Pathway & Network Analysis of Omics Data: Networks in Biology

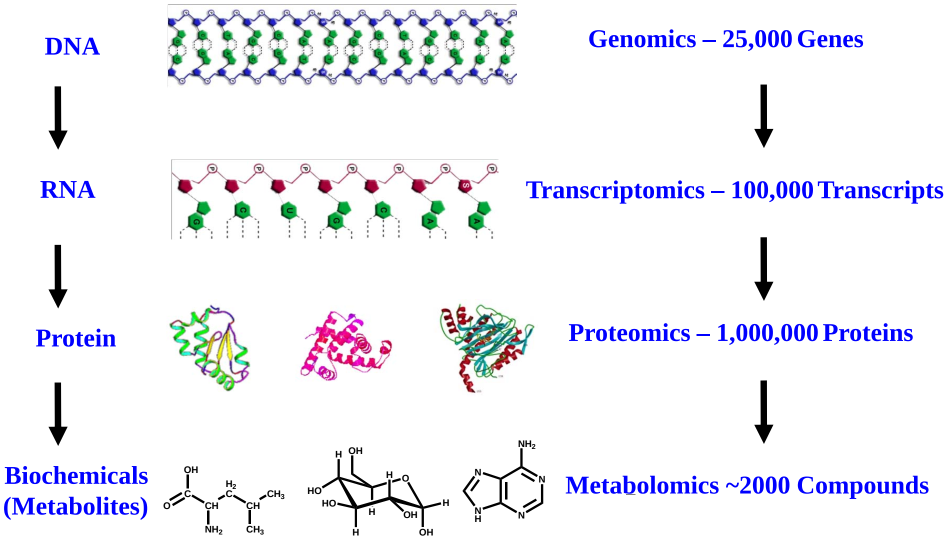
Ali Shojaie
Department of Biostatistics
University of Washington
faculty.washington.edu/ashojaie

Summer Institute for Statistical Genetics – 2016

Why Study Networks?

- Components of biological systems, e.g. genes, proteins, metabolites, interact with each other to carry out different functions in the cell.
- 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 ($2 + 2 \neq 4!$).
- Networks provide an efficient representation of complex reaction in the cells, as well as basis for mathematical/statistical models for the study of these systems.

Central Dogma of Molecular Biology (Extended)



DNA → RNA → Protein → Biochemicals (Metabolites)

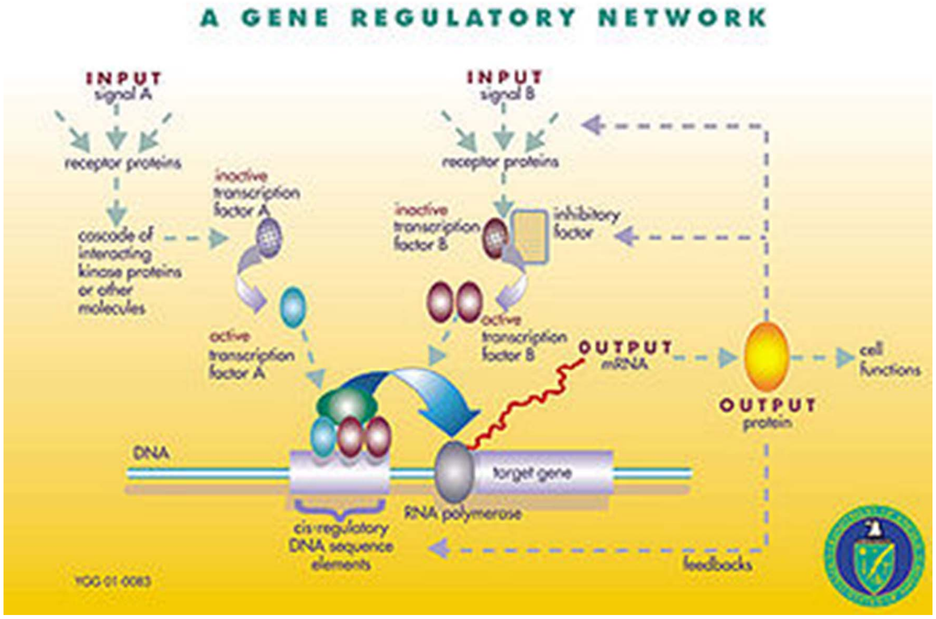
Genomics – 25,000 Genes

Transcriptomics – 100,000 Transcripts

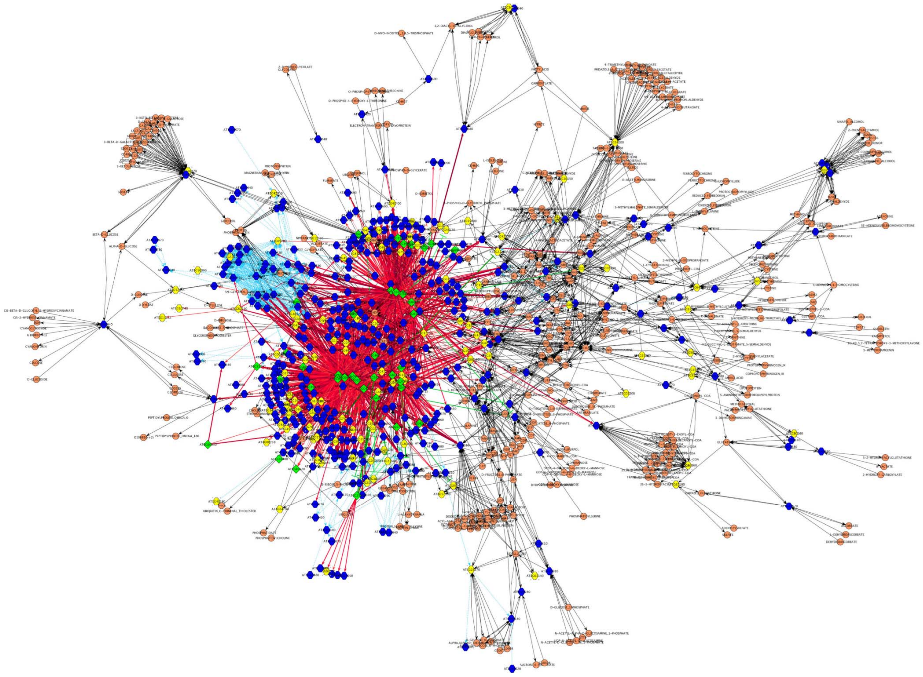
Proteomics – 1,000,000 Proteins

Metabolomics ~2000 Compounds

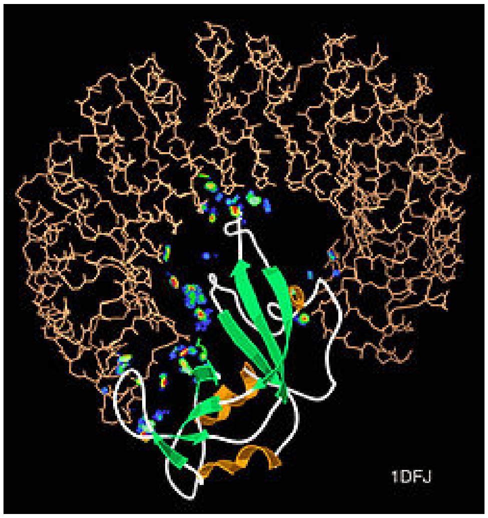
Networks in Biology: Gene Regulatory Interactions


A GENE REGULATORY NETWORK
INPUT signal A
INPUT signal B
receptor proteins
receptor proteins
inactive transcription factor A
cascade of interacting kinase proteins or other molecules
inactive transcription factor B
inhibitory factor
active transcription factor A
active transcription factor B
OUTPUT mRNA
cell functions
OUTPUT protein
DNA
target gene
RNA polymerase
cis-regulatory DNA sequence elements
feedbacks
YOG-01-0083

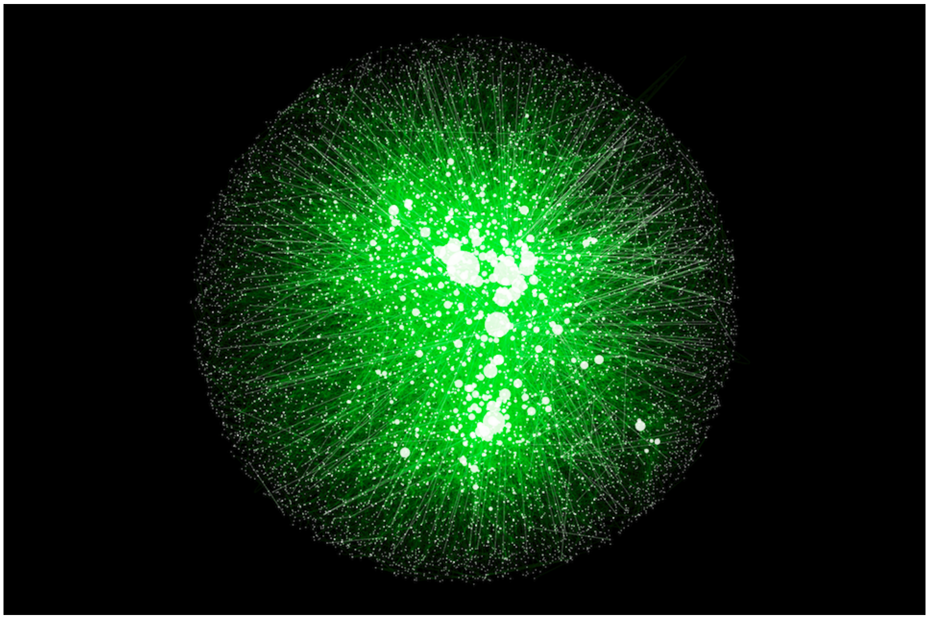
Networks in Biology: Gene Regulatory Networks



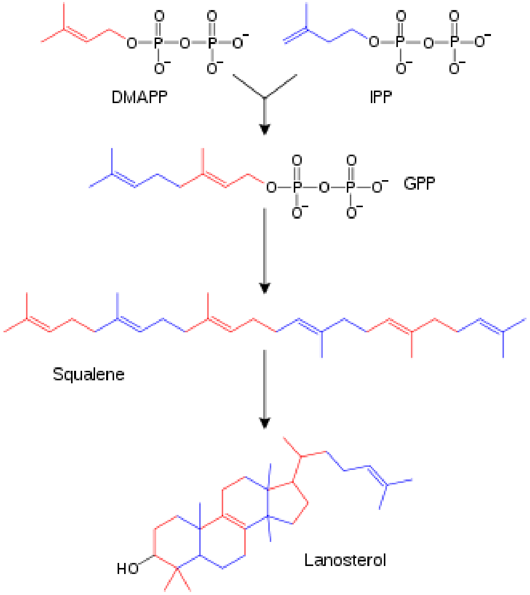
Networks in Biology: Protein-Protein Interaction



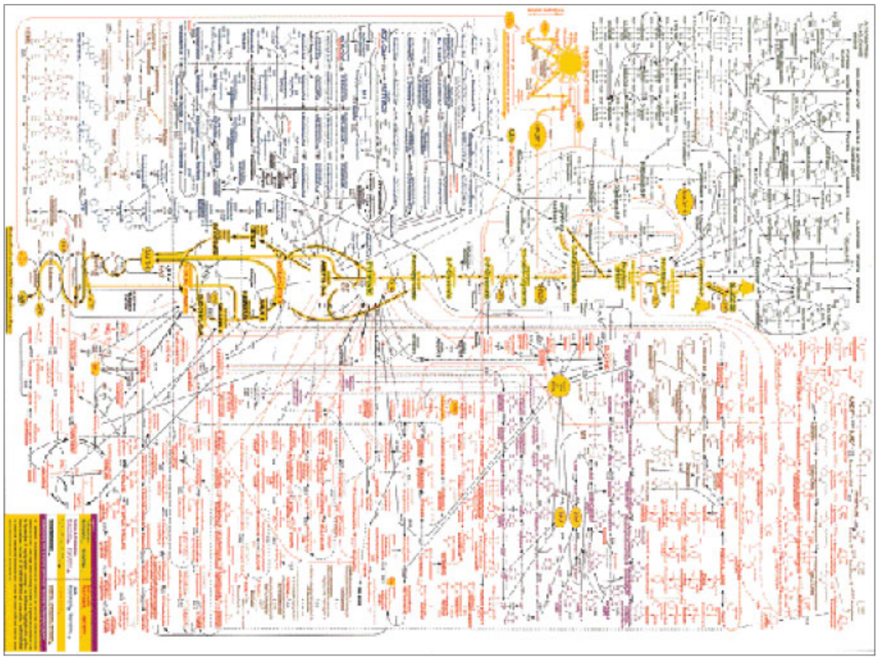
Networks in Biology: Protein-Protein Interaction (PPI) Networks



Networks in Biology: Metabolic Reactions

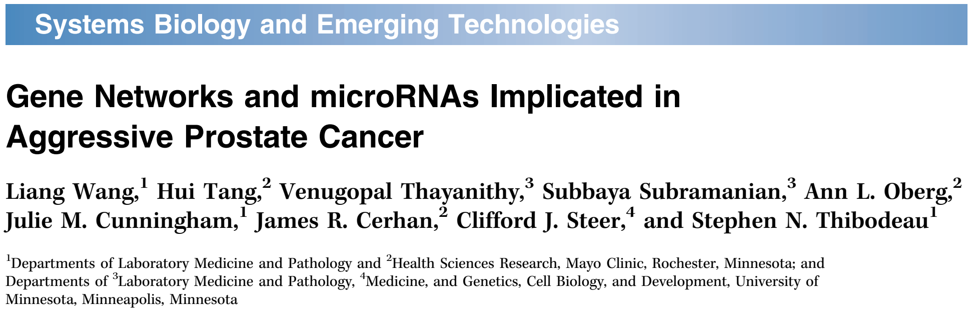


Networks in Biology: Metabolic Pathways



But Do Networks Matter?

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



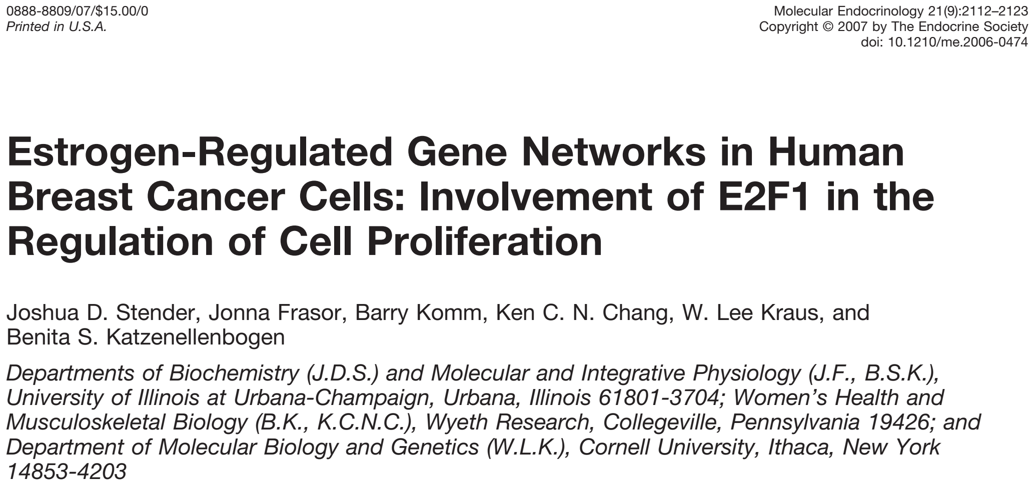
Systems Biology and Emerging Technologies

Gene Networks and microRNAs Implicated in Aggressive Prostate Cancer

Liang Wang,[1] Hui Tang,[2] Venugopal Thayanithy,[3] Subbaya Subramanian,[3] Ann L. Oberg,[2] Julie M. Cunningham,[1] James R. Cerhan,[2] Clifford J. Steer,[4] and Stephen N. Thibodeau[1]

[1]Departments of Laboratory Medicine and Pathology and [2]Health Sciences Research, Mayo Clinic, Rochester, Minnesota; and Departments of [3]Laboratory Medicine and Pathology, [4]Medicine, and Genetics, Cell Biology, and Development, University of Minnesota, Minneapolis, Minnesota

But Do Networks Matter?



0888-8809/07/$15.00/0
Printed in U.S.A.

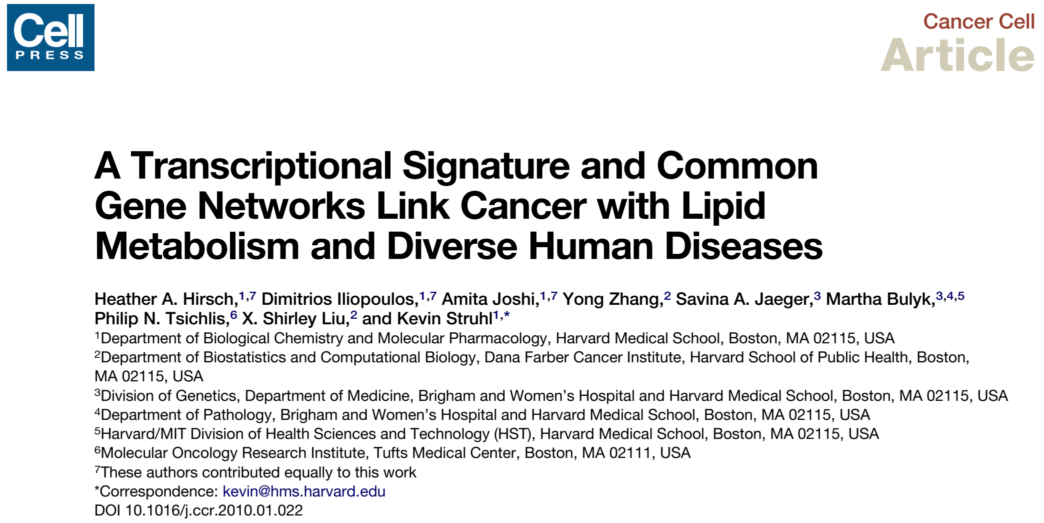
Molecular Endocrinology 21(9):2112–2123
Copyright © 2007 by The Endocrine Society
doi: 10.1210/me.2006-0474

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

But Do Networks Matter?



