\gamma_1 + \rho_{23}\gamma_2 + \gamma_3 \\
\end{align*}
\]

For non-Gaussian variables, these equations will involve non-linear relationships.
Estimation of DAGs in Biological Settings

- Estimation of DAGs is (in general) computationally very hard (in fact, it’s NP-hard): there are \( \sim 2^p \) DAGs with \( p \) nodes!
- Three different types of biological data can be used for estimation of directed graphs:
  i) observational data: steady-state data, or data comparing normal & cancer cells
  ii) time-course data: time-course gene expression data
  iii) perturbation data: data from knockouts experiments
- This lecture, we will cover (i), next lecture we will cover (ii) and (iii)
Estimation of DAGs from Observational Data

Algorithms for estimation of DAGs can be broadly categorized into two groups:

- **constraint-based** methods
  - often based on tests for CI & provide theoretical guarantees
  - PC algorithm, Grow-Shrink
- **score & search** methods
  - They assign a “score” to each estimated graph (e.g. based on likelihood, Bayes factor, AIC etc)
  - Then do a (greedy) search to find the best scoring graph
  - Hill Climbing algorithm
- **“hybrid”** methods
  - Usually first find the Markov blanket (e.g. the moral graph)
  - Then perform a search in a restricted space
  - Max-Min Hill Climbing algorithm

Constraint-Based Methods

- Need a conditional independence test (to test if $X \perp Y \mid Z$)
  - For Gaussian data, we can use partial correlation (or the Fisher’s Z-transformation of it)
  - For Binary data, we can use logOR
  - In general, we can use conditional mutual information
- The idea is to see if there exists a set $S$, for each pair of nodes $j, j'$, such that $X_j \perp X_{j'} \mid S$
  - $S$ can have 0 to $p-2$ members! usually stop at some $k \ll p$
  - I.e., for each pair of variables (all $\binom{p}{2}$ of them), we need to look at all possible subsets of remaining variables!!
- Recall that conditional independence is symmetric $\Rightarrow$ undirected graph!!
- So, these methods find the structure/skeleton of the DAG (will talk about direction later)
PC Algorithm (Spirtes et al, 1993)

- One of the first algorithms for learning structure of DAGs
- Efficient implementations that allow for learning DAG structures with \( p \) up to \( \sim 1000 \)
  - \( \text{R-package } \text{pcafg} \) (Kalisch & Buhlmann, 2007)
- The algorithm starts with a complete graph (i.e. a fully connected graph)
- Then for each pair of nodes \( j, j' \) it finds a separating set, \( S \) such that \( X_j \perp \!\!\!\perp X_{j'} \mid S \)
- If a set is found, then remove the edge, otherwise, \( j - j' \)

Start with a complete undirected graph, and set \( i = 0 \)

Repeat

- For each \( j \in V \)
- For each \( j' \in \text{ne}(j) \)
- Determine if \( \exists S \subset \text{ne}(j) \setminus \{j'\} \text{ with } |S| = i \)
  - Test for CI: is \( X_j \perp \!\!\!\perp X_{j'} \mid S \)?
  - If such an \( S \) exists, then set \( S_{jj'} = S \), remove \( j - j' \) edge
- \( i = i + 1 \)

Until \( |\text{ne}(j)| < i \) for all \( j \)
Example

\[ i = 0 \quad S_{1,2} = \emptyset \]
\[ S_{1,4} = \emptyset \]

\[ i = 1 \quad S_{3,4} = \{2\} \]

\[ i = 2 \quad S_{1,5} = \{3, 4\} \]
\[ S_{2,5} = \{3, 4\} \]

\[ i = 3 \quad \text{STOP (|\text{ne}_j| < 3 \ \forall j)} \]
Analysis of Protein Flow Cytometry using \texttt{pcalg}

\begin{verbatim}
> dat <- read.table('sachs.data')
> p <- ncol(dat)
> n <- nrow(dat)
## define independence test (partial correlations)
> indepTest <- gaussCItest
## define sufficient statistics
> suffStat <- list(C=cor(dat), n=n)
## estimate CPDAG
> pc.fit <- pc(suffStat, indepTest, p, alpha=0.1, verbose=FALSE)
> plot(pc.fit, main='PC Algorithm')
\end{verbatim}

- Need to determine the type of CI test (\texttt{indepTest}), and sufficient statistics (\texttt{suffStat})
- Also need to choose \(\alpha\) (\texttt{alpha}), the probability of false positive for selecting edges.
  - Larger values of \(\alpha\) allow more edges (not adjusted for multiple comparisons)
  - The algorithm works faster when \(\alpha\) is small

\begin{center}
\textbf{PC Algorithm}
\end{center}

\begin{itemize}
\item But wait, where did the \textit{directions} come from? And why are only some of the edges directed?
\end{itemize}
Markov Equivalence

Consider the following 4 graphs

Which graphs satisfy $X_1 \perp\!
\perp X_3 \mid X_2$?
Markov Equivalence

Consider the following 4 graphs

In the first 3 graphs, $X_1 \perp X_3 \mid X_2$?
Two graphs that imply the same CI relationships via d-separation are called Markov equivalent
Representation of Markov Equivalence

- Markov equivalent graphs correspond to the same probability distribution and cannot be distinguished from each other based on observations!
- Therefore, the direction of edges that correspond to Markov equivalent graphs cannot be determined
- We show these edges using undirected edges in the graph
- The resulting graph is a CPDAG (completed partially directed acyclic graph), and is really the best we can do!

CPDAGs
Finding Partial Directions in DAGs

- Partial directions in DAGs can be determined from unmarried colliders:
  - For each unmarried collider $i - k - j$
  - If $k \notin S_{ij}$, orient $i - k - j$ as $i \rightarrow k \leftarrow j$
- In addition to the above rule
  - Orient each remaining unmarried collider $i \rightarrow k - j$ as $i \rightarrow k \rightarrow j$
  - If $i \rightarrow k \rightarrow j$ and $i - j$ then orient as $i \rightarrow j$
  - If $i - m - j$ and $i \rightarrow k \leftarrow j$ are unmarried colliders and $m - k$, then orient as $m \rightarrow k$
Example

\[ i = 0 \quad S_{1,2} = \emptyset \]
\[ S_{1,4} = \emptyset \]
\[ i = 1 \quad S_{3,4} = \{2\} \]
\[ i = 2 \quad S_{1,5} = \{3, 4\} \]
\[ S_{2,5} = \{3, 4\} \]