Cell PRESS

Cancer Cell
Article

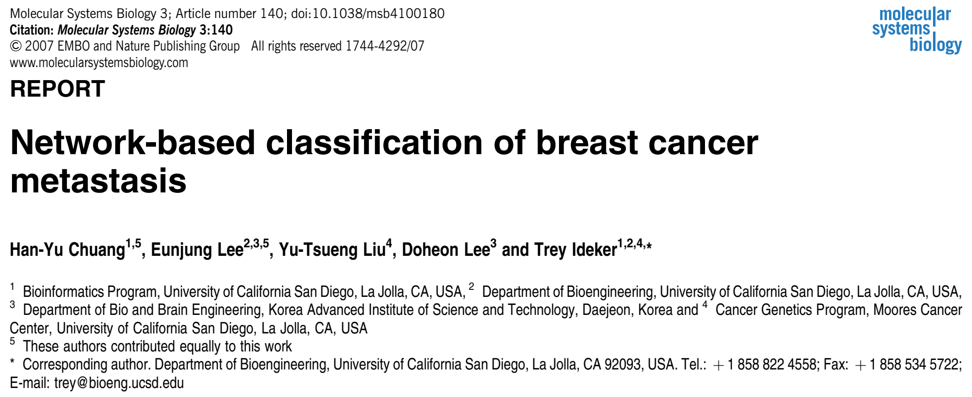
A Transcriptional Signature and Common Gene Networks Link Cancer with Lipid Metabolism and Diverse Human Diseases

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

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



Molecular Systems Biology 3; Article number 140; doi:10.1038/msb4100180
Citation: ***Molecular Systems Biology*** **3:140**
© 2007 EMBO and Nature Publishing Group All rights reserved 1744-4292/07
www.molecularsystemsbiology.com

molecular systems biology

REPORT

Network-based classification of breast cancer metastasis

Han-Yu Chuang[1,5], Eunjung Lee[2,3,5], Yu-Tsueng Liu[4], Doheon Lee[3] and Trey Ideker[1,2,4,*]

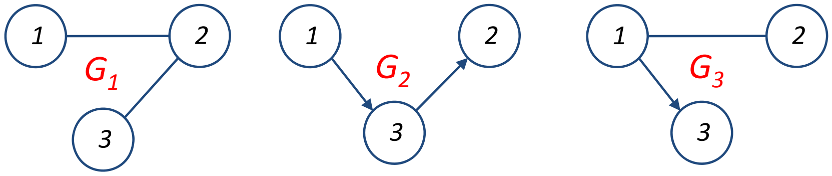
[1] Bioinformatics Program, University of California San Diego, La Jolla, CA, USA, [2] Department of Bioengineering, University of California San Diego, La Jolla, CA, USA, [3] Department of Bio and Brain Engineering, Korea Advanced Institute of Science and Technology, Daejeon, Korea and [4] Cancer Genetics Program, Moores Cancer Center, University of California San Diego, La Jolla, CA, USA
[5] These authors contributed equally to this work
[*] Corresponding author. Department of Bioengineering, University of California San Diego, La Jolla, CA 92093, USA. Tel.: +1 858 822 4558; Fax: +1 858 534 5722; E-mail: trey@bioeng.ucsd.edu

Why Do We Need Network Inference?

- Despite progress, our knowledge of interactions in the genome 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.

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 in the network can be directed $X \rightarrow Y$ or undirected $X - Y$.



- In all these example, the nodes are $V = \{1, 2, 3\}$.
- The edges are:

$$\begin{aligned} E_1 &= \{1 - 2, 2 - 3\} \\ E_2 &= \{1 \rightarrow 3, 3 \rightarrow 2\} \\ E_3 &= \{1 - 2, 1 \rightarrow 3\} \end{aligned}$$

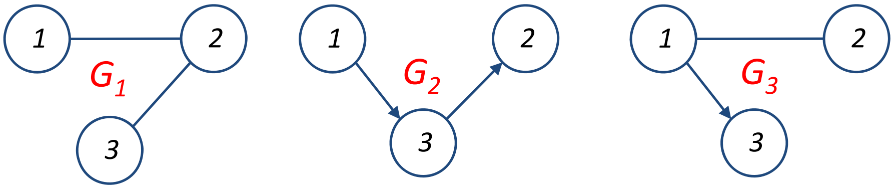
Networks: A Short Premier

- 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

$$A = \begin{bmatrix} . & \mathbf{x} & . \\ . & . & . \\ . & . & . \end{bmatrix}$$ 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
- For undirected edges, we add a **1** in both directions

Networks: A Short Premier



$$A = \begin{bmatrix} 0 & \mathbf{1} & 0 \\ \mathbf{1} & 0 & \mathbf{1} \\ 0 & \mathbf{1} & 0 \end{bmatrix} \quad A = \begin{bmatrix} 0 & 0 & \mathbf{1} \\ 0 & 0 & 0 \\ 0 & \mathbf{1} & 0 \end{bmatrix} \quad A = \begin{bmatrix} 0 & \mathbf{1} & \mathbf{1} \\ \mathbf{1} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

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”
- In other words, each node of the network represents one of the variables in the study
  - In gene regulatory networks, nodes ≡ genes
  - In PPI networks, nodes ≡ proteins
  - In metabolic networks, nodes ≡ 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

An Overview of Methods for Network Inference

Network Inference 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 (`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 reconstructing undirected networks
  - Co-expression Networks (WGCNA)
  - ARACNE
  - Conditional Independence Graphs
    - Gaussian Observations (`glasso`, etc)
    - Non-Gaussian and Non-Linear Data (`nonparanormal`, etc)
- Methods for reconstructing directed networks
  - Bayesian Networks (basic concepts, reconstruction algorithm)
  - Reconstructing directed networks from time-course data (dynamic Bayesian networks)
  - Reconstructing 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 – 2016

An Overview of Network Reconstruction Methods

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
- 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, τ 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 τ.
- 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

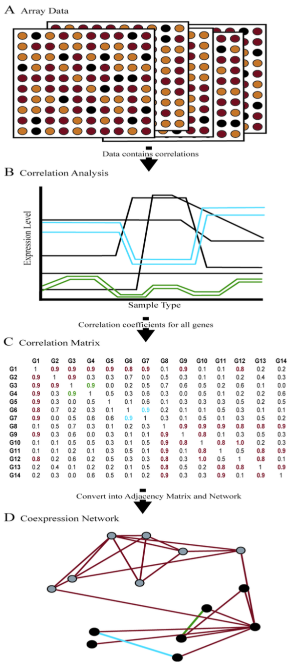
Limitations of Co-Expression Networks

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

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

  - Alternatively, we can work with "weighted" co-expression networks

Weighted Gene Co-expression Network Analysis[1]


A Array Data
Data contains correlations
B Correlation Analysis
Expression Level
Sample Type
Correlation coefficients for all genes
C Correlation Matrix
Convert into Adjacency Matrix and Network
D Coexpression Network

	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14
G1	1	0.9	0.9	0.9	0.9	0.8	0.9	0.1	0.9	0.1	0.1	0.8	0.2	0.2
G2	0.9	1	0.9	0.3	0.3	0.7	0.0	0.5	0.3	0.1	0.1	0.2	0.4	0.3
G3	0.9	0.9	1	0.9	0.0	0.2	0.5	0.7	0.6	0.5	0.2	0.6	0.1	0.0
G4	0.9	0.3	0.9	1	0.5	0.3	0.6	0.3	0.0	0.5	0.1	0.2	0.2	0.6
G5	0.9	0.3	0.0	0.5	1	0.1	0.6	0.1	0.3	0.3	0.3	0.5	0.2	0.5
G6	0.8	0.7	0.2	0.3	0.1	1	0.9	0.2	0.1	0.1	0.5	0.3	0.1	0.1
G7	0.9	0.0	0.5	0.6	0.6	0.9	1	0.3	0.1	0.5	0.1	0.3	0.5	0.2
G8	0.1	0.5	0.7	0.3	0.1	0.2	0.3	1	0.9	0.9	0.9	0.8	0.8	0.9
G9	0.9	0.3	0.6	0.0	0.3	0.1	0.1	0.9	1	0.8	0.1	0.3	0.5	0.3
G10	0.1	0.1	0.5	0.5	0.3	0.1	0.5	0.9	0.8	1	0.8	1.0	0.2	0.3
G11	0.1	0.1	0.2	0.1	0.3	0.5	0.1	0.9	0.1	0.8	1	0.5	0.8	0.9
G12	0.8	0.2	0.6	0.2	0.5	0.3	0.3	0.8	0.3	1.0	0.5	1	0.8	0.1
G13	0.2	0.4	0.1	0.2	0.2	0.1	0.5	0.8	0.5	0.2	0.8	0.8	1	0.9
G14	0.2	0.3	0.0	0.6	0.5	0.1	0.2	0.9	0.3	0.3	0.9	0.1	0.9	1

- 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

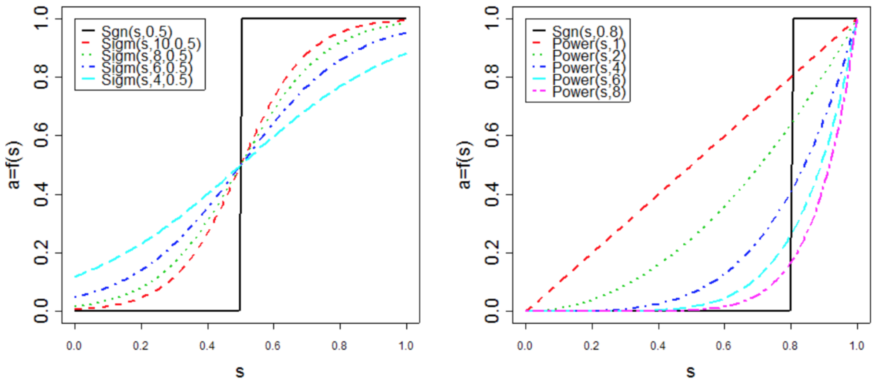
$$A_{ij} = |r_{ij}|^{\beta}$$

- Perform downstream network analysis (clustering, etc) on weighted networks

[1] Zhang and Horvath, A General Framework for Weighted Gene Co-Expression Network Analysis, Stat App in Gen and Mol Bio, 2005

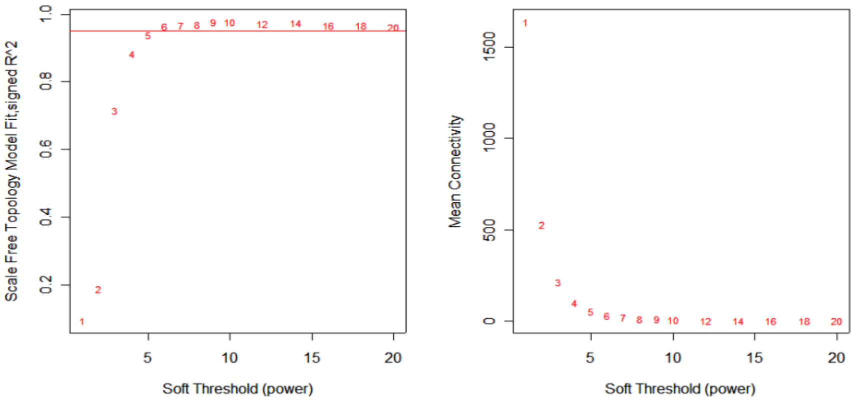
Choice of Parameters

- 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’)
```

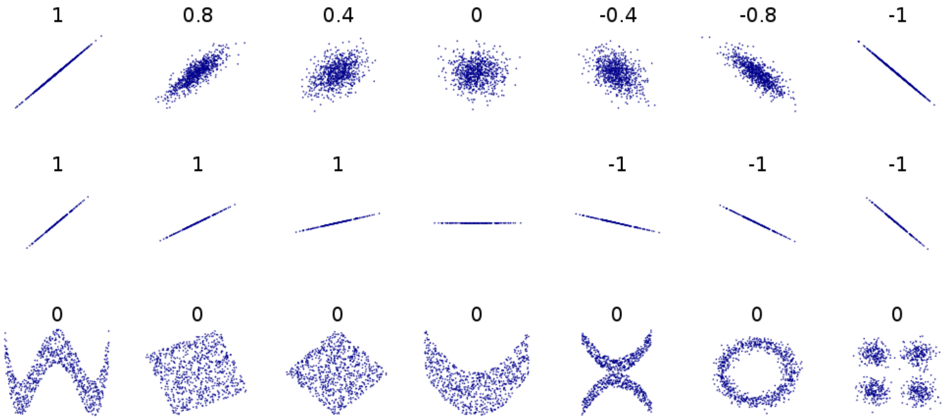
- Main estimation function

```
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

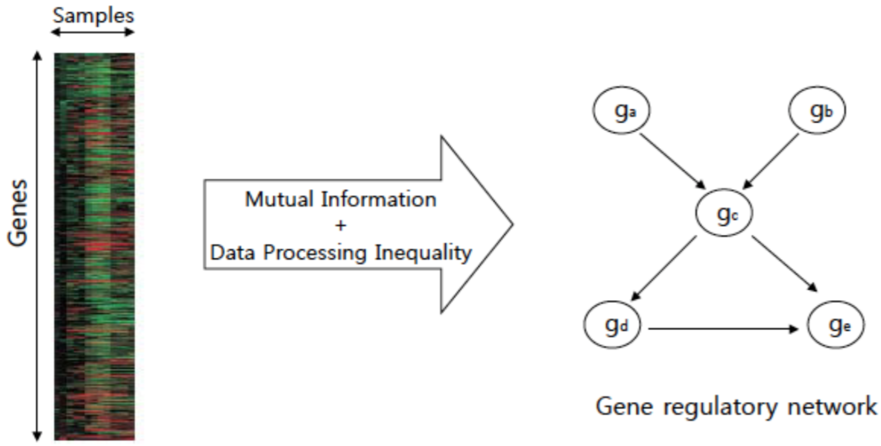
- Correlation is a measure of linear association, but many biological relationships are nonlinear
  - We can use other measures of association, for instance, Spearman correlation or Kendal's τ.
    - These methods define correlation between two variables, based on the ranking of observations, and not 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]

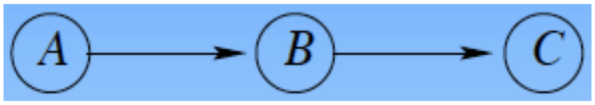
1. Identifies statistically significant gene-gene co-regulation based on mutual information
2. It then eliminates indirect relationships in which two genes are co-regulated through one or more intermediates

[2] ARACNE: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context, Margolin et al, BMC Bioinfo, 2006

ARACNE



Data Processing Inequality (DPI)



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

where

$$I(g_i, g_j) = \log P(g_i, g_j)/P(g_i)P(g_j)$$

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

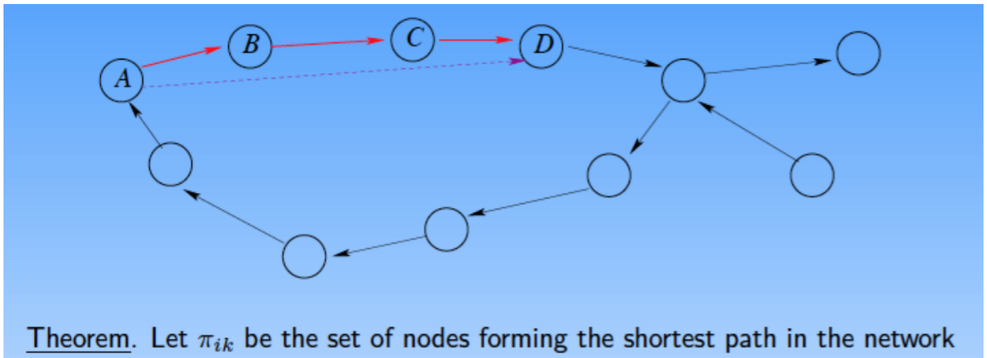
Algorithm Details

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

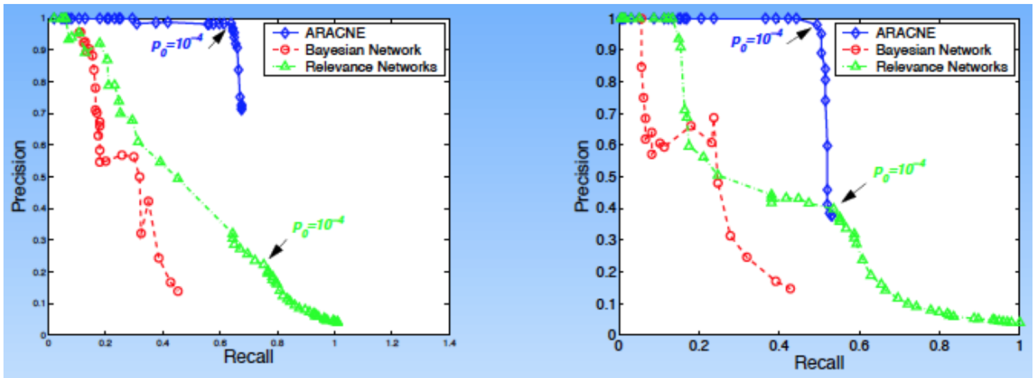
- If MIs can be estimated with no errors, then ARACNE reconstructs the underlying interaction network exactly, provided this network is a tree and has only pairwise interactions.
- The maximum MI spanning tree is a subnetwork of the network built by ARACNE.