The \textit{bnlearn} package

- There are a number of R-packages for learning the structure of DAGs, including \texttt{pclag}, \texttt{bnlearn}, \texttt{deal}
- \texttt{bnlearn} implements a number of estimation methods, both constraint-based and search-based:
  - constraint-based:
    - Grow-Shrink (GS);
    - Incremental Association Markov Blanket (IAMB);
    - Fast Incremental Association (Fast-IAMB);
    - Interleaved Incremental Association (Inter-IAMB);
  - the following score-based structure learning algorithms:
    - Hill Climbing (HC);
    - Tabu Search (Tabu);
  - the following hybrid structure learning algorithms:
    - Max-Min Hill Climbing (MMHC);
    - General 2-Phase Restricted Maximization (RSMAX2);
Analysis of Protein Flow Cytometry using `bnlearn`

```r
> dag1 <- gs(dat, alpha=0.01)   # GS method
> dag2 <- hc(dat2)              # Hill-Climbing search
>
> par(mfrow=c(1,2))
> plot(dag1)
> plot(dag2)
>
> compare(dag1, dag2)           # compare the two DAGs
```

- For GS need to choose $\alpha$ (alpha), the false positive probability for selecting edges
- `gs` (and other structure-based methods) find a PCDAG
- `hc` gives a directed graph (with highest score)
  - A number of criteria for choosing the "best" graph are implemented
  - To "search" the space either a new edge is added, or a current edge is removed, or reversed (if no cycles)

---

Analysis of Protein Flow Cytometry using `bnlearn`

```r
> dag1

Bayesian network learned via Constraint-based methods

model:
[partially directed graph]

nodes: 11
arcs: 26
  undirected arcs: 3
  directed arcs: 23
average markov blanket size: 6.00
average neighbourhood size: 4.73
average branching factor: 2.09

learning algorithm: Grow-Shrink
conditional independence test: Pearson’s Linear Correlation
alpha threshold: 0.01
tests used in the learning procedure: 2029
optimized: TRUE
```
Analysis of Protein Flow Cytometry using bnlearn

> dag2
Bayesian network learned via Score-based methods

model:

nodes: 11
arcs: 35
undirected arcs: 0
directed arcs: 35
average markov blanket size: 8.00
average neighbourhood size: 6.36
average branching factor: 3.18

learning algorithm: Hill-Climbing
score: Bayesian Information Criterion (Gaussian)
penalization coefficient: 4.459057
tests used in the learning procedure: 505
optimized: TRUE

The two graphs are quite different

> compare(dag1,dag3)
$tp
[1] 9
$fp
[1] 26
$fn
[1] 17
The estimated graphs are quite different

- The constrained-based methods seem to have more similarities (at least in terms of structure)
- The estimate from HC has more edges; we can change e.g. the score, but cannot directly control the sparsity
Penalized Likelihood Estimation of DAGs

- Recall that structural equation models can be used to represent causal relationships (and probability distributions) on DAGs

\[ X_i = f_i(pa_i, \gamma_i), \quad i = 1, \ldots, p \]

- And, for Gaussian random variables, we can write

\[ X_i = \sum_{j \in pa_i} \rho_{ji} X_j + \gamma_i, \quad i = 1, \ldots, p \]
Penalized Likelihood Estimation of DAGs

\[ X_1 = \gamma_1 \]
\[ X_2 = \rho_{12} X_1 + \gamma_2 = \rho_{12} \gamma_1 + \gamma_2 \]
\[ X_3 = \rho_{23} X_2 + \gamma_3 = \rho_{23} \rho_{12} \gamma_1 + \rho_{23} \gamma_2 + \gamma_3 \]
Penalized Likelihood Estimation of DAGs

\[ X_1 = \gamma_1 \]
\[ X_2 = \rho_{12} X_1 + \gamma_2 = \rho_{12} \gamma_1 + \gamma_2 \]
\[ X_3 = \rho_{23} X_2 + \gamma_3 = \rho_{23} \rho_{12} \gamma_1 + \rho_{23} \gamma_2 + \gamma_3 \]

Thus \( X = \Lambda \gamma \) where

\[
\Lambda = \begin{pmatrix}
1 & 0 & 0 \\
\rho_{12} & 1 & 0 \\
\rho_{12} \rho_{23} & \rho_{23} & 1
\end{pmatrix}
\]

\[\text{1Shojaie & Michailidis (2010)}\]
Penalized Likelihood Estimation of DAGs

- It turns out that $\Lambda = (I - A)^{-1}$, where $A$ is the weighted adjacency matrix of the DAG\(^1\)
- Thus, for Gaussian random variables, if we know the ordering of the variables (which is a BIG assumption!)
  after some math...
  we can estimate the adjacency matrix of DAGs, by minimizing the log-likelihood as a function of $A$:
  $$\hat{A} = \arg \min_{A \in \mathcal{A}} \{ \text{tr}[(I - A)^T(I - A)S] \}$$

\(^1\)Shojaie & Michailidis (2010)

Penalized Likelihood Estimation of DAGs

- In high dimensions, we can solve a penalized version of this problem, e.g. by adding a lasso penalty $\lambda \sum_{i<j} |A_{ij}|$
- It turns out that, the problem can be reformulated as $(p - 1)$ lasso problems, where we regress each variable, on those appearing earlier in the ordering:
  $$\hat{A}_{k,1:k-1} = \arg \min_{\theta \in \mathbb{R}^{k-1}} \left\{ n^{-1} \|X_{1:k-1}\theta - X_{k}\|_2^2 + \lambda \sum_{j=1}^{k-1} |\theta_j| w_j \right\}$$
  - As in glasso, $\lambda$ is a tuning parameter that controls the amount of sparsity; $\lambda = \frac{2}{\sqrt{n}Z_\alpha/(2p^2)}$ controls a false positive probability at level $\alpha$
Computational Complexity

- Compared to pcalg, this method runs much faster: $\sim np^2$ operations vs $\sim p^q$ ($q$ is the max degree)
- Can be easily implemented in R as $p - 1$ regressions using glmnet. A more general version is available in the spacejam package, which also includes estimation for non-Gaussian data.
Simulations

- Settings:
  \( p = 50, 100, 200 \)
  \( n = 100 \)
  Total number of edges in the network = \( n \)
  100 repetitions

- Performance Criteria
  1. Matthew’s Correlation Coefficient (MCC): ranges between \(-1\) (worst fit) and 1 (best fit), similar to \( F_1 \)
  2. Structural Hamming Distance (SHD): sum of false positive and false negatives
  3. True positive and false positive rates

- Tuning parameter for both PC-Algorithm and penalized likelihood method based on false positive error \( \alpha \)

Gaussian Observations

- pcalg
- lasso
- Alasso, \( \gamma = 1 \)

![](chart.png)
Gaussian Observations

Random Ordering of Variables
Regulatory Network of E-Coli

- Regulatory network of E-coli with \( p = 49 \) genes (7 TFs)
- Want to identify regulatory interactions among TFs and regulated genes
Summary

- Estimation of DAGs from observational data is both conceptually and computationally difficult
- Constraint-based and search-based algorithms become slow in high dimensions
- Also, may not be able to distinguish DAGs from observational data (Markov equivalence)
- Efficient penalized likelihood methods can estimate DAGs if the ordering is known
- Efficient implementations in R available for most methods
- Different methods need different tuning parameters...
Pathway & Network Analysis of Omics Data: 
Reconstructing Regulatory Networks 
from Time-Course & Perturbation Data

Ali Shojaie
Department of Biostatistics
University of Washington
faculty.washington.edu/ashojaie

Summer Institute for Statistical Genetics – 2018

MAPK/ERK Pathway
Estimation of Gene Regulator Networks

- Using steady-state gene expression data:
  - undirected association graphs: Graphical lasso (glasso), ARACNE, ...
  - DAGs or CPDAGs: PC-Algorithm, ...