Rationale and Guarantees

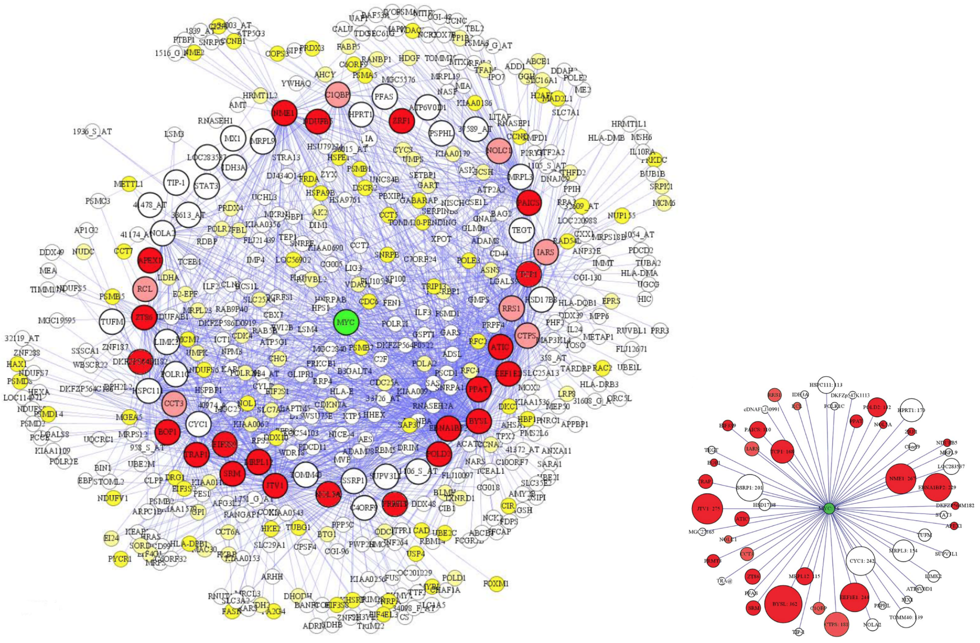


Theorem. Let π_{ik} be the set of nodes forming the shortest path in the network between nodes i and k. Then, if MIs can be estimated without errors, ARACNE reconstructs an interaction network without false positives edges, provided: (a) the network consists only of pairwise interactions, (b) for each $j \in \pi_{ik}$, $I_{ij} \geq I_{ik}$. 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`:

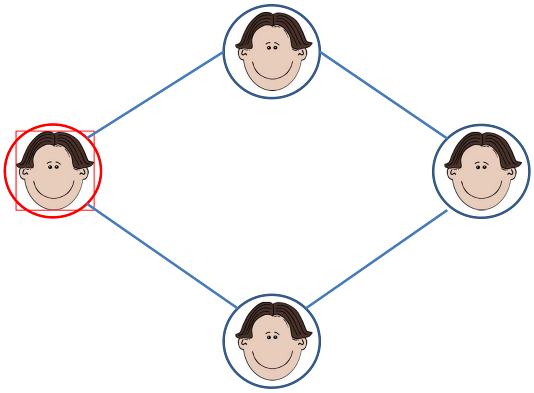
```
source("http://bioconductor.org/biocLite.R")
biocLite("minet")
```

- Main estimation function `aracne(mim, eps=0)`
  - mim: mutual information matrix

    ```
    mim <- build.mim(syn.data, estimator="spearman")
    ```
  - eps: threshold for setting an edge to zero, prior to searching over triplets

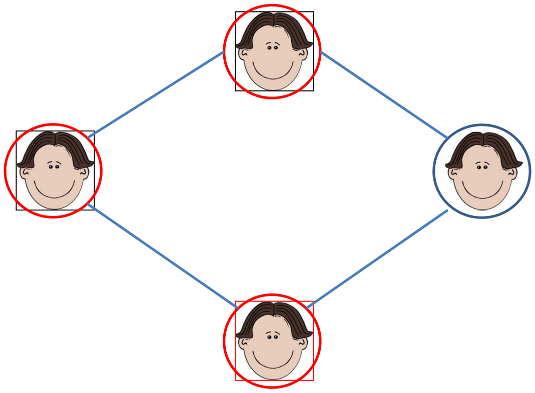
Limitations of Co-Expression Networks

- The estimation is highly dependent on the choice of τ
- 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!



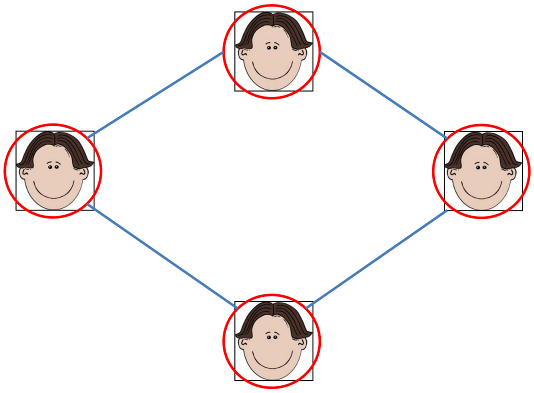
Limitations of Co-Expression Networks

- The estimation is highly dependent on the choice of τ
- 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!



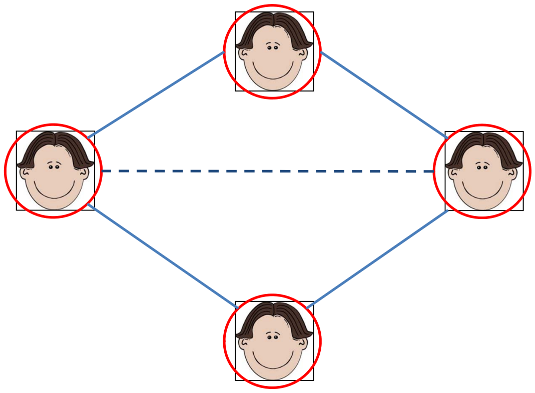
Limitations of Co-Expression Networks

- The estimation is highly dependent on the choice of τ
- 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 τ
- 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 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:

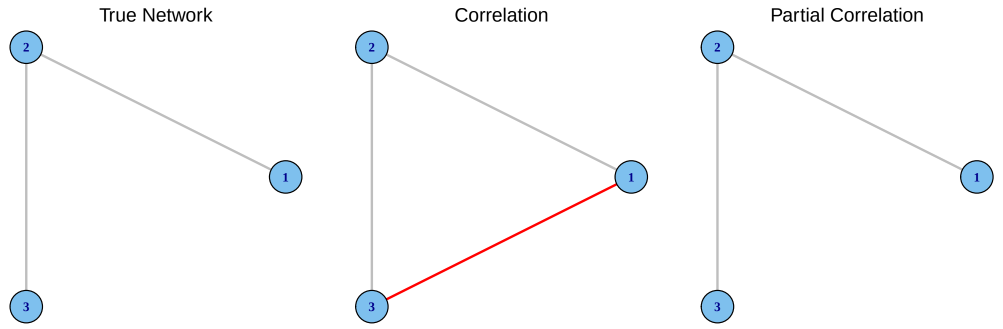
$$\rho_{ij\cdot k} \equiv \rho(X_i, X_j | X_k) = \frac{\rho_{ij} - \rho_{ik}\rho_{jk}}{\sqrt{1-\rho_{ik}^2}\sqrt{1-\rho_{jk}^2}}.$$

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

$$\mathit{Correlation} = \begin{bmatrix} 1 & -.8 & .7 \\ -.8 & 1 & -.8 \\ .7 & -.8 & 1 \end{bmatrix} \mathit{PartialCorr} = \begin{bmatrix} 1 & .6 & 0 \\ .6 & 1 & .6 \\ 0 & .6 & 1 \end{bmatrix}$$



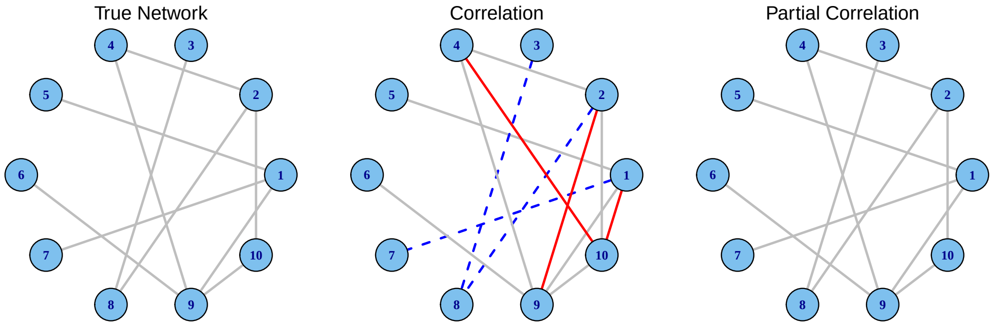
True Network

Correlation

Partial Correlation

A larger example

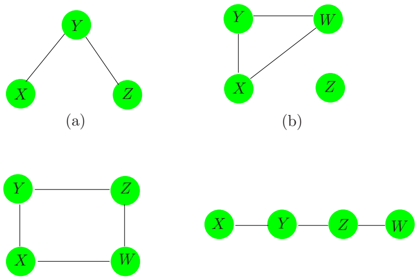
- A network with 10 nodes and 20 edges
- $n = 100$ observations
- Estimation using correlation & partial correlation (20 edges)



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.
- In other words, the (i, j) entry in Σ^{-1} gives the partial correlation between X_i and X_j given all other variables $X_{\backslash i,j}$.
- 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 $\Sigma^{-1}_{ij} \neq 0$

Partial Correlation for Gaussian Random Variables



(a) (b)

(c) (d)

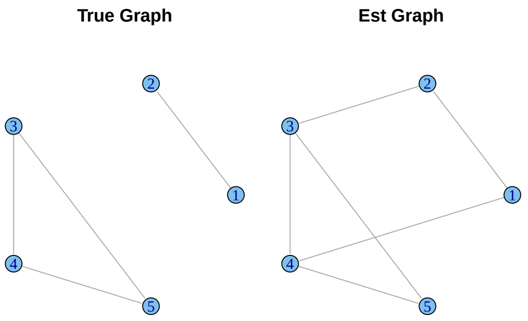
$$\begin{pmatrix} - & x & 0 \\ x & - & x \\ 0 & x & - \end{pmatrix} \quad \begin{pmatrix} - & x & x & 0 \\ x & - & x & 0 \\ x & x & - & 0 \\ 0 & 0 & 0 & - \end{pmatrix}$$

$$\begin{pmatrix} - & x & 0 & x \\ x & - & x & 0 \\ 0 & x & - & x \\ x & 0 & x & - \end{pmatrix} \quad \begin{pmatrix} - & 0 & 0 & x \\ 0 & - & x & 0 \\ 0 & x & - & x \\ x & 0 & x & - \end{pmatrix}$$

Estimation

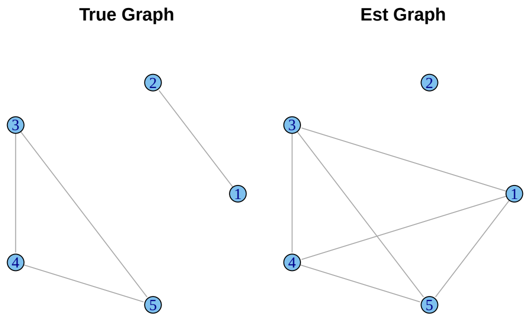
Therefore, to estimate the edges in the graph G,

- First, calculate the empirical covariance matrix of the observations $S = 1/(n-1)X^{\top}X$ (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!!



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 n is not very large, we may still get poor estimates, and we may get more false positives and false negatives.



Estimation in High Dimensions – Method 1

- 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 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 fond from these individual regression problems.

Estimation in High Dimensions – Method 2

- In the second approach, called graphical lasso, we directly estimate the inverse covariance matrix by maximizing the ℓ_1 penalized log likelihood
- It is easy to see that, the log likelihood function of (mean 0) Gaussian random variables can be written as

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

  where Θ is the $p \times p$ inverse covariance matrix (also known as precision matrix).
- Therefore, we can estimate Θ by maximizing the penalized log-likelihood objective function

$$\text{logdet}(\Theta) - \text{tr}(S\Theta) - \lambda\|\Theta\|_1,$$

- Here, logdet gives the logarithm of determinant of matrix; tr gives the trace of the matrix, or some of its diagonal values; and λ is the tuning parameter.

Comparing the Two Approaches

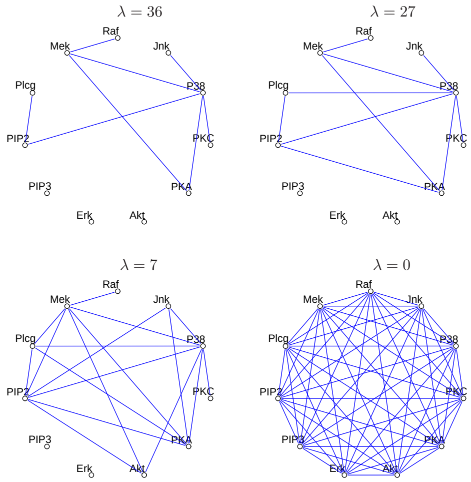
- It turns out that the neighborhood selection approach is an approximation to the graphical lasso problem:
  - Consider regression of X_j on $X_k, j \neq k$
  - Then the regression coefficient for neighborhood selection is related to the j, k element of Θ:

$$\beta_k = -\frac{\Theta_{jk}}{\Theta_{jj}}$$

- 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.
- 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 λ is very difficult in this case.
- As the previous example shows, as λ gets larger, we get sparser graphs.
- However, there is no systematic way of choosing the right λ.
- 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 λ controls the probability of falsely connecting disconnected components at level α (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.

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.
- It turns out that this relationship can be used for solving the graphical lasso problem efficiently.
- The idea is to turn the problem into iterating 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, it is often better to use the empirical correlation matrix

An Example in R

- Download the empirical covariance matrix from `http://www-stat.stanford.edu/~tibs/ElemStatLearn/`
- Install the R-package `glasso`

```
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 denoted by `rho` in the code).
- Try the estimation with and without setting `penalize.diagonal=FALSE`. What do you see?
- Try the estimation with the empirical correlation matrix instead (you may find the function `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 (we can obtain a good approximation using the neighborhood selection approach 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.
- 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
- It is often difficult to validate the estimates; however, in case of biological networks, we can compare our findings with known interactions from literature.

Pathway & Network Analysis of Omics Data: Undirected Graphical Models - II

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

Summer Institute for Statistical Genetics – 2016

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

- 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

- 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

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,
  - $X \perp\!\!\!\perp Y \mid Z$ iff $P(X = x, Y = y|Z = z) = P(X = x|Z = z)P(Y = y|Z = z)$
  - If X and Y are neighbors ($X - Y$), they are conditionally dependent
  - X is conditionally independent of all other nodes, given $\mathrm{neighbors(X)}$: $Z \notin \mathrm{neighbors(X)}$, then $X \perp\!\!\!\perp Z \mid \mathrm{neighbors(X)}$

Nonparanormal (Gaussian Copula) Models

- Suppose $X \nsim N(0, \Sigma)$, but there exists monotone functions $f_j, j = 1, \dots p$ such that $[f_1(X_1), \dots f_p(X_p)] \sim N(0, \Sigma)$
  - We say that X has a nonparanormal distribution $X \sim NPN_p(f, \Sigma)$.
  - f and Σ 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 nonparanomal network:
  - i) transform the data: $[f_1(X_1), \dots f_p(X_p)]$
  - 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_j^i be the rank of x_j^i among $x_j^1, \dots, x_j^n$ and $\bar{r}_j = (n+1)/2$ be the average rank
  - Calculate Spearman's ρ or Kendall's τ

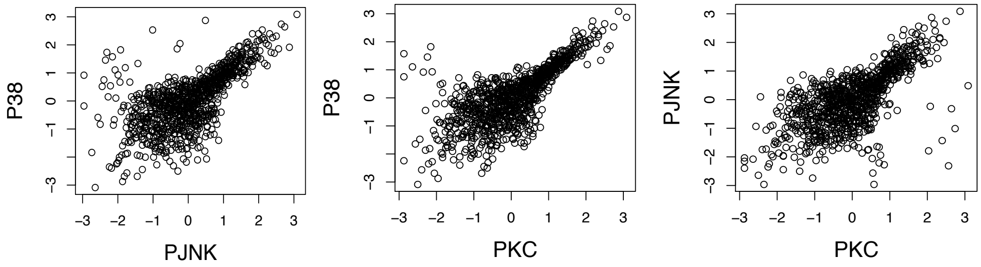
$$\hat{\rho}_{jk} = \frac{\sum_{i=1}^{n} (r_j^i - \bar{r}_j)(r_k^i - \bar{r}_k)}{\sqrt{\sum_{i=1}^{n} (r_j^i - \bar{r}_j)^2 \sum_{i=1}^{n} (r_k^i - \bar{r}_k)^2}}$$

$$\hat{\tau}_{jk} = \frac{2}{n(n-1)} \sum_{1 \le i < i' \le n} \operatorname{sign}\left((x_j^i - x_j^{i'})(x_k^i - x_k^{i'})\right)$$

- 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 Σ^{-1} by plugging in rank-based correlations into graphical lasso (R-package `huge`)

A Real Data Example

- Protein cytometry data for cell signaling data (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
- 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

- 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 $(j,k) \in E$, ϕ_{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

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

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

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

Graphical Models for Binary Random Variables

- We can consider a neighborhood selection[2] approach with an ℓ_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)

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

  - $f(\theta; x) = \log\left\{\exp\left(\sum_{k\in -j} \theta_{jk} x_k\right) + \exp\left(-\sum_{k\in -j} \theta_{jk} x_k\right)\right\}$
- It turns out this is equivalent to solving p penalized logistic regression problems, which is rather easy (R-package `glmnet`)

[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 `glmnet` package[3]
    - We can even learn networks with multiple types of nodes (gene expression, SNPs, and CNVs)[4]

[3] Yang et al (2012)

[4] Yang et al (2014), Chen et al (2015)

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 $E \subset V \times V$ that satisfy $\forall j \in V$, $\exists N(j)$ such that:

$$p_j(X_j \mid \{X_k, k \neq j\}) = p_j(X_j \mid \{X_k : k \in N(j)\}) = p_j(X_j \mid \{X_k : (k,j) \in E\})$$

- $N(j)$ is the minimal set of variables on which the conditional densities depend

Estimating Conditional Independencies

Question: how to condition?

- Approach 1: Estimate the joint density $f(X_1, \dots, X_p)$; 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!
- Approach 2: Estimate the conditionals directly $f_j(X_j \mid X_{-j})$
  - 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):

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

- 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[5]

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

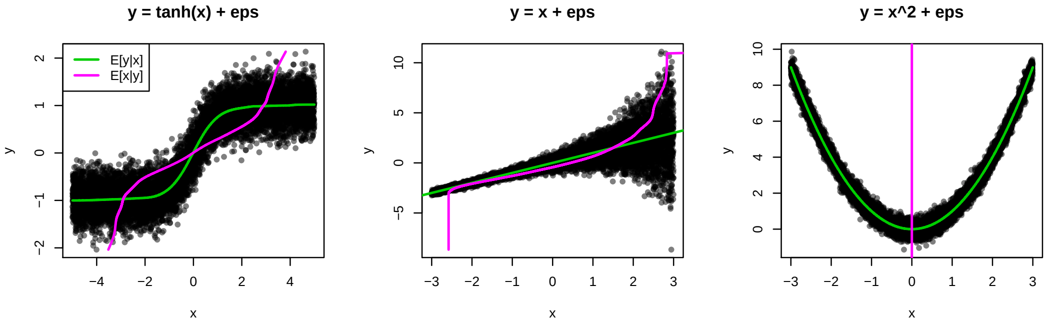
$$\underset{f_{jk}\in\mathcal{F}}{\text{minimize}}\ \frac{1}{2n}\sum_{j=1}^{p}\|\vec{x}_j-\sum_{k\neq j}f_{jk}(\vec{x}_k)\|_2^2+\lambda\sum_{k>j}\left(\|f_{jk}(\vec{x}_k)\|_2^2+\|f_{kj}(\vec{x}_j)\|_2^2\right)^{1/2}$$

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

[5] Voorman et al (2014) *Biometrika*, R-package `spacejam`

SpaCE JAM

Estimating f_{jk} and f_{kj} seems redundant...



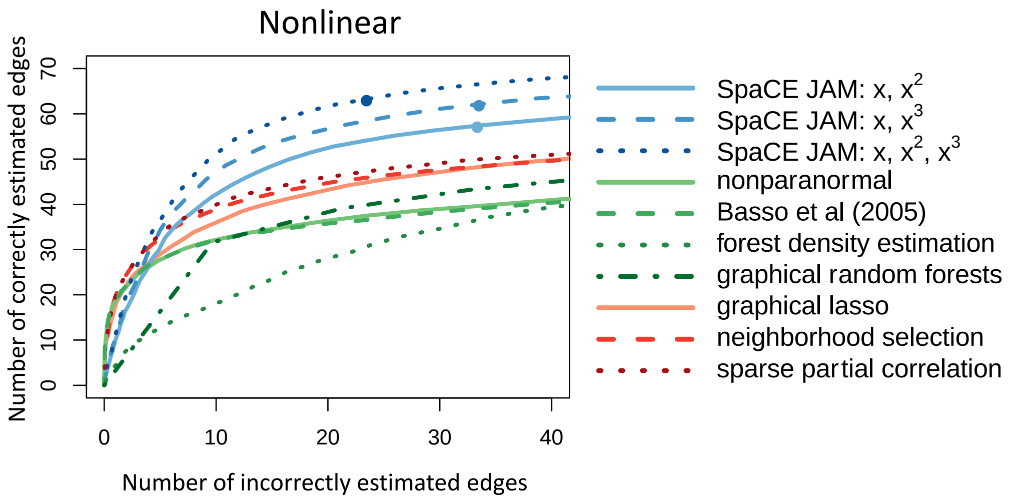
but necessary for non-linear functions

Other Flexible Procedures

- Forest density estimation (Liu et al, 2011) assumes that 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
  - They consider conditional dependencies through conditional mean
  - They allow for general random variables, discrete or continuous
  - Use a random forest to estimate $E[X_j \mid X_{\setminus j}]$ non-parametrically
  - Theoretical properties have not yet been justified

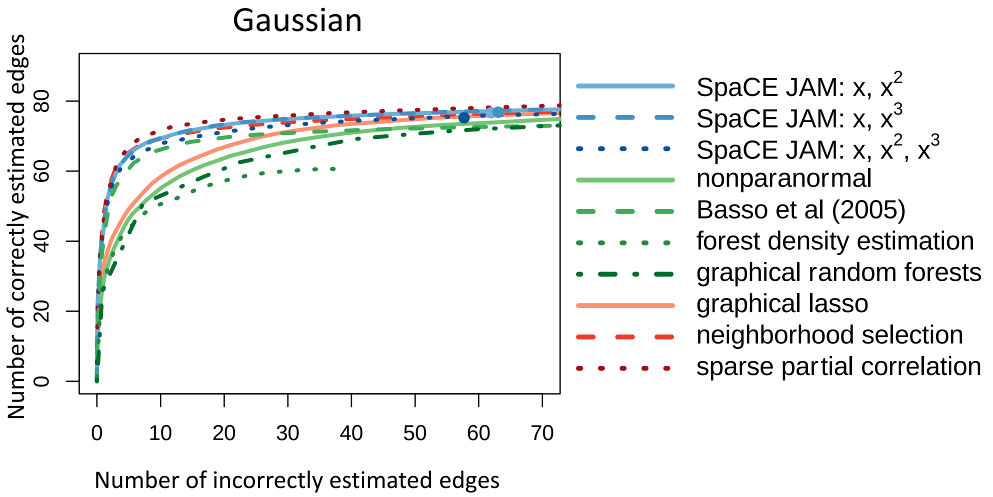
Comparison on Simulated Data

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

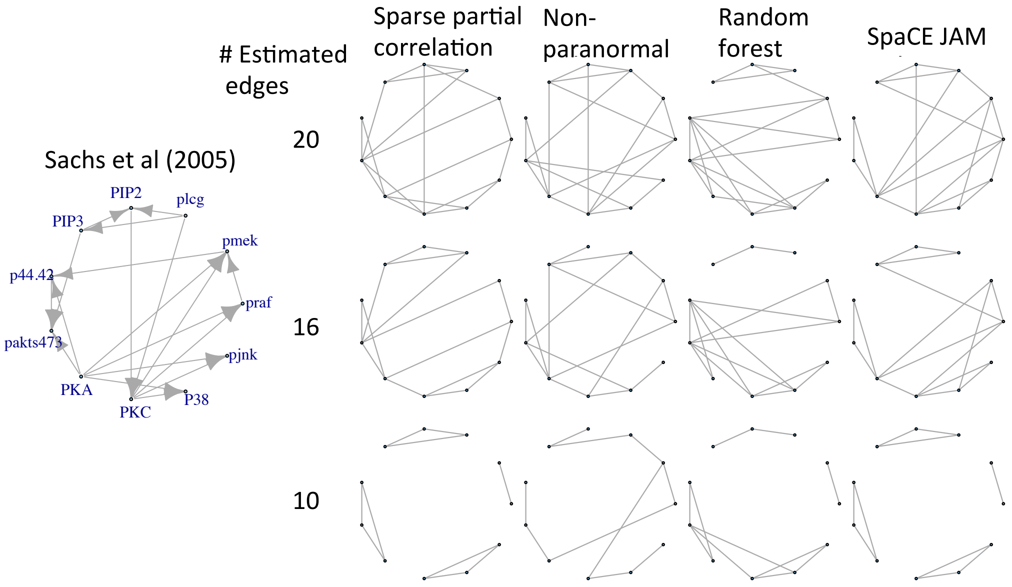


Comparison on Simulated Data

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



Estimation of Cell Signaling Network



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
- 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, suited for functional data, to enforce "symmetry" in terms of edge selection

Pathway & Network Analysis of Omics Data: Bayesian Networks – Basic Concepts

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

Summer Institute for Statistical Genetics – 2016

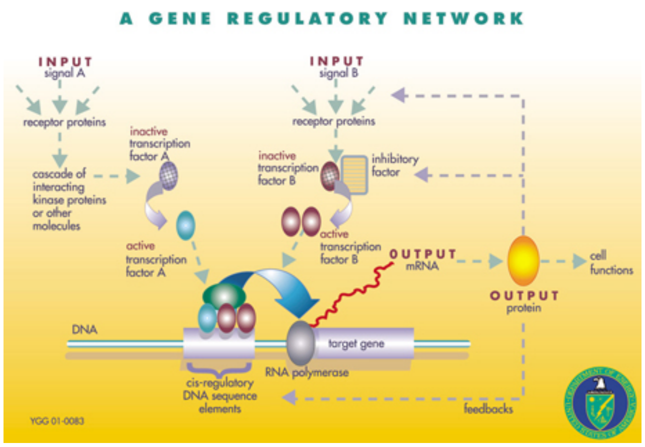
Bayesian Networks

- 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?

Many biological networks include directed edges:

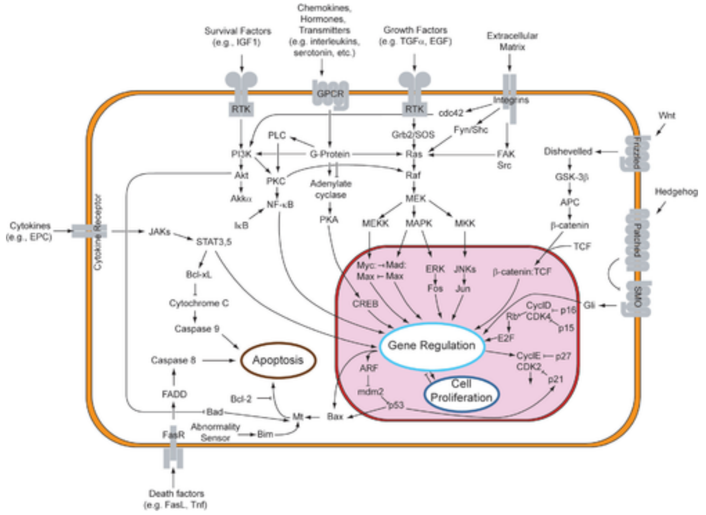
- In gene regulatory networks, protein products of transcription factors can alter the expression of target genes, but the target genes (usually) don't have a direct effect on the expression of transcription factors


A GENE REGULATORY NETWORK
INPUT
signal A
receptor proteins
cascade of interacting kinase proteins or other molecules