- Using time-course gene expression data
  - Dynamic Bayesian networks
  - Granger causality

- Using perturbation screens, obtained by “perturbing” the biological system, often in the form of knockout or knockdown experiments, where in each experiment one or more genes are perturbed.
  - Model-based approaches: Nested Effect Models (NEM), methods of causal inference
  - Heuristic approaches: e.g. Pinna et al (2010),

Gene Regulatory Networks
Consider a simple regulatory network, with two transcription factors and one gene:
Gene Regulatory Networks

Consider a simple regulatory network, with two transcription factors and one gene:

\[
\begin{align*}
  g_1 & : \text{Inducer} \\
  g_2 & : \text{Inhibitor} \\
  g_3 & : \text{Regulated Gene}
\end{align*}
\]

The temporal expressions patterns of \( g_1 \), \( g_2 \) and \( g_3 \) may look like:
Gene Regulatory Networks

The temporal expressions patterns of $g_1$, $g_2$ and $g_3$ may look like:

Temporal patterns in Gene Regulatory Networks

- $g_1$: Inducer
- $g_2$: Inhibitor
- $g_3$: Regulated Gene
Temporal patterns in Gene Regulatory Networks

- $g_1$: Inducer
- $g_2$: Inhibitor
- $g_3$: Regulated Gene

Goal: discover the regulatory interactions from time-course gene expression data.
Temporal patterns in Gene Regulatory Networks

- \( g_1 \): Inducer
- \( g_2 \): Inhibitor
- \( g_3 \): Regulated Gene

Estimation of Gene Regulatory Networks from Time-Course Data

- The goal is to discover interactions among genes from time-course data, i.e., patterns of expressions over time.
- A suitable framework for inferring such mechanisms is Granger causality:
  - the idea is to see if changes in expression of gene \( X \) are predictive of those in \( Y \)
  - this model is closely related to the Dynamic Bayesian Networks (DBNs)
  - can handle self-regulatory effects and feedback loops
Granger Causality

We say $X$ is Granger-causal for $Y$ if $X_t = 0.7X_t + 0.3Y_t + 0.2Y_{t-2} + c$.
Granger Causality

We say $X$ is Granger-causal for $Y$ if:

$$X_t = \mathbf{0} \rightarrow Y_t = \mathbf{0}.$$
Granger Causality

We say $X$ is Granger-causal for $Y$. 

$X_t = 0.7 Y_{t-1} + 0.4 X_{t-1} + 0.2 X_{t-2} + c_t$

$Y_t = 0.8 Y_{t-1} + 0.2 Y_{t-2}$
Granger Causality

We say \( X \) is Granger-causal for \( Y \)

\[
Y_t = 0.7Y_{t-1} + 0.4X_{t-1} + 0.2X_{t-2} + \varepsilon_t
\]

- A time series \( X \) is said to be Granger-causal for \( Y \) if past values of \( X \) provide statistically significant information about future values of \( Y \)
- This is traditionally checked using a series of \( F \)-tests, on lagged values of \( X \)
- Granger causality \( \neq \) causality: Granger causality is about prediction and does not imply true causal effects
- Recent work extends this framework beyond Gaussian variables
- We focus on extension of this idea to high dimensional settings, which we refer to as Network Granger Causality
Network Granger Causality: Illustration

\( p \) variables observed over \( T \) time points

- \( X_1 \)
- \( X_2 \)
- \( X_3 \)

\( T-3 \) \( T-2 \) \( T-1 \) \( T \)

\( d = 2 \)
Network Granger Causality: VARs

- $X_1, \ldots, X_p$ time series for $p$ variables
- $X^t = (X^t_1, \ldots, X^t_p)'$ are realizations at time $t$
Network Granger Causality: VARs

- $X_1, \ldots, X_p$ time series for $p$ variables
- $X^t = (X^t_1, \ldots, X^t_p)'$ are realizations at time $t$
- VAR model for NGC:

$$X^T = A_1^1 X^{T-1} + \cdots + A_d^d X^{T-d} + \varepsilon^T$$

$A_{11}$: Autoregressive effect of $X_1$ on itself
Network Granger Causality: VARs

- $X_1, \ldots, X_p$ time series for $p$ variables
- $X^t = (X^t_1, \ldots, X^t_p)'$ are realizations at time $t$
- VAR model for NGC:
  \[ X^T = A_1^1 X^{T-1} + \cdots + A^d X^{T-d} + \epsilon^T \]

$A_{12}$: Autoregressive effect of $X_2$ on $X_1$

- $X_j$ Granger-causal for $X_i$ if $A^k_{i,j} \neq 0$ for some $k$ ($k = 1, \ldots, d$)
NGC Estimation

- With $d$ known, this is really a multivariate regression problem!
NGC Estimation

- With $d$ known, this is really a multivariate regression problem!
- When $p$ is large, we need some form of regularization

\[
\hat{A}^1 : \hat{A}^d = \arg \min_{A^1, \ldots, A^d} \left\| \sum_{t=0}^{T_0} \left( X^{T-t} - \sum_{k=1}^{d} A^k X^{T-t-k} \right) \right\|_2^2 + \lambda \sum_{k=1}^{d} \|A^k\|_1
\]
NGC Estimation

- With $d$ known, this is really a multivariate regression problem!
- When $p$ is large, we need some form of regularization
  - Can use e.g. the LASSO (Tibshirani, 1996)

$$\hat{A}^1 : \hat{A}^d = \arg\min_{A^1, \ldots, A^d} \left\{ \sum_{t=0}^{T_0} \left( X^{T-t} - \sum_{k=1}^{d} A^k X^{T-t-k} \right) \right\}_2^2 + \lambda \sum_{k=1}^{d} \|A^k\|_1$$

- The problem can be solved as $p$ separate lasso regressions
- The LASSO estimate is sparse: many entries of $\hat{A}^k$s exactly 0
- $\lambda$ is the tuning parameter controlling the degree of regularization: the larger the lambda, the sparser the estimates

- Often $d \ll T$, but unknown!
NGC Estimation: Unknown $d$

**Idea:** Use weighted penalties to simultaneously estimate $d$ and $A^k$

$$\hat{A}^1 : \hat{A}^d = \arg \min_{A^1, \ldots, A^d, d} \left\| \sum_{t=0}^{T_0} \left( X^{T-t} - \sum_{k=1}^d A^k X^{T-t-k} \right) \right\|^2 + \lambda \sum_{k=1}^d \psi^k(A^k) \left\| A^k \right\|_1$$

The weights $\psi^k(A^k)$ are functions of $A^k$: $\psi^k(A^k) = \infty \Rightarrow \hat{A}^k = 0$
NGC Estimation: Unknown $d$

Idea: Use weighted penalties to simultaneously estimate $d$ and $A^k$

$$\hat{A}^1 : \hat{A}^d = \arg \min_{A^1,...,A^d,d} \left\{ \sum_{t=0}^{T_0} \left( X^{T-t} - \sum_{k=1}^d A^k X^{T-t-k} \right)^2 + \lambda \sum_{k=1}^d \psi^k(A^k) \|A^k\|_1 \right\}$$

The weights $\psi^k(A^k)$ are functions of $A^k$: $\psi^k(A^k) = \infty \Rightarrow \hat{A}^k = 0$

1. Truncating LASSO (S. & Michailidis, 2010): assumes that if $A^k = 0$ for some $k$, then $A^k' = 0$ for all $k' > k$

2. Thresholded LASSO penalty (S., Basu & Michailidis, 2012): no structural assumption — use thresholding to decide if $A^k = 0$
NGC for Grouped Variables

- Incorporating inherent group structure into NGC estimation (e.g. genetic pathways, financial sectors)
- Variables partitioned into $M$ non-overlapping groups $G_1, \ldots, G_M$
NGC for Grouped Variables

- Incorporating inherent group structure into NGC estimation (e.g. genetic pathways, financial sectors)
- Variables partitioned into $M$ non-overlapping groups $G_1, \ldots, G_M$
- Idea: variables in $m$th group have similar effects on other variables, but groups can be misspecified, or heterogenous

An Illustration: Truncating Lasso Penalty

True

lasso

Alasso

Tlasso

TAlasso
Example I: Gene Network of HeLa Cells

9 genes, 47 time points
\( d = 3 \)

Example II: Gene Regulatory Networks of Yeast

5 Transcription Factors, 37 genes (\( p = 42 \)), 8 time points
\( d = 2 \)
Non-decaying Granger-causal effects

\[ X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow T\]

\[ X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow T-1 \]

\[ X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow T-2 \]

\[ X_1 \rightarrow X_2 \rightarrow X_3 \rightarrow T-3 \]
Regulatory Network of T-Cell Activation

- Data from Rangel et al (2004) on activation of T-cells
- $p = 58$ genes, $n = 44$ samples, and $T = 10$ time points
- Goal is to estimate the regulatory interactions

Adjacency Matrices of Estimated Networks
Adjacency Matrices of Estimated Networks

- Alasso
- Talasso
- Thlasso
Estimated Regulatory Networks

<table>
<thead>
<tr>
<th></th>
<th>Allasso</th>
<th>TAllasso</th>
<th>Thlasso</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allasso</td>
<td>(96)</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>TAllasso</td>
<td>99</td>
<td>(101)</td>
<td>–</td>
</tr>
<tr>
<td>Thlasso</td>
<td>35</td>
<td>102</td>
<td>(79)</td>
</tr>
</tbody>
</table>
R Implementation

Library ngc currently on github @ https://github.com/shojaie/ngc

gc(X, #input array dim=(n,p,T) (longitudinal), or (p,T) (time series); last time d = NULL, #number of time lags to consider method = 'regular', #method to use ("regular", "truncate", "threshold") group = NULL, #vector of group indices of length p or p*d; no groups if null groupByTime = FALSE, #whether to group lags of each variable over time typeIerr = NULL, #acceptable type I error rate for selecting lambda typeIIerr = 0.1, #acceptable type II error rate weights = NULL, #weights for Adaptive lasso (use regular lasso if NULL). thresholdConstant = NULL, #constant used for calculating threshold value refit = FALSE, #whether to refit a linear regression after initial thresholding covNames = NULL #covariate names)

Simple use: ngc(X)
R Implementation

```
# Fit with CV
fitA <- ngc(X, q = 6, method = "regular", covRakes = covRakes)
plot(fitA)

# Granger causality plot
plot(fitA, nge.type = "granger")
```

Comments

- **Benefits:**
  - The optimization problem is convex, and can be solved efficiently.
  - Does not require structural assumptions (**no decay assumption**)

- **Drawbacks:**
  - Requires more tuning parameters
  - Can be less efficient than truncating lasso **if the decay assumption holds**

- The tuning parameters can be chosen so that the method has desirable performance
- Penalized methods implemented in the R package `ngc`
Data from Perturbation Screens

- Steady-state data are easy to obtain, but only represent association among genes and hence have insufficient informational content.
- Perturbation data provide direct evidence on causal directions, but are expensive to obtain. This becomes more complicated if perturbing a particular gene is lethal.
- Data is obtained by knockout or knockdown experiments on one or more genes at a time. The data then measures the effect of the experiments on other genes in the network.
Data from Perturbation Screens

- In practice, due to limited sample size, the perturbation data are often *discretized*: genes are categorized as up/down regulated or active/inactive.
Data from Perturbation Screens

- In practice, due to limited sample size, the perturbation data are often **discretized**: genes are categorized as up/down regulated or active/inactive.

- The **discretized** perturbation data
  1. do not provide enough information to construct the structure of regulatory networks.
  2. provide enough information to determine causal (topological) ordering(s) of nodes.
Methods for Estimation of Regulatory Networks from Perturbation Data

- **Nested Effect Model (NEM):** defines a probability distribution for perturbed (knockout) genes, and estimates the networks using a Bayesian framework
- **Heuristic approaches:** start with the network of significant effects of genes on all other genes (based on the perturbation data) and try to trim this network using features of observed networks
- **Causal inference methods:** in particular, using the intervention calculus (Pearl, 2000) which describes the joint probability distribution of random variables in the setting of experiments

Nested Effect Models

- Motivated by RNAi experiments: few knocked-out genes (called \( S \) genes), and many affected genes (called \( E \) genes)

- Assumes that each \( S \) gene affect few \( E \) genes
- More importantly, assumes that each \( E \) genes is only affected by one \( S \) gene
- The network of \( S \) gens is arbitrary, but there is no association among \( E \) genes (condition on \( S \) genes)
- Considers the setting where \( S \) genes are (potentially) not observable, but \( E \) genes are observed
- The goal is to learn the relationship among \( S \) genes, based on the patterns of \( E \) genes, which is a difficult problem!
Nested Effect Models

- Works with **discretized data**: there is either an effect (1) from knocking out of $S_i$ on $E_j$ or not (0)
- Assumes there are **positive and negative control samples**
- Allows for presence of false positives and false negatives in the discretized data

In the simplest form (a) a chain with 3 nodes is assumed, and the model tries to learn the relationship between $S$ genes based on the $E$ genes that are affected by each perturbation (b)

- The matrix $\Phi$ is the **influence matrix** discussed before
- To simplify computation, the task of structure learning is broken down into **triplets** of $S$ genes
Nested Effect Models

- Reconstruction of network of $S$ genes is performed by first clustering the $E$ genes into groups with similar patterns.
- It is then decided whether a cluster is up-stream or down-stream the other one based on the patterns of effects (subset relationships).

```
> library(nem)
> data("BoutrosRNAi2002")
> disc <- nem.discretize(D=BoutrosRNAiExpression,neg=1:4,pos=5:8)
> res <- nem(D=disc$dat,para=disc$para,inference="search")
```

D data matrix with experiments in the columns (binary or continuous).