inactive transcription factor A
active transcription factor A
INPUT
signal B
receptor proteins
inactive transcription factor B
inhibitory factor
active transcription factor B
OUTPUT
mRNA
OUTPUT
protein
cell functions
DNA
target gene
RNA polymerase
cis-regulatory DNA sequence elements
feedbacks
YGG 01-0083

Why Bayesian Networks?

Many biological networks include directed edges:

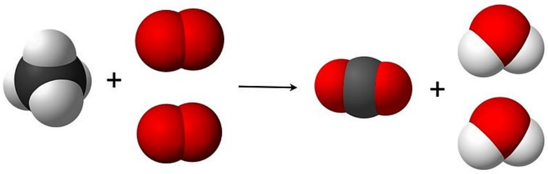
- In cell signaling networks, the signal from the cell's environment is transducted into the cell, and results e.g. in (global) changes in gene expression, but gene expression may not affect the environmental factors



Why Bayesian Networks?

Many biological networks include directed edges:

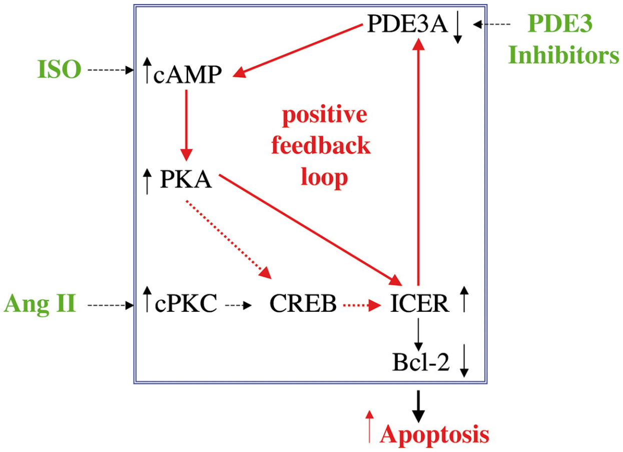
- Biochemical reactions in metabolic networks, may not reversible, and in that case, one metabolite may affect the other, but the relationship is ont reciprocated



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)



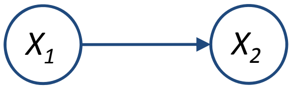
which make estimation even more difficult!

What's the Difference?

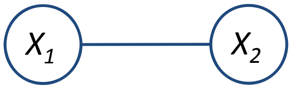
- Bayesian networks are widely used to model causal relationships between variables.
- Undirected networks (e.g. GGM) provide information about associations among variables; while this greatly helps in the study of biological systems, in some cases, they are not enough (e.g. drug development).
- The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



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



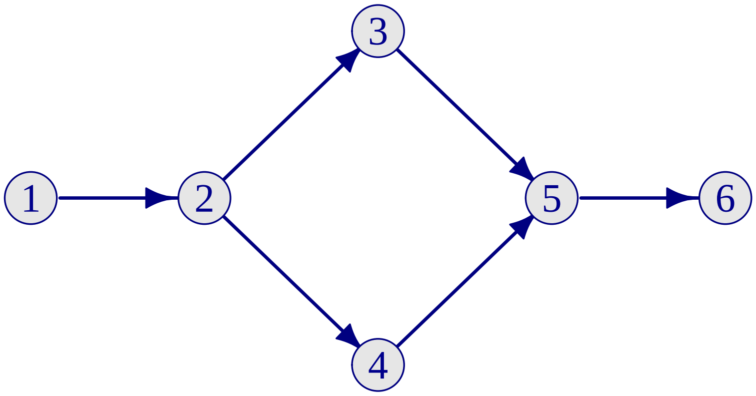
Outline

- 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
- The parents of node j are $\{k : k \to j\}$, we denote this by pa_j or $\mathrm{pa}(j)$
- The children of node j are $\{k : j \to k\}$
- Two vertices connected by an edge are called adjacent

Directed Graphs: Some Terminology

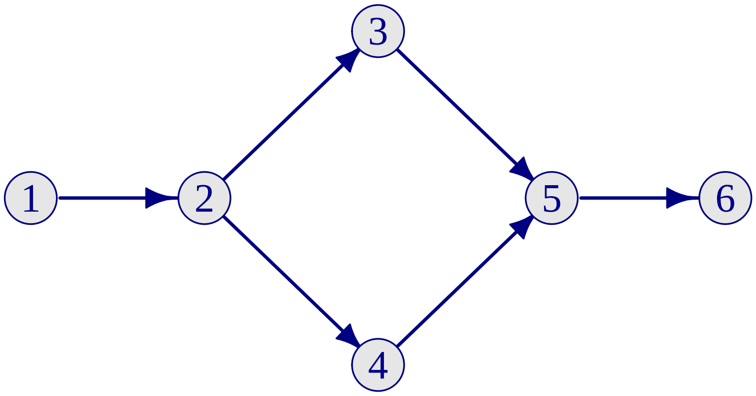


- $\mathrm{pa}(1) = \emptyset$, $\mathrm{pa}(2) = 1$, $\mathrm{pa}(3) = \mathrm{pa}(4) = \{2\}$, $\mathrm{pa}(5) = \{3, 4\}$
- What are children of $\{1, \ldots 5\}$?

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
- i is an ancestor of j if there is a directed path of length ≥ 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 $\{1, \dots 5\}$?

Directed Graphs: Some Terminology

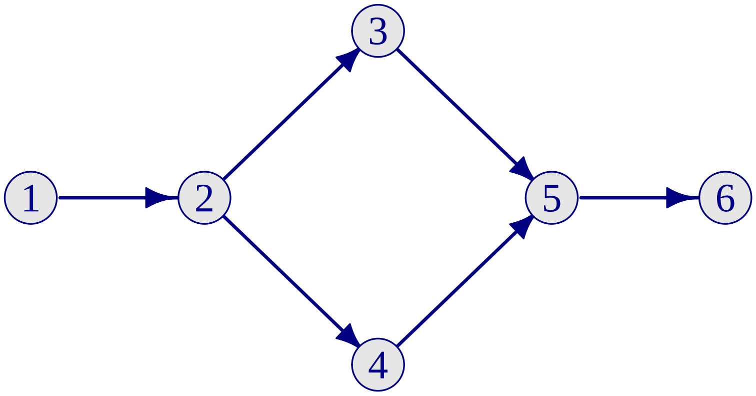
An important concept in DAGs is that of colliders (aka "inverted forks"):

- 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

$$i \dots \to k \leftarrow \dots j$$

- k is an non-collider if it is not an end-point, and is not a collider on a path:
  - $i \dots \leftarrow k \leftarrow \dots j$
  - $i \dots \to k \to \dots j$
  - $i \dots \leftarrow k \to \dots j$
- 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



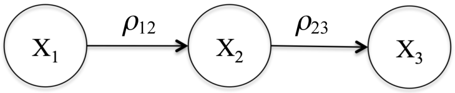
- What are the colliders on paths between 1&4, 3&4, 2&6?
- What are the non-colliders on paths between 1&4, 3&4, 2&6?

Factorization of Probability Distributions over DAGs

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

$$\begin{aligned} P(X_1, X_2, X_3) &= P(X_1 \mid X_2, X_3) P(X_2 | X_3) P(X_3) \\ &= P(X_3 \mid X_1, X_2) P(X_2 | X_1) P(X_1) \\ &= \cdots \end{aligned}$$

- Now, consider this simple DAG



- Then, the probability distribution can be factorized as

$$P(X_1, X_2, X_3) = P(X_3 \mid X_2) P(X_2 | X_1) P(X_1)$$

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

$$P(V) = \prod_{j \in V} P(X_j \mid \mathrm{pa}_j)$$

- Compare this with the general probability decomposition

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

- This means that on DAGs we have

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

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

Independence (unconditional)

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

Conditional Independence

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

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

  ii) $P(X = x|Y = y, Z = z) = P(X = x|Z = z)$ (is symmetric)

- We also have,

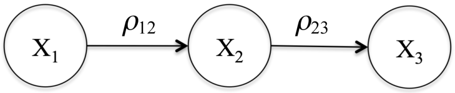
$$P(X = x, Y = y, Z = z) = \frac{P(X = x, Z = z)P(Y = y, Z = z)}{P(Z = z)}.$$

- Intuitively, if $X \perp\!\!\!\perp Y$ then if Z is known, knowledge of X provides no information about Y.
- These can be generalized for vectors.

Conditional Independence

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

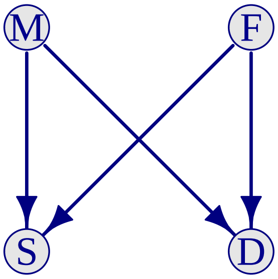
The Toy Example, Revisited



- 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 $X_3 \perp\!\!\!\perp X_1 | X_2$ (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

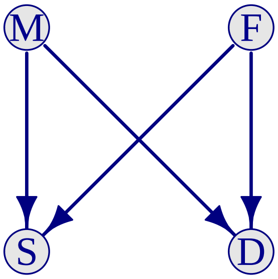
Consider an example from population genetics:



- We have genetic information for *M*other, *F*ather, *D*aughter and *S*on 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

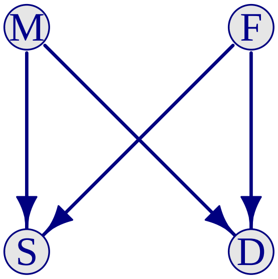
Consider an example from population genetics:



- Now, it is natural to assume that given the parents' genetic information, the genotypes of Son and Daughter are independent $\Rightarrow S \perp\!\!\!\perp D \mid \{M, F\}$

An example from genetics

Consider an example from population genetics:



- Also, one can assume independence among genotypes of M and $F \Rightarrow M \perp\!\!\!\perp F$
- However, if we know that e.g. Son has Aa, and Mother has aa, then Father should have Aa or AA $\Rightarrow M \not\perp\!\!\!\perp F | S$

d-separation

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

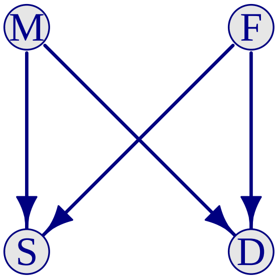
1. π includes a chain $i \to m \to j$ or a fork $i \leftarrow m \to j$ such that the middle note is in Z, or
2. π contains a collider (or inverted fork) $i \to m \leftarrow j$ such that neither the middle node m nor its descendants are NOT in Z.

How is this used?

- If i and j are d-separated given Z, then $X_i \perp\!\!\!\perp X_j | Z$ for any probability distribution P factorizing according to G
- If i and j are d-separated given $\emptyset$, then $X_i \perp\!\!\!\perp X_j$ for any probability distribution P factorizing according to G

Genetics example, revisited

Consider an example from population genetics:



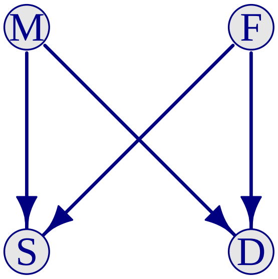
- $\{M, F\}$ block all paths from S to $D \Rightarrow D \perp\!\!\!\perp S \mid \{M, F\}$
- Is $M \perp\!\!\!\perp F$?
- Is $M \perp\!\!\!\perp F \mid \{S, D\}, \mid S, \mid D$?

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, $X_i \perp\!\!\!\perp X_j | Z$ iff Z separates i and j in $\tilde{G}$

Genetics example, revisited (again)

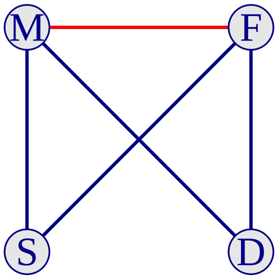
Consider an example from population genetics:



- Is $S \perp\!\!\!\perp D \mid \{M, F\}$
- Is $M \perp\!\!\!\perp F$?
- Is $M \perp\!\!\!\perp F \mid \{S, D\}, \mid S, \mid D$?

Genetics example, revisited (again)

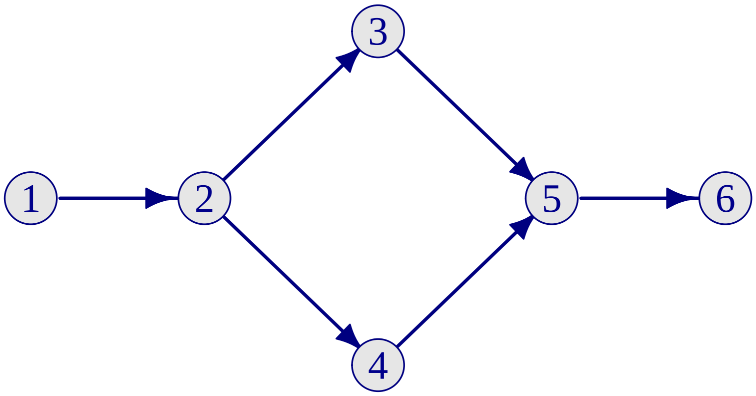
Consider an example from population genetics:



- Is $S \perp\!\!\!\perp D \mid \{M, F\}$
- Is $M \perp\!\!\!\perp F$?
- Is $M \perp\!\!\!\perp F \mid \{S, D\}, \mid S, \mid D$?

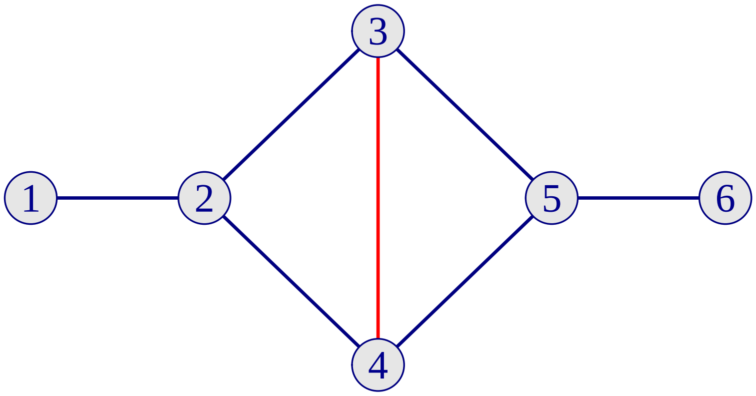
A More Complex Example

What are conditional independence relations in this graph?



A More Complex Example

What are conditional independence relations in this graph?



Structural Equation Models

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

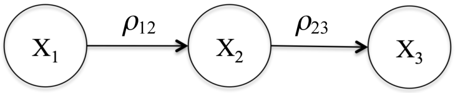
$$X_j = f_j(pa_j, \gamma_j), \quad j = 1, \dots, p$$

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

$$X_j = \sum_{j' \in pa_j} \rho_{jj'} X_{j'} + \gamma_j, \quad j = 1, \dots, p$$

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

A Toy Example



Assuming normality we can write:

$$\begin{aligned} X_1 &= \gamma_1 \\ X_2 &= \rho_{12} X_1 + \gamma_2 = \rho_{12}\gamma_1 + \gamma_2 \\ X_3 &= \rho_{23} X_2 + \gamma_3 = \rho_{23}\rho_{12}\gamma_1 + \rho_{23}\gamma_2 + \gamma_3 \end{aligned}$$

For non-Gaussian variables, these equations will involve non-linear relationships.

Pathway & Network Analysis of Omics Data: Bayesian Networks – Estimation from Observational Data

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

Summer Institute for Statistical Genetics – 2016

Estimation of DAGs in Biological Settings

- Estimation of DAGs is (in general) computationally very hard (in fact, it's NP-hard): there are $\sim 2^{p^2}$ 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 $X \perp\!\!\!\perp Y \mid Z$)
  - 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 $X_j \perp\!\!\!\perp X_{j'} \mid S$
  - 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)

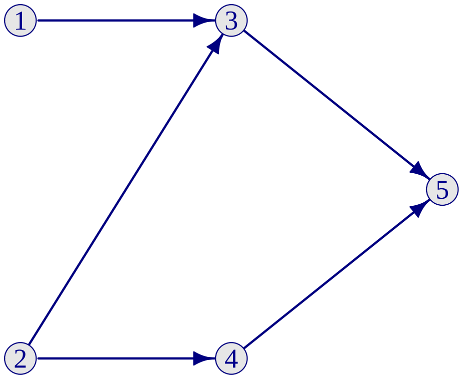
PC Algorithm (Spirtes et al, 1993)

- One of the first algorithms for learning structure of DAGs
- Efficient implementations that allow for learning DAG structures with p up to ~ 1000
  - `R`-package `pcalg` (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, $j - j'$

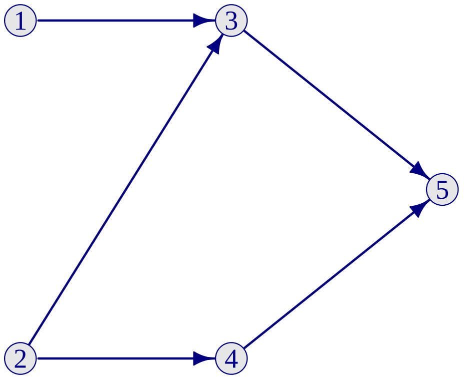
PC Algorithm (Spirtes et al, 1993)

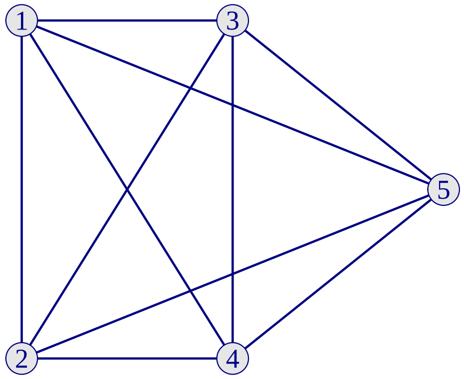
```
Start with a complete undirected graph, and set $i = 0$
Repeat
    ▸ For each $j \in V$
    ▸ For each $j' \in \mathrm{ne}(j)$
    ▸ Determine if $\exists S \subset \mathrm{ne}(j) \backslash \{j'\}$ with $|S| = i$
        ▸ Test for CI: is $X_j \perp\!\!\!\perp X_{j'} \mid S$?
        ▸ If such an $S$ exists, then set $S_{jj'} = S$, remove $j - j'$ edge
    ▸ $i = i + 1$
Until $|\mathrm{ne}(j)| < i$ for all $j$
```

Example

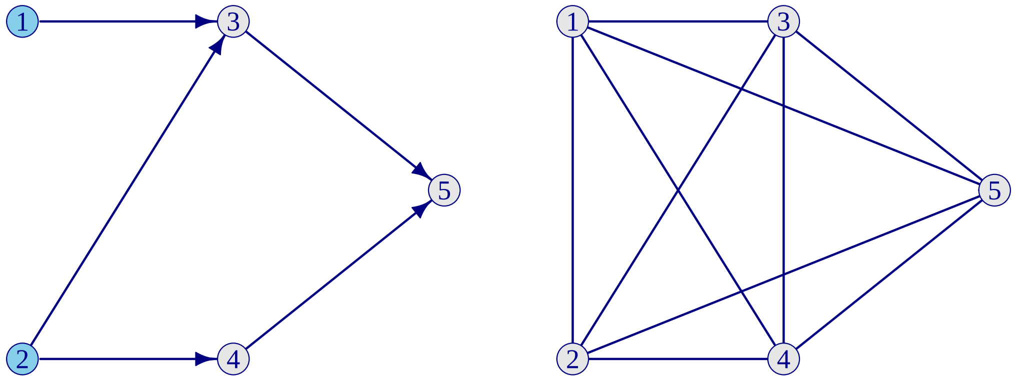


Example



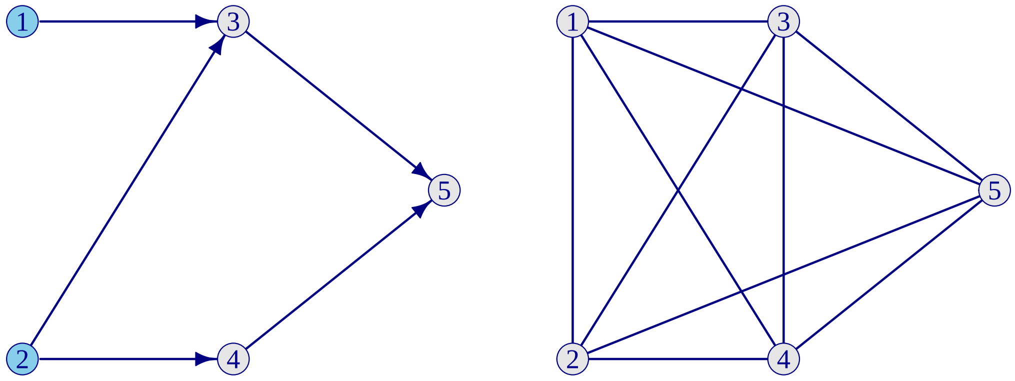


Example



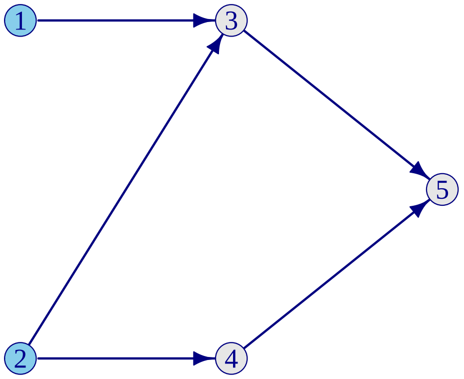
$i = 0$

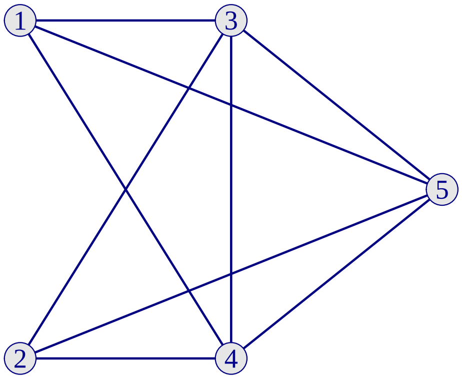
Example



$i = 0$ $S_{1,2} = \emptyset$

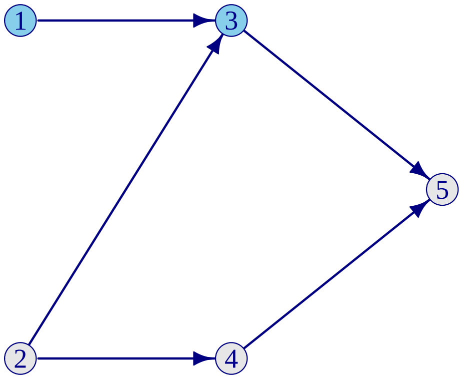
Example

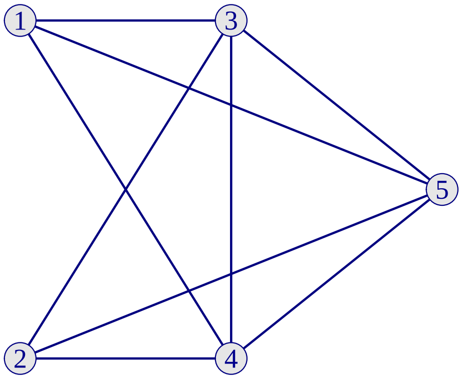




$i = 0$ $S_{1,2} = \emptyset$

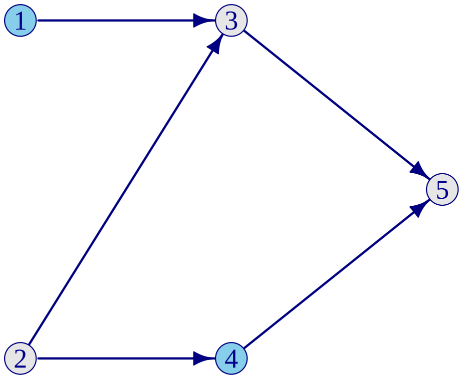
Example

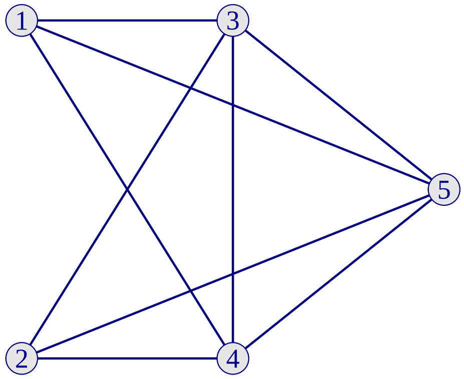




$i = 0$ $S_{1,2} = \emptyset$

Example

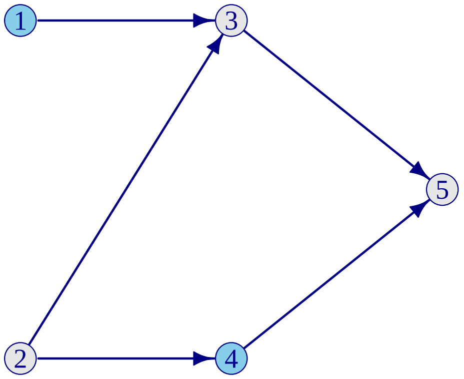


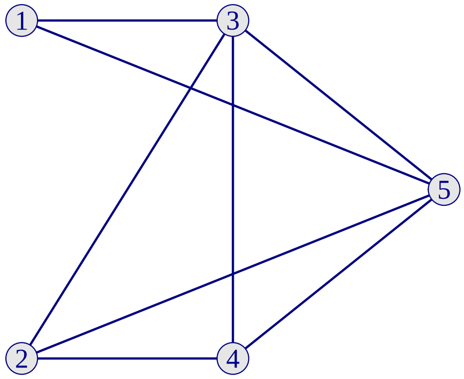


$i = 0$ $\quad S_{1,2} = \emptyset$

$S_{1,4} = \emptyset$

Example

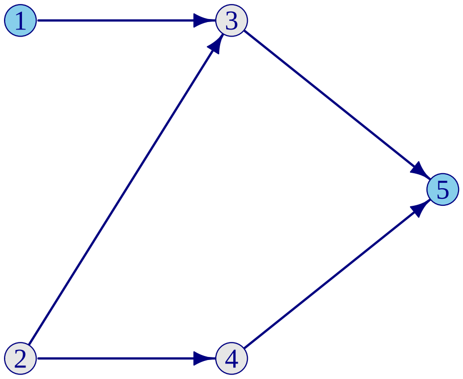


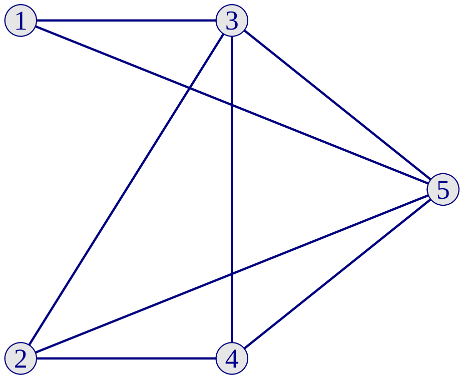


$i = 0$ $\quad S_{1,2} = \emptyset$

$S_{1,4} = \emptyset$

Example

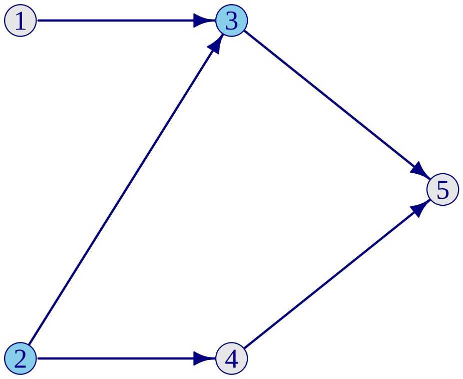


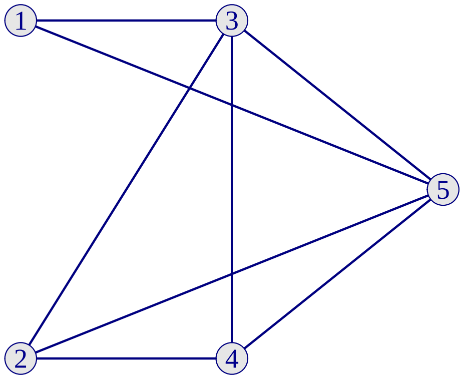


$i = 0$ $\quad S_{1,2} = \emptyset$

$S_{1,4} = \emptyset$

Example

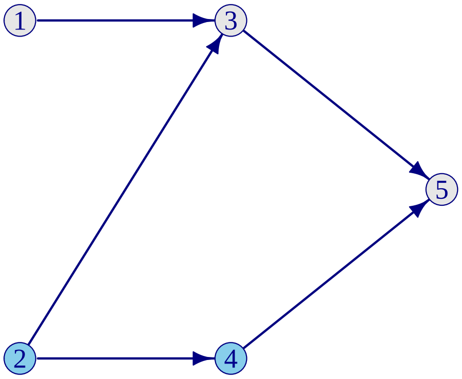


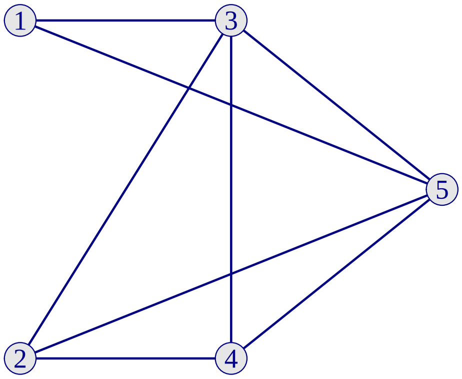


$i = 0$ $\quad S_{1,2} = \emptyset$

$S_{1,4} = \emptyset$

Example

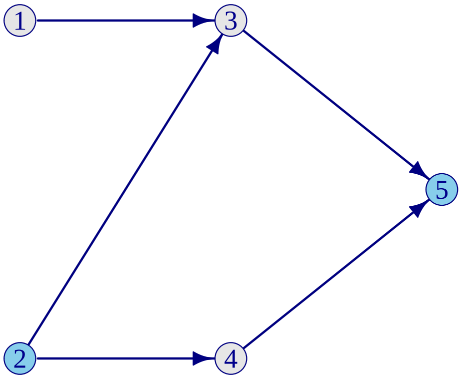


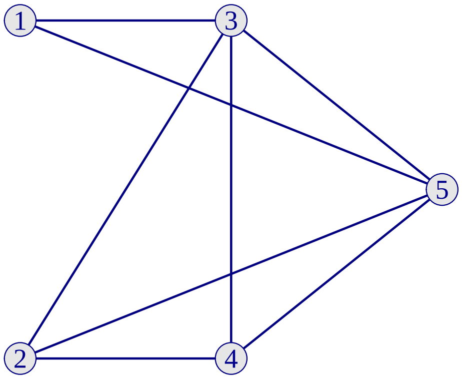


$i = 0$ $S_{1,2} = \emptyset$

$S_{1,4} = \emptyset$

Example

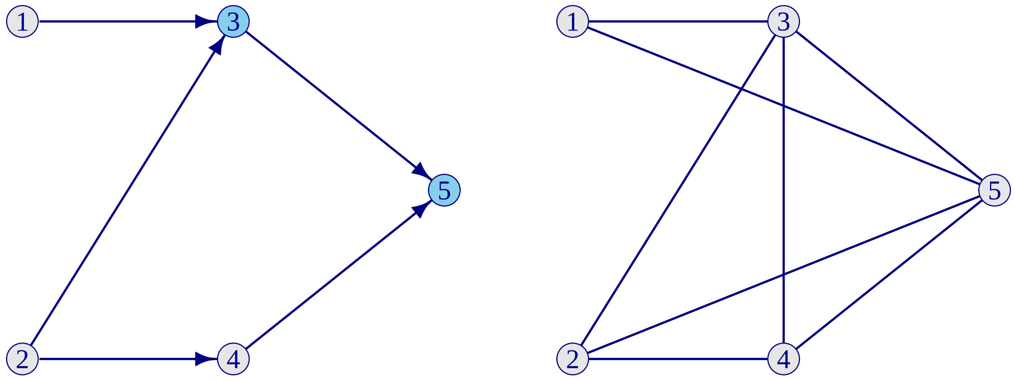




$i = 0$ $S_{1,2} = \emptyset$

$S_{1,4} = \emptyset$

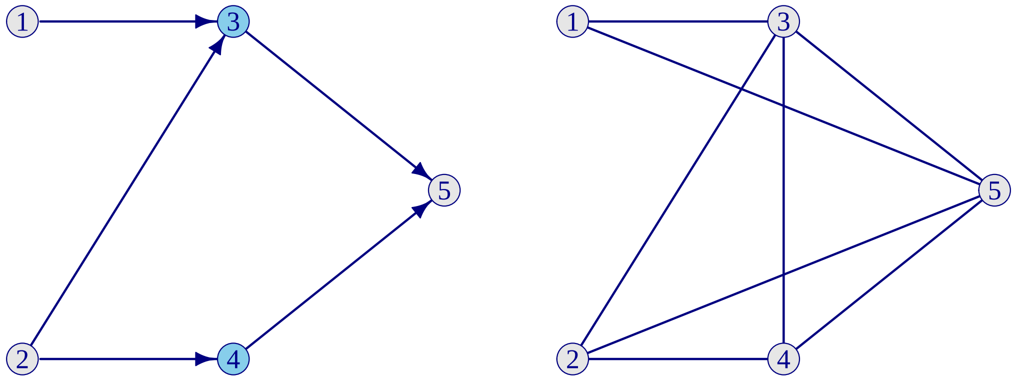
Example



$\color{red}{i = 0}$ $\quad S_{1,2} = \emptyset$

$S_{1,4} = \emptyset$

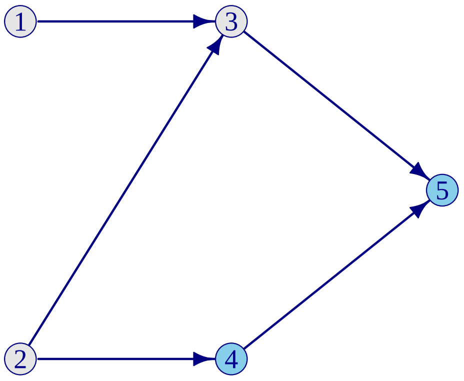
Example

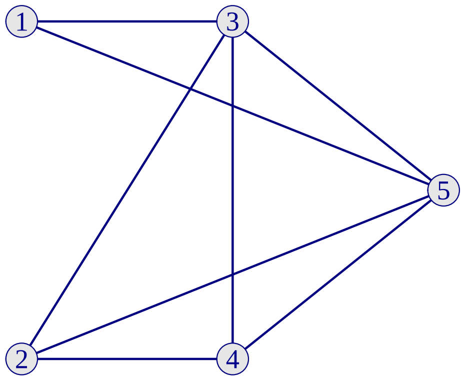


$\color{red}{i = 0}$ $\quad S_{1,2} = \emptyset$

$S_{1,4} = \emptyset$

Example

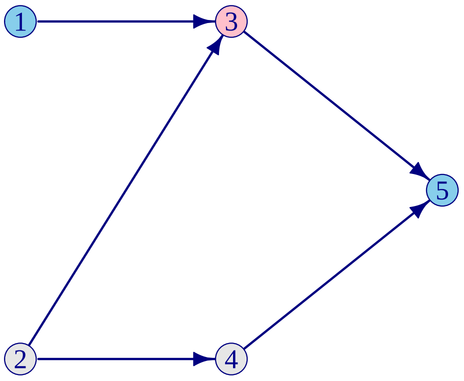


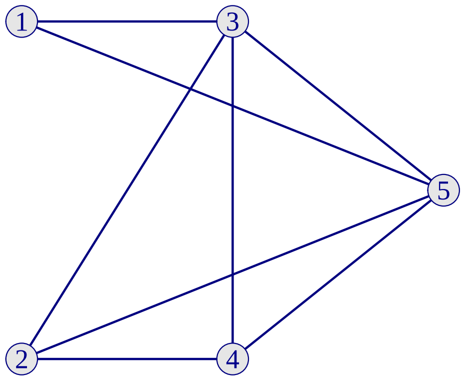


$i = 0$ $\quad S_{1,2} = \emptyset$

$S_{1,4} = \emptyset$

Example



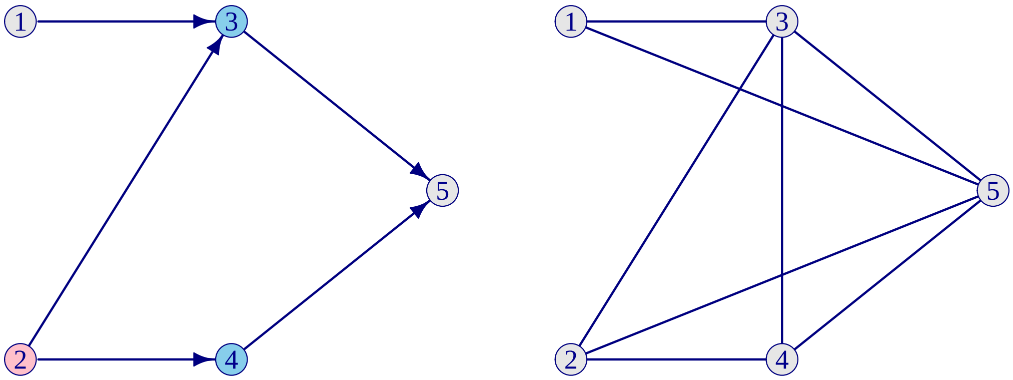


$i = 0$ $\quad S_{1,2} = \emptyset$

$S_{1,4} = \emptyset$

$i = 1$

Example

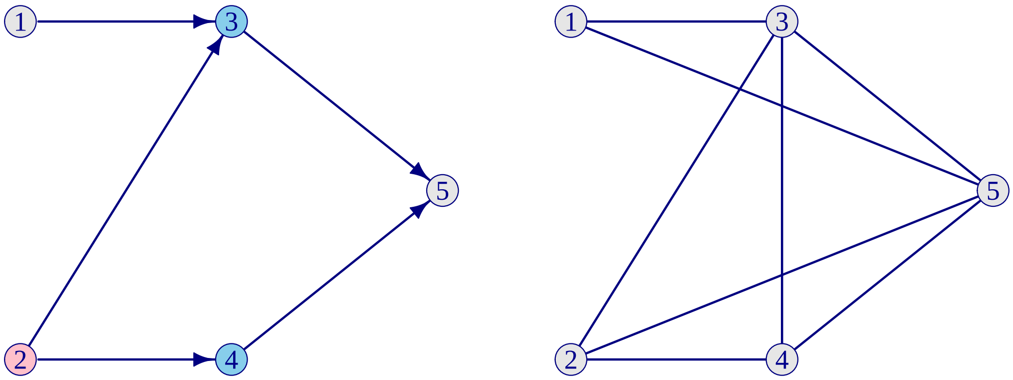


$i = 0$ $\quad S_{1,2} = \emptyset$

$\qquad\quad S_{1,4} = \emptyset$

$i = 1$

Example

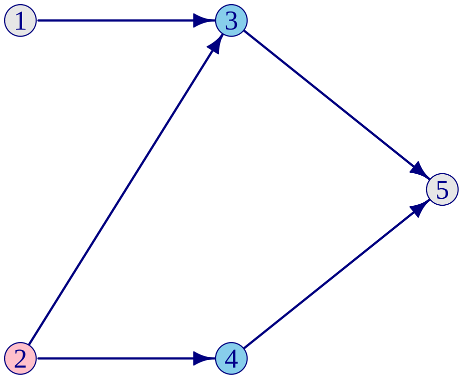


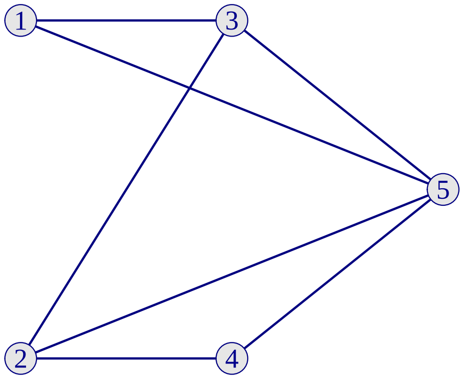
$i = 0$ $\quad S_{1,2} = \emptyset$

$\qquad\quad S_{1,4} = \emptyset$

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

Example



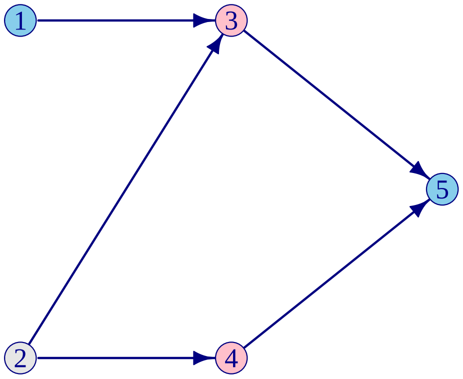


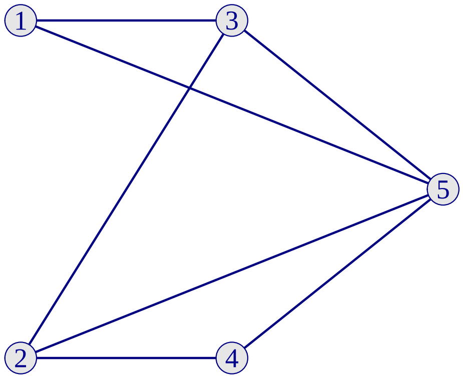
$i = 0$ $\quad S_{1,2} = \emptyset$

$\quad S_{1,4} = \emptyset$

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

Example





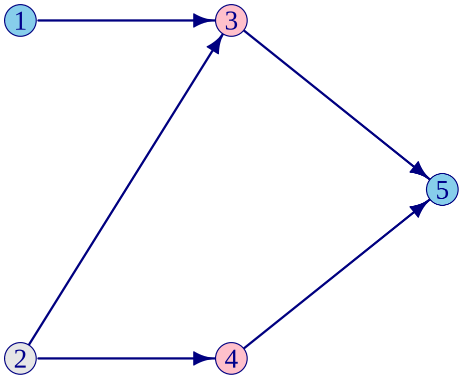
$i = 0$ $\quad S_{1,2} = \emptyset$

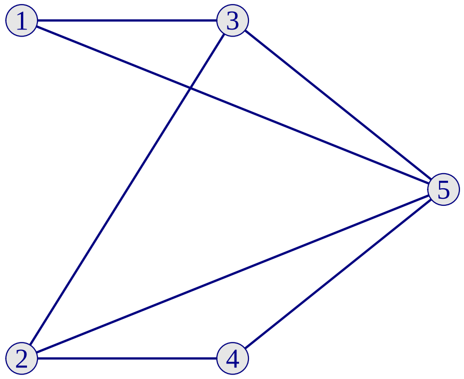
$\quad S_{1,4} = \emptyset$

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

$i = 2$

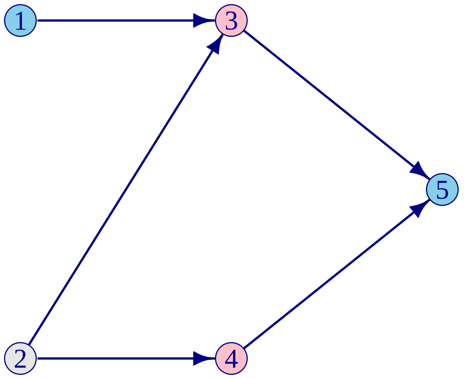
Example

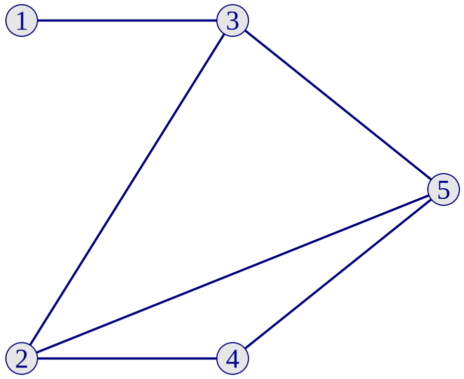




$$\begin{array}{ll} i = 0 & S_{1,2} = \emptyset \\ & S_{1,4} = \emptyset \\ i = 1 & S_{3,4} = \{2\} \\ i = 2 & S_{1,5} = \{3, 4\} \end{array}$$

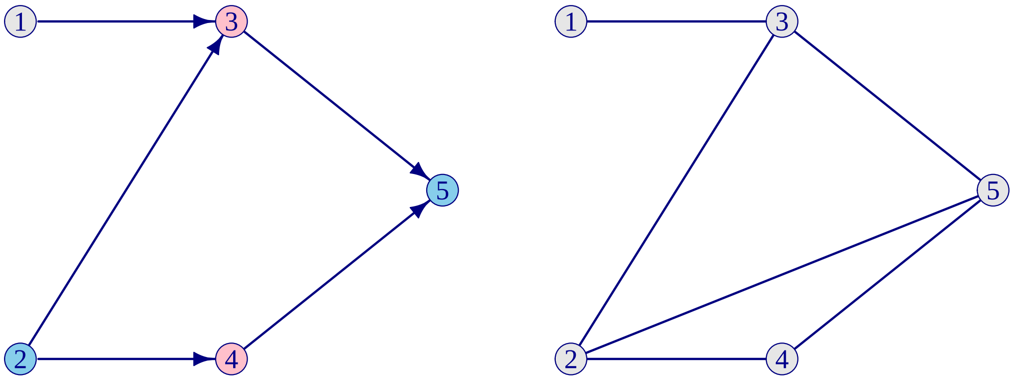
Example





$$\begin{array}{ll} i = 0 & S_{1,2} = \emptyset \\ & S_{1,4} = \emptyset \\ i = 1 & S_{3,4} = \{2\} \\ i = 2 & S_{1,5} = \{3, 4\} \end{array}$$

Example



$i = 0$ $\quad S_{1,2} = \emptyset$

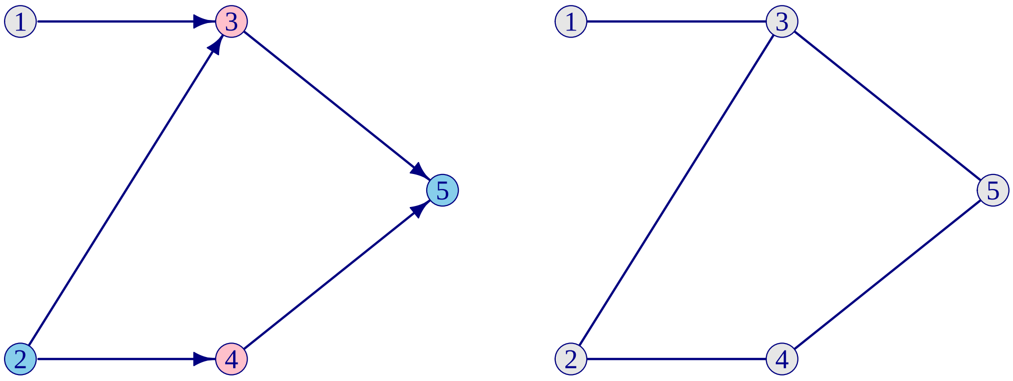
$\qquad\quad S_{1,4} = \emptyset$

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

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

$\qquad\quad S_{2,5} = \{3, 4\}$

Example



$i = 0$ $\quad S_{1,2} = \emptyset$

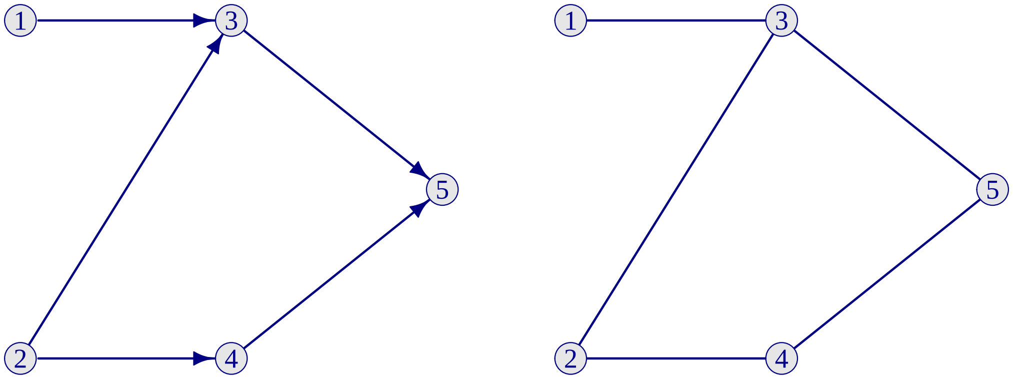
$\qquad\quad S_{1,4} = \emptyset$

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

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

$\qquad\quad S_{2,5} = \{3, 4\}$

Example



$i = 0$ $S_{1,2} = \emptyset$

$S_{1,4} = \emptyset$

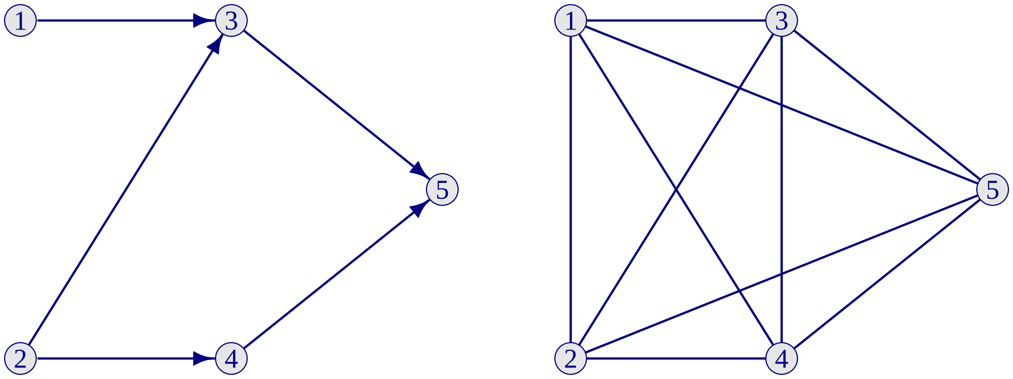
$i = 1$ $S_{3,4} = \{2\}$

$i = 2$ $S_{1,5} = \{3, 4\}$

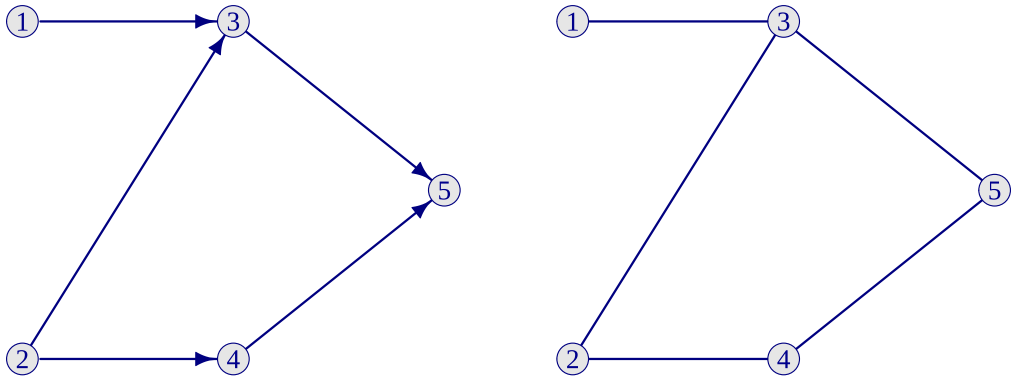
$S_{2,5} = \{3, 4\}$

$i = 3$ STOP $(|\text{ne}_j| < 3 \forall j)$

Example



Example



$i = 0$ $S_{1,2} = \emptyset$

$S_{1,4} = \emptyset$

$i = 1$ $S_{3,4} = \{2\}$

$i = 2$ $S_{1,5} = \{3, 4\}$

$S_{2,5} = \{3, 4\}$

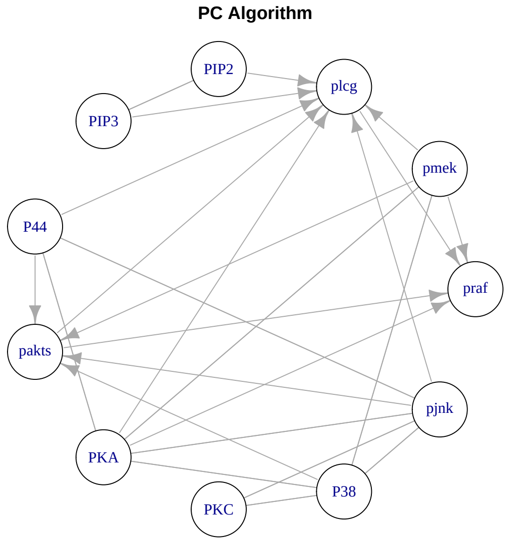
$i = 3$ STOP $(|\mathrm{ne}_j| < 3 \;\; \forall j)$

Analysis of Protein Flow Cytometry using pcalg