- R package nem implements the original NEM model, as well as some of its extensions.
- The package works well for up to $\sim 100$ $S$ genes (though very slow), but may not work for larger experiments.
The RIPE Algorithm

RIPE integrates two sources of data, from perturbation screens and steady-state expression profiles, to give better estimates of regulatory networks.

I) Use perturbation data to determine causal ordering(s) among nodes

II) For each ordering from step (I), use steady-state gene expression data to estimate the structure of the graph

III) Use model averaging to construct a consensus graph

1Regulatory Network Inference from joint Perturbation and Expression data (Shojaie et al, 2014), package ripe on github
Step I) Determining Causal Orderings

- First, obtain the influence graph $P$ from the perturbation data (this can be done many different ways: p-value cutoff/fold-change cutoff etc).

In absence of noise, the influence graph is obtained from the original graph by connecting node $i$ to $j$ if there is a directed path from $i$ to $j$. 
Step I) Determining Causal Orderings

- First, obtain the influence graph $P$ from the perturbation data (this can be done many different ways: p-value cutoff/fold-change cutoff etc)
- In absence of noise, the influence graph is obtained from the original graph by connecting node $i$ to $j$ if there is a directed path from $i$ to $j$
- In practice, the influence graph will likely include false positive and false negative edges.
Step I) Determining Causal Orderings

- Create a hyper-graph of strong connected components (SCC), where each node is a collection of $\geq 1$ nodes that cannot be further ordered (i.e. there is a cycle).
- Find an ordering (topological sorting) of the SCC graph (note, this is by construction a DAG) using Depth First Search algorithm (DFS).
- Find all possible orderings of each connected component (using backtracking algorithm of Knuth, or Monte Carlo DFS MC-DFS).

Step II) Estimation of the Structure

- Given a topological ordering of nodes, the nodes of the graph can be rearranged to form a DAG.
- For each ordering, estimate (the structure of) one DAG using the penalized likelihood method of the previous lecture, (by solving $p - 1$ lasso regression problems):

$$
\hat{A}_{k,1:k-1} = \arg\min_{\theta \in \mathbb{R}^{k-1}} \left\{ n^{-1}\|X_{1:k-1}\theta - X_{k}\|_2^2 + \lambda \sum_{j=1}^{k-1} |\theta_j|w_j \right\}
$$
Step II) Estimation of the Structure

- Given a topological ordering of nodes, the nodes of the graph can be rearranged to form a DAG.
- For each ordering, estimate (the structure of) one DAG using the penalized likelihood method of the previous lecture, (by solving $p-1$ lasso regression problems):

$$
\hat{A}_{k,1:k-1} = \arg \min_{\theta \in \mathbb{R}^{k-1}} \left\{ n^{-1} \| X_{1:k-1} \theta - X_{k} \|_2^2 + \lambda \sum_{j=1}^{k-1} |\theta_j| w_j \right\}
$$

$$
\xymatrix{ X_1 \ar[r]^{\rho_{32}} & X_2 \ar[r]^{\rho_{23}} & X_1 }
$$

Step III) Building a Consensus Graph

- For each ordering, the estimated graph is a DAG.
- However, the true graph may include cycles. Also, results from one ordering may be inaccurate (noise...).
Step III) Building a Consensus Graph

- For each ordering, the estimated graph is a DAG.
- However, the true graph may include cycles. Also, results from one ordering may be inaccurate (noise...).

Solution: average over edges with the best scores:

$$\hat{A}_{i,j} = \frac{1}{|Q|} \sum_{k \in Q} 1_{\{A_{i,j}^k > 0\}} \quad \hat{E} = \{(i,j) : \hat{A}_{i,j} \geq \tau\}$$

- $L_q$: lower $q$th quantile of (penalized) negative log-likelihoods
- $Q = \{o \in O : \ell(o) \leq L_q\}$ set of orderings for these likelihoods
Simulate Network: DAG of size $p = 20$

Data Generation

Perturbation data: Adjacency matrices of true and noisy influence graphs

Steady-state expression data: generated $n = 50$ Gaussian observations according to the true DAG.
Comparison of $F_1$ measures

How Many Orderings?
For $P_3$, 3,962 orderings are found using backtracking.
High Dimensional Cyclic Graphs \((p = 1000)\)

**Effect of FP and FN errors**

![Graph showing effect of FP and FN errors](image)

A More Complicated Example: DREAM-4 Challenge

- The DREAM project (Dialogue for Reverse Engineering Assessments and Methods) is an attempt to construct realistic regulatory networks.
- DREAM-4 challenge had multiple competitions, including reverse engineering 5 networks of size 100 selected from true regulatory components of yeast and E-coli.
- The perturbation data is simulated based on the true network (using coupled ODE).
- Two types of perturbation data are available: **knockout** and **knockdown** experiments.
- Pinna et al (PINNA) was the winner of the high dimensional reconstruction challenge (on networks of size 100).
DREAM Network 1 (Simplest)

DREAM Network 5 (Most Difficult!)
### Comparison of $F_1$ Measures

![Comparison of $F_1$ Measures](image)

### Example of estimated modules

**Largest cyclic component in DREAM1 network**

When the perturbation data includes cycles, the consensus graph will be cyclic.
Network of Yeast Transcription Factors

- 269-node corresponding to known yeast TF's \((p = 269)\)
- **Perturbation data**: knockout experiments from Hu et al (2007, Nat Genetics)
- **Steady-state expression data**: \(n = 200\) day-to-day variation samples of yeast (publicly available), not really iid!
- Used 10,000 orderings
- To evaluate: use available data on yeast regulatory network, which is (most likely) incomplete. Therefore, "false positives" may be true edges

Significance of true positives (TP), in comparison to the BioGrid network

Histograms show number of TP’s in random networks of equal sizes
Extension: $k \ll p$

- In many biological experiments, perturbation screens are only run on a subset of genes ($k$ out of $p$)
- If perturbation is available on TFs, the RIPE algorithm can be modified to estimate the network

RIPE Performance in yeast regulatory network (6051 genes)

TP = 134, |E| = 10014 ($p$-value < 0.001)
Summary

• Estimation of regulatory networks is difficult! In addition to need for causal inference, the presence of feedback loops, and the small sample size of biological experiments hinder estimation of directed regulatory networks.

• Available data differ in informational content and available sample size (and hence noise level).

• Time-course and perturbation data offer greater potential for learning the structure of DAGs; however, they also introduce new challenges.

• Computational complexity is a bottleneck of many proposed methods, many existing methods are approximations of the biology, or make strong assumptions.

• This is an active area of research, with many methods being developed and implemented...
Introduction
Topology-Based Pathway Enrichment Methods
Systematic Comparisons

Pathway & Network Analysis of Omics Data:
Topology-Based Pathway Enrichment Analysis

Ali Shojaie
Department of Biostatistics
University of Washington
faculty.washington.edu/ashojaie

Summer Institute for Statistical Genetics – 2018

Recap: Gene Set Enrichment Analysis

Subramanian et al. (2005) proposed gene set enrichment analysis (GSEA); Efron & Tibshirani (2007) formalized the GSEA approach, and proposed a more efficient test statistic

- Tests the significance of a priori defined gene sets
- Preserves the correlation among genes in the gene set
- Competitive null hypothesis, where activity of each pathway is compared with other pathways, using a permutation test
  - Competitive tests assume that few genes have differential activity, and may be sensitive to the choice of gene sets
  - Self-contained tests address these issues, but may be sensitive to their modeling assumptions (Goeman & Buhlmann (2007), Ackermann & Strimmer (2009))
Yeast GAL Pathway
Ideker et al, 2001

Issues of Interest

- Incorporate the network information
- Consider changes in the gene (protein, metabolite) expressions
- Consider changes in the network structure
- Test the “effect” of pre-specified subnetwork/pathway, sharing common biological function, chromosomal location etc
- A general framework for inference in complex experiments
PathNet (Dutta et al, 2012)

A simple topology-based pathway enrichment method:

Each gene’s \( p \)-value from differential expression is combined with \( p \)-values of its neighbors using Fisher’s methods

\[
SI_j = \sum_{k \in \text{nn}(j)} \left\{ -\log_{10} \left( p_k^D \right) \right\}.
\]

- The indirect \( p \)-value, \( p^I \) is calculated from \( SI_j \) by permutation
- Direct \( (p_j^D) \) and indirect \( (p_j^I) \) \( p \)-values are then combined \( (p_j^C) \)
- The significance of \( p_j^C \) for genes in each pathway is assessed using a hypergeometric test
- Implemented in Bioconductor package PathNet
**topologyGSA (Massa et al, 2010)**

- topologyGSA (Gene Set Analysis Exploiting Pathway Topology) assumes that data are normally distributed:
  \[ X^1 \sim N(\mu^1, \Sigma^1), \quad X^2 \sim N(\mu^2, \Sigma^2) \]
- It obtains estimates of \( \Sigma^1 \) and \( \Sigma^2 \) based on the networks (think graphical lasso, but with known nonzero entries)
- It then performs two tests:
  - equality of covariance matrices: \( H_0^c : \Sigma^1 = \Sigma^2 \)
  - equality of means \( H_0^m : \mu^1 = \mu^2 \) — it uses different methods depending on the result of \( H_0^c \)
- Implemented in R-package topologyGSA (also in graphite)

---

**Signaling Pathway Impact Analysis (SPIA)**

- Combines overrepresentation analysis (ORA) with measure of perturbation of a given pathway under a given condition
- A bootstrap procedure is used to assess the significance of the observed pathway perturbation (difficult to extend to comparison of > 2 conditions)
- Currently not applicable to all pathways (more later)
- Analyzes each pathway separately (ignores connections between pathways)
- Implemented in the Bioconductor package SPIA
The SPIA Methodology

SPIA combines two types of evidence
The SPIA Methodology

SPIA combines two types of evidence

(i) the overrepresentation of DE genes in a given pathway

- measured by the p-value for the given number of DE genes

\[ P_{NDE} = P(X \geq N_{DE} | H_0) \]
The SPIA Methodology

SPIA combines two types of evidence

(ii) the abnormal perturbation of the pathway
The SPIA Methodology

SPIA combines two types of evidence

(ii) the abnormal perturbation of the pathway
  ▶ the perturbation for each gene in the pathway is defined as
  \[ PF(g_i) = \Delta E(g_i) + \sum_{j=1}^{p} \beta_{ij} \frac{PF(g_j)}{N_{DS}(g_j)} \]

\( PF(g_i) \) is the perturbation factor of gene \( i \) (not known)
\( \beta_{ij} \) is the magnitude of effect of gene \( j \) on gene \( i \); currently, \( \beta_{ij} = 1 \) if \( j \rightarrow i \)
\( \Delta E(g_i) \) is the fold change in expression of gene \( i \)
\( N_{DS}(g_j) \) is the number of downstream genes from gene \( j \)
The SPIA Methodology

- The accumulated activity of each gene can then be calculated as $ ACC(g_i) = B \cdot (I - B)^{-1} \Delta E $
The SPIA Methodology

- The accumulated activity of each gene can then be calculated as $ACC(g_i) = B \cdot (I - B)^{-1} \Delta E$
  - $B$ is the normalized matrix of $\beta$’s: $B_{ij} = \beta_{ij} / N_{DS}(g_j)$
  - $\Delta E$ is the vector of fold changes
  - Requires $B$ to be invertible; would not work otherwise

- The total accumulated perturbation of the pathway is then given by $t_A = \sum_i ACC(g_i)$
The SPIA Methodology

- The accumulated activity of each gene can then be calculated as $ACC(g_i) = B \cdot (I - B)^{-1} \Delta E$
  - $B$ is the normalized matrix of $\beta$'s: $B_{ij} = \beta_{ij} / N_{DS}(g_j)$
  - $\Delta E$ is the vector of fold changes
  - Requires $B$ to be invertible; would not work otherwise
- The total accumulated perturbation of the pathway is then given by $t_A = \sum_i ACC(g_i)$
- The p-value for pathway perturbation is given by $P_{PERT} = P(T_A \geq t_A \mid H_0)$, which is calculated using a bootstrap approach
The SPIA Methodology

SPIA combines two types of evidence
The SPIA Methodology

SPIA combines two types of evidence

- The final p-value for each pathway is calculated based on the p-values from parts (i) and (ii):
The SPIA Methodology