```
> 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')
```

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

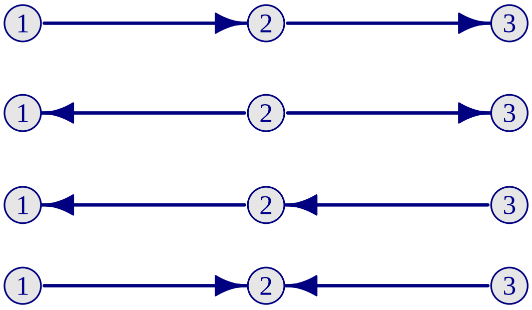
Analysis of Protein Flow Cytometry using `pcalg`



But wait, where did the directions come from? And why are only some of the edges directed?

Markov Equivalence

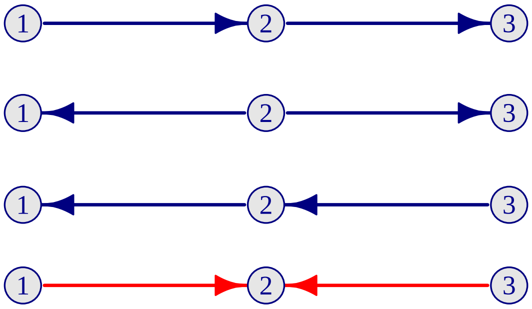
Consider the following 4 graphs



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

Markov Equivalence

Consider the following 4 graphs

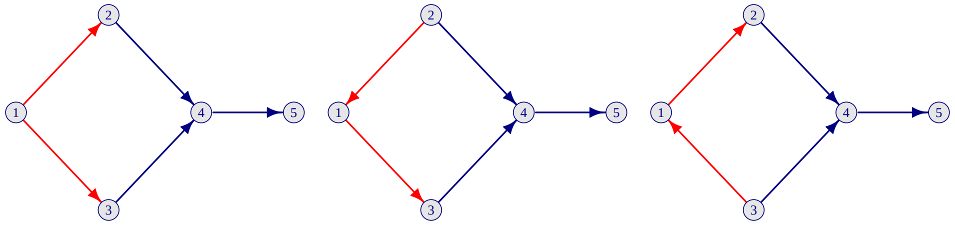


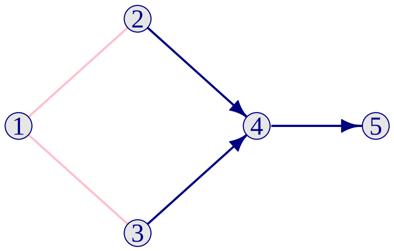
In the first 3 graphs, $X_1 \perp\!\!\!\perp X_3 \mid X_2$?
Two graphs that imply the same CI relationships via d-separation are called Markov equivalent

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

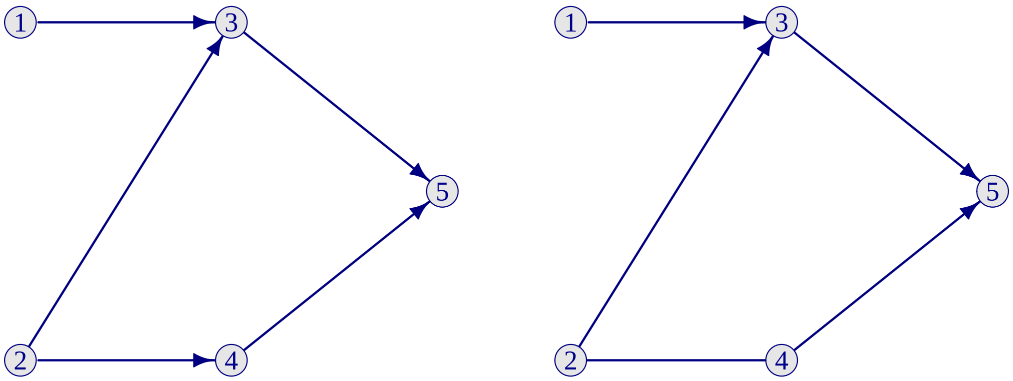




Finding Partial Directions in DAGs

- 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



$i = 0$ $\quad S_{1,2} = \emptyset$

$\quad S_{1,4} = \emptyset$

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

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

$\quad S_{2,5} = \{3, 4\}$

The `bnlearn` package

- There are a number of `R`-packages for learning the structure of DAGs, including `pclag`, `bnlearn`, `deal`
- `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