SPIA combines two types of evidence
- The final p-value for each pathway is calculated based on the p-values from parts (i) and (ii):
  - \( P_G(i) = c_i - c_i \ln(c_i) \)
  - \( c_i = P_{NDE}(i)P_{PERT}(i) \)

An Example in R: Data on Colorectal Cancer

data(colorectalcancer)
#pathway analysis using SPIA
#use nB=2000 or higher for more accurate results
#uses older version of KEGG signaling pathways graphs
res <- spia(de=DE_Colorectal, all=ALL_Colorectal, organism="hsa", beta=NULL, nB=2000, plots=FALSE, verbose=TRUE, combine="fisher")
#now combine pNDE and pPERT using the normal inversion method without running spia function again
res$pG=combfunc(res$pNDE,res$pPERT,combine="norminv")
res$pGfdr=p.adjust(res$pG,"fdr")
res$pGFWER=p.adjust(res$pG,"bonferroni")
plotP(res,threshold=0.05)
#highlight the colorectal cancer pathway in green
points(I(-log(pPERT)))~I(-log(pNDE)),data=res[res$ID=="05210",],col="green", pch=19,cex=1.5)
The SPIA Methodology

Network-Based Gene Set Analysis (NetGSA)

- Combines the ideas of gene set analysis methods, and network-based single gene analysis
- Generalizes SPIA, to allow for more complex experiments & incorporate interactions among pathways
- Assesses the overall behavior of arbitrary subnetworks (pathways): changes in gene expression & network structure
- Uses latent variables to model the interaction between genes defined by the network
- Uses mixed linear models for inference in complex data
- Computationally challenging for large networks (e.g. not applicable to whole genome sequencing data) unless, pathways separated (similar to SPIA)
Problem Setup

- Gene (protein/metabolite) expression data for $K$ experimental conditions and $J_k$ time points
- Network information (partially) available in the form of a directed weighted graph $G = (V, E)$, with vertex set $V$ corresponding to the genes/proteins/metabolites and edge set $E$ capturing their associations
- Network edges can be directed $j \rightarrow k$ or undirected $j \leftrightarrow k$
- Edges define the effect of nodes on their immediate neighbors; the weight associated with each edge corresponds to the value of partial correlation
- Represent the network by its adjacency matrix $A$: $A_{jk} \neq 0$ iff $k \rightarrow j$ & for undirected edges, $A_{jk} = A_{kj}$
- Pathways defined a priori based on common biological functions, etc

The Latent Variable Model: Main Idea

\[
\begin{align*}
X_1 & = \gamma_1 \\
X_2 & = \rho_{12}X_1 + \gamma_2 = \rho_{12}\gamma_1 + \gamma_2 \\
X_3 & = \rho_{23}X_2 + \gamma_3 = \rho_{23}\rho_{12}\gamma_1 + \rho_{23}\gamma_2 + \gamma_3
\end{align*}
\]
The Latent Variable Model: Main Idea

\[
\begin{align*}
X_1 &= \gamma_1 \\
X_2 &= \rho_{12}X_1 + \gamma_2 = \rho_{12}\gamma_1 + \gamma_2 \\
X_3 &= \rho_{23}X_2 + \gamma_3 = \rho_{23}\rho_{12}\gamma_1 + \rho_{23}\gamma_2 + \gamma_3
\end{align*}
\]

Thus \( X = \Lambda\gamma \) where

\[
\Lambda = \begin{pmatrix}
1 & 0 & 0 \\
\rho_{12} & 1 & 0 \\
\rho_{12}\rho_{23} & \rho_{23} & 1
\end{pmatrix}
\]

The Latent Variable Model

- Let \( Y \) be the \( i \)th sample in the expression data
- Let \( Y = X + \varepsilon \), with signal \( X \) and noise \( \varepsilon \sim N_p(0, \sigma_\varepsilon^2 I_p) \)
- The influence matrix \( \Lambda \) measures the propagated effect of genes on each other through the network, and can be calculated based on the adjacency matrix \( A \)
- Using \( X = \Lambda\gamma \), we get

\[
Y = \Lambda\gamma + \varepsilon, \quad \Rightarrow \quad Y \sim N_p(\Lambda\mu, \sigma_\gamma^2 \Lambda^2 + \sigma_\varepsilon^2 I_p)
\]

where \( \gamma \sim N_p(\mu, \sigma_\gamma^2 I_p) \) are latent variables
Mixed Linear Model Representation

Rearranging the expression matrix into $np$-vector $\mathbf{Y}$, we can write

$$\mathbf{Y} = \Psi \beta + \Pi \gamma + \varepsilon$$

where $\beta$ and $\gamma$ are fixed and random effect parameters and

$$\varepsilon \sim N_{np}(0, R(\theta_\varepsilon)), \quad \gamma \sim N_{np}(0, \sigma_\gamma^2 \mathbf{I}_{np})$$

- **Temporal Correlation** incorporated through $R$

In general, the design matrices, $\Psi$ and $\Pi$ depend on the experimental settings (similar to ANOVA), and are **functions of $\Lambda$**
Estimation of MLM Parameters

MLE for $\beta$:

$$\hat{\beta} = (\psi' \hat{W}^{-1} \psi)^{-1} \psi' \hat{W}^{-1} Y$$

where $W = \sigma^2 \gamma \Pi' + R$.

$\hat{\beta}$ depends on estimates of $\sigma^2$ and $\theta^2$ (estimated using restricted maximum likelihood (REML)).

Inference using MLM

- Let $\ell$ be a contrast vector (a linear combination of fixed effects), and consider the test:

  $$H_0 : \ell \beta = 0 \quad vs. \quad H_1 : \ell \beta \neq 0$$
Inference using MLM

- Let $\ell$ be a contrast vector (a linear combination of fixed effects), and consider the test:

$$H_0 : \ell \beta = 0 \quad \text{vs.} \quad H_1 : \ell \beta \neq 0$$

- Use t-test to test the significance of each hypothesis separately

$$T = \frac{\ell \hat{\beta}}{\sqrt{\ell C \ell'}}$$

where $C = (\Psi' W^{-1} \Psi)^{-1}$

- Under the null hypothesis, $T$ is approximately $t$-distributed with degrees of freedom that needs to be estimated

“Optimal” Choice of Contrast Vector

- One intuitive choice is to use the indicator vector for the members of pathway $b$, but this only reflects changes in the mean vector
- Need to de-couple the effect of each subnetwork from other nodes
“Optimal” Choice of Contrast Vector

- One intuitive choice is to use the indicator vector for the members of pathway $b$, but this only reflects changes in the mean vector.
- Need to *de-couple the effect of each subnetwork* from other nodes.

Can be shown that $(b \cdot b)^\gamma$ is not affected by nodes outside $b$, but includes the effects of nodes in $b$ on each other.

In the case-control case, the optimal contrast vector is:

$$\ell^* = \left(-b \cdot b^C, b \cdot b^T\right)$$
“Optimal” Choice of Contrast Vector

Consider the set, \( b = (0, 1, 1) \); then

\[
(b \Lambda) = \left( \rho_{12} + \rho_{12} \rho_{23}, 1 + \rho_{23}, 1 \right)
\]

On the other hand,

\[
(b \Lambda \cdot b) = (0, 1 + \rho_{23}, 1)
\]
Yeast Galactose Utilization Pathway

*Ideker et al* (2001) data on yeast Galactose Utilization Pathway

- Gene expression data for 2 experimental conditions: (gal+) and (gal–)
- Gene-gene and protein-gene interactions as well as association weights found from previous studies
- Q: which pathways respond to the change in growth medium?

Analysis of Yeast GAL Data

- Data:
  - gene expression data for 343 genes
  - 419 interactions found from previous studies and integration with protein expression (*association among genes also available*)
- Results:
  - GSEA finds *Galactose Utilization Pathway* significant
  - NetGSA finds several other pathways with biologically meaningful functions related to survival of yeast cells in gal–
Environmental Stress Response in Yeast

Gene expression data on Yeast Environmental Stress Response (ESR) (Gasch et al., 2000)

- 3 combinations of experimental factor, heat shock and osmotic changes (sorbitol), over 3 time points
- Temporal correlation
- Network correlation
- Q: Which pathways indicate response to environmental stress
  - in different experimental conditions
  - over time

Yeast ESR Data

- Gene Expression Data

<table>
<thead>
<tr>
<th>Experiment</th>
<th>Obs. Time (after 33C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mild heat shock (29C to 33C), no sorbitol</td>
<td>5, 15, 30 min</td>
</tr>
<tr>
<td>Mild Heat Shock, 1M sorbitol at 29C &amp; 33C</td>
<td>5, 15, 30 min</td>
</tr>
<tr>
<td>Mild Heat Shock, 1M sorbitol at 29C</td>
<td>5, 15, 30 min</td>
</tr>
</tbody>
</table>

- Network Data
  - Use YeastNet (Lee et al., 2007) for gene-gene interactions (102,000 interactions among 5,900 yeast genes)
  - Use independent experiments of Gasch et al. to estimate weights
  - Pathways are defined using GO functions
Model and Results

- Model: Let $j$ and $k$ be indices for time and levels of sorbitol

\[
\mathbb{E}Y_{11} = \Lambda \mu, \quad \mathbb{E}Y_{jk} = \Lambda (\mu + \alpha_j + \delta_k) \quad j, k = 2, 3
\]

- Temporal correlation is modeled directly via $R$ (as AR(1) process)

- Results:
  - $\sim 3000$ genes,
  - 47 pathways showed significant changes of expression
  - 24 pathways showed changes over time
  - 29 pathways showed changes in response to different sorbitol levels
  - 12 pathways showed both types of changes
  - Significant pathways overlap with the gene functions recognized by Gasch et al.

Yeast ESR Network

- Non-DE
- DE

- Pos Effect
- Neg Effect
Significant subnetworks

- a) Cell Cycle
- b) Secretion
- c) Signaling
- d) Respiration

Expression Profiles
Average Standardized Expression Levels of Pathways

- **Induced** and **Suppressed** Pathways
- Can observe the **transient patterns of expressions** as predicted by **Gasch et al.**
Effect of Noise In Network Information

- Let $\tilde{A}$ be observed network information, and $A$ be the truth.
- It can be shown that, if $\|\tilde{A} - A\|$ is small then, NetGSA still works (is asymptotically most powerful unbiased test)
Metabolic Profiling in Bladder Cancer

Targeted metabolic profiling of bladder cancer (BCa) (Putluri et al., 2012)

- 58 bladder cancer and adjacent benign samples
- Pathways information obtained from KEGG

- Varying number of identified metabolites per pathway (3-15)
- Q: Which pathways show differential activity in BCa?

Metabolic Profiling in BCa

- 63 metabolites identified, mapped to 70 pathways
- 27 pathways with at least 3 members
Metabolic Profiling in BCa

- Small pathway sizes & significant overlap among pathways

![Histograms showing metabolites in pathway and pathways overlap](image)

- Existing methods may not work well...

Metabolic Interaction Network

![Network diagram](image)
Significant Pathways

- **GSEA** does not identify any pathway as differential
- **GSA** identifies **Fatty Acid Biosynthesis** as differential
- **NetGSA** identifies another 7 pathways corresponding to role of **Amino Acid Metabolism** in BCa, similar to *Putluri et al* (2012)

R package **netgsa**

- **Basic usage:**
  
  ```r
  NetGSA(x, y, A, B)
  ```

  - A: a list of $p \times p$ weighted adjacency matrices, e.g. for each condition (e.g. normal vs cancer), to allow for changes in the network
  - B: a $K \times P$ 0-1 matrix of pathway membership: $B_{k,j} = 1$ if gene/protein/metabolite $j$ in pathway $k$
  - **Output:** test statistics and p-values for each pathway
  - The NetGSA function takes a weighted $A$ as input. The package includes functions to learn $A$ for undirected networks from a (partial) list of network edges
    - function to extract this info from existing datasets coming soon
Comparison Using Synthetic Data (Ma, S., Michailidis, 2018)

- Comparison of topology-based pathway enrichment methods using two synthetic data sets
  - Gene expression data \( p \approx 3000 \)
  - Metabolomics data \( p \approx 100 \)
- *In silico* data sets with known signal:
  1. Remove the original signal, but *keep the correlation structure*
  2. *Perturb means in one condition* (differential expression) for nodes in selected pathways
  3. Also use sample permutation to *create data with equal correlation structure*
Results for Gene Expression Data — Equal Covariance

Results for Gene Expression Data — Diff Covariance
Results for Gene Expression Data

Results for Metabolomics Data — Equal Covariance

\[ \text{DC} \leq 0.5 \text{ & size} \leq 11 \]

\[ \text{DC} > 0.5 \text{ & size} \leq 11 \]

\[ \text{DC} \leq 0.5 \text{ & size} > 11 \]

\[ \text{DC} > 0.5 \text{ & size} > 11 \]
Introduction
Topology-Based Pathway Enrichment Methods
Systematic Comparisons

Results for Metabolomics Data — Diff Covariance

DC <= 0.5 & size <= 11

DC <= 0.5 & size > 11

DC > 0.5 & size <= 11

DC > 0.5 size > 11

PathNet
CAMERA
DEGraph
NetGSA

© Ali Shojaie
SISG: Pathway & Networks
Summary

- Network-based enrichment analysis methods (SPIA, NetGSA) can be more powerful (if their assumptions are not violated!)
- Active area of research: a number of other methods have been recently proposed
- Focus is shifting towards estimating changes in the structure of networks: differential network biology

\[^1\text{Ideker & Krogan (2012)}\]