```
> 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 α (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

```
> 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
```

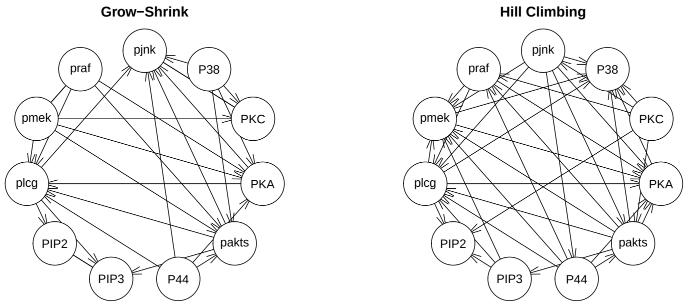
Analysis of Protein Flow Cytometry using bnlearn

```
> dag2
  Bayesian network learned via Score-based methods

  model:
   [PKC][pjnk|PKC][P44|pjnk][pakts|P44:PKC:pjnk][praf|P44:pakts:PKC][PIP3|pakts
   [plcg|praf:PIP3:P44:pakts:pjnk][pmek|praf:plcg:PIP3:P44:pakts:pjnk]
   [PIP2|plcg:PIP3:PKC][PKA|praf:pmek:plcg:P44:pakts:pjnk]
   [P38|pmek:plcg:pakts:PKA:PKC:pjnk]
  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 (Gaussia
  penalization coefficient:              4.459057
  tests used in the learning procedure:  505
  optimized:                             TRUE
```

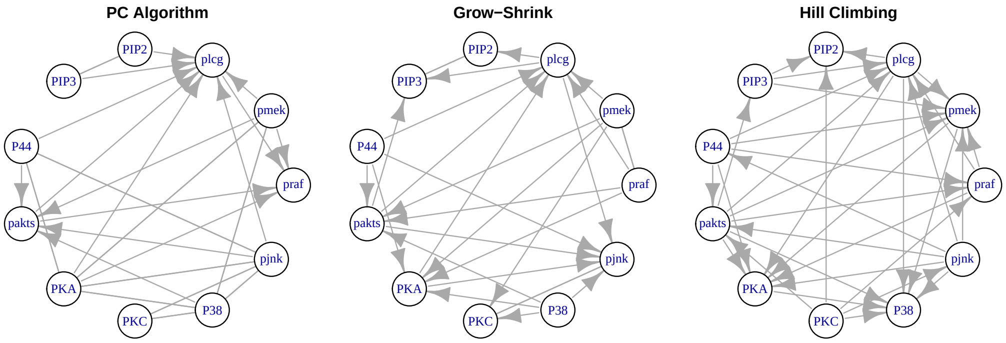
Analysis of Protein Flow Cytometry using bnlearn



The two graphs are quite different

```
> compare(dag1,dag3)
$tp
[1] 9
$fp
[1] 26
$fn
[1] 17
```

Comparison of Results for Protein Flow Cytometry Data



- 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

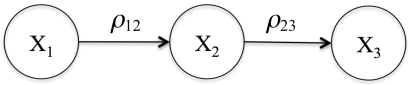
Penalized Likelihood Estimation of DAGs

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

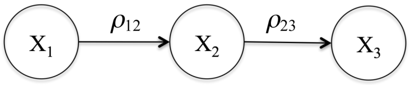
$$X_i = f_i(\mathrm{pa}_i, \gamma_i), \qquad i = 1, \cdots, p$$

- And, for Gaussian random variables, we can write

$$X_i = \sum_{j \in \mathrm{pa}_i} \rho_{ji} X_j + \gamma_i, \qquad i = 1, \cdots, p$$



Penalized Likelihood Estimation of DAGs



$$\begin{aligned} X_1 &= \gamma_1 \\ X_2 &= \rho_{12} X_1 + \gamma_2 = \rho_{12}\gamma_1 + \gamma_2 \\ X_3 &= \rho_{23} X_2 + \gamma_3 = \rho_{23}\rho_{12}\gamma_1 + \rho_{23}\gamma_2 + \gamma_3 \end{aligned}$$

Thus $X = \Lambda\gamma$ where

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

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:

$$\hat{A} = \underset{A \in \mathcal{A}}{\arg\min} \left\{ \mathrm{tr}\left[(I - A)^{\mathsf{T}} (I - A) S \right] \right\}$$

[1] Shojaie & Michailidis (2010)

Penalized Likelihood Estimation of DAGs

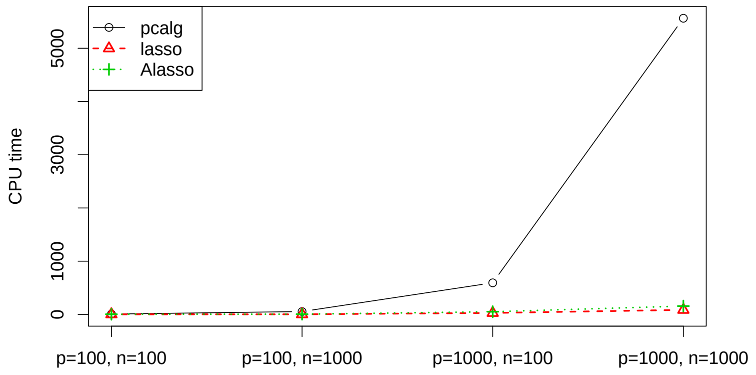
- In high dimensions, we can solve a penalized version of this problem, e.g. by adding a lasso penalty $\lambda \sum_{i<j} |A_{ij}|$
- It turns out that, the problem can be reformulated as $(p-1)$ lasso problems, where we regress each variable, on those appearing earlier in the ordering:

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

- As in `glasso`, λ 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 α

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



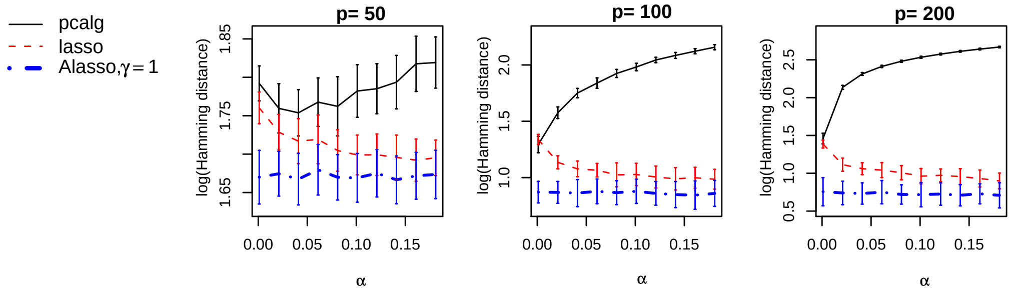
Simulation Studies

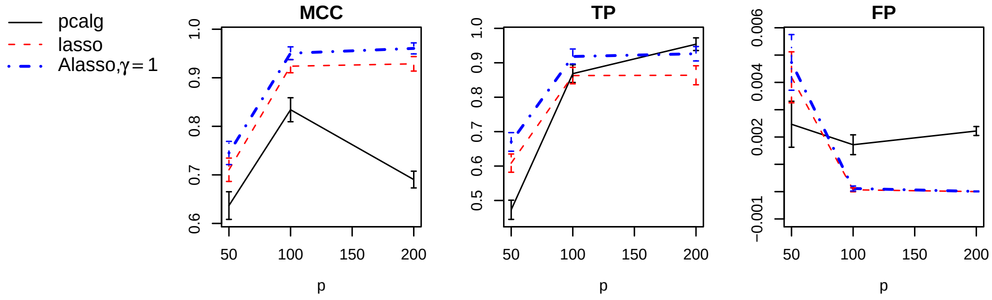
- Settings:
  $p = 50, 100, 200$
  $n = 100$
  Total number of edges in the network $= n$
  100 repetitions

- 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

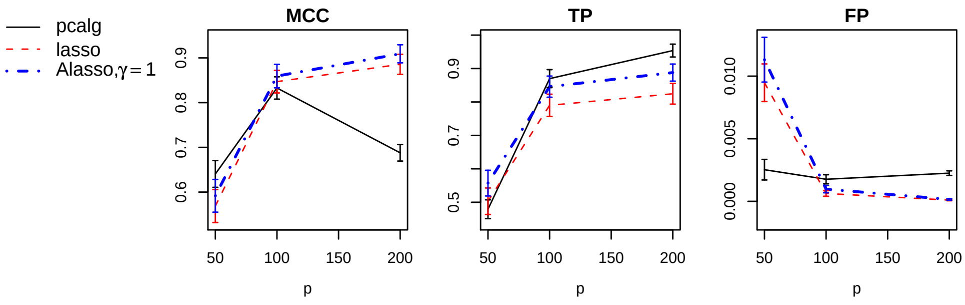
- Tuning parameter for both PC-Algorithm and penalized likelihood method based on false positive error α

Gaussian Observations



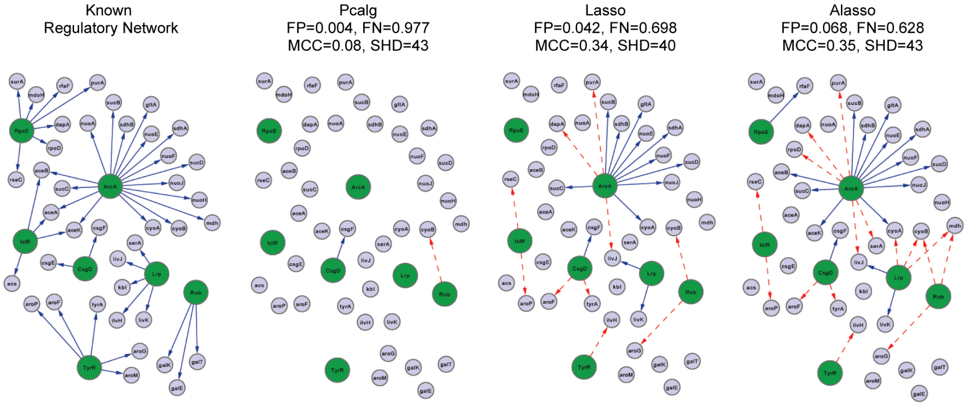


Random Ordering of Variables



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


Known
Regulatory Network

Pcalg
FP=0.004, FN=0.977
MCC=0.08, SHD=43

Lasso
FP=0.042, FN=0.698
MCC=0.34, SHD=40

Alasso
FP=0.068, FN=0.628
MCC=0.35, SHD=43

Summary

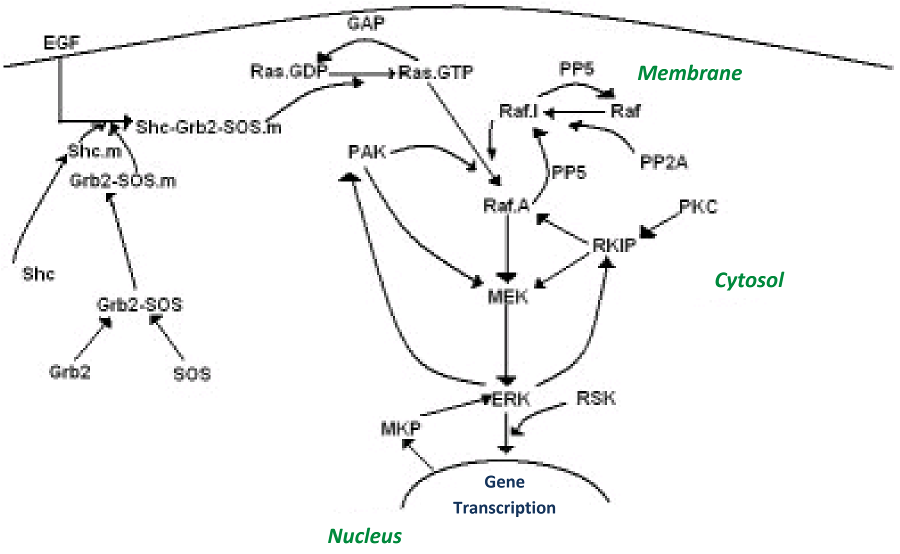
- Estimation of DAGs from observational data is both conceptually and computationally difficult
- Constraint-based and search-based algorithms become slow in high dimensions
- Also, may not be able to distinguish DAGs from observational data (Markov equivalence)
- Efficient penalized likelihood methods 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 – 2016

MAPK/ERK Pathway

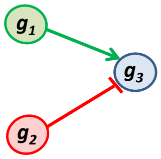


Estimation of Gene Regulator Networks

- 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

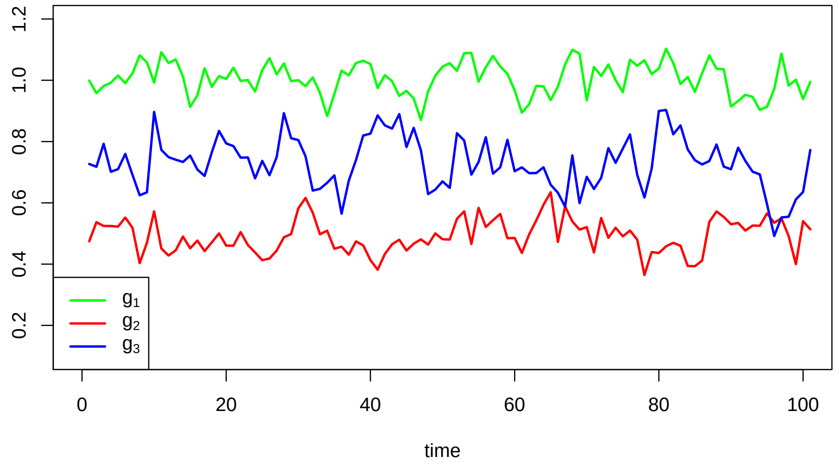
Consider a simple regulatory network, with two transcription factors and one gene:



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

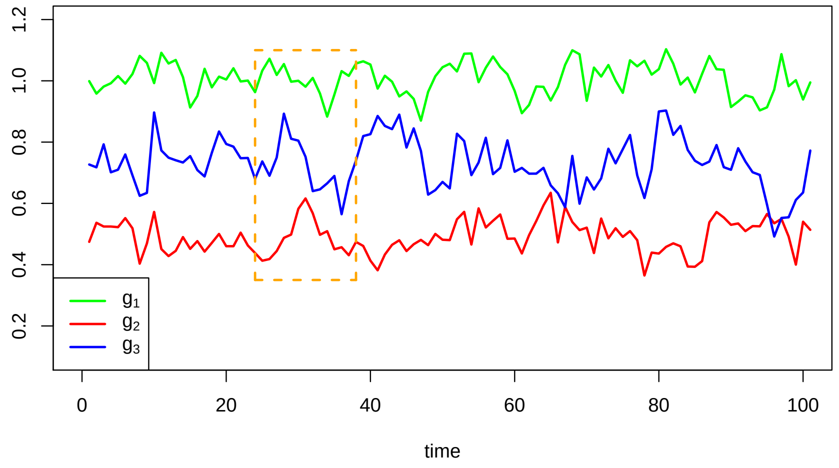
Gene Regulatory Networks

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



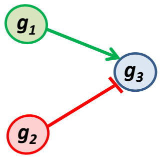
Gene Regulatory Networks

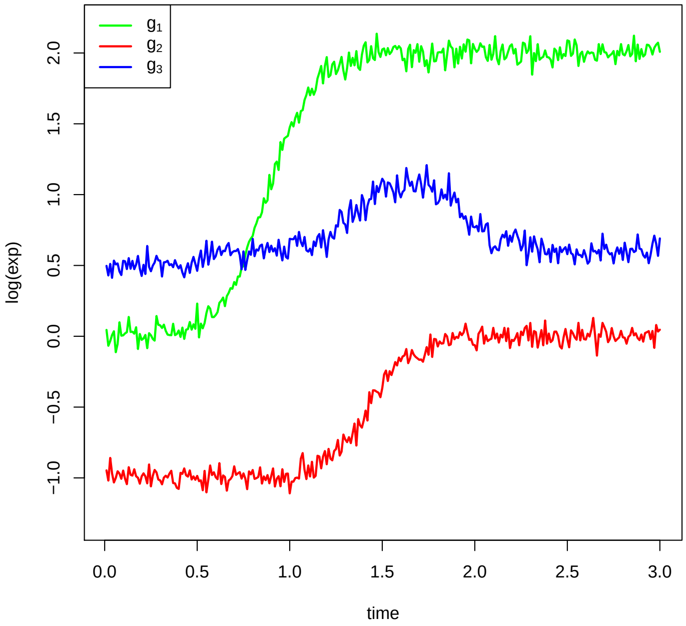
The temporal expressions patterns of g_1, g_2 and g_3 may look like:



Temporal patterns in Gene Regulatory Networks

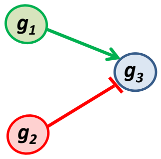
- g_1 : Inducer
- g_2 : Inhibitor
- g_3 : Regulated Gene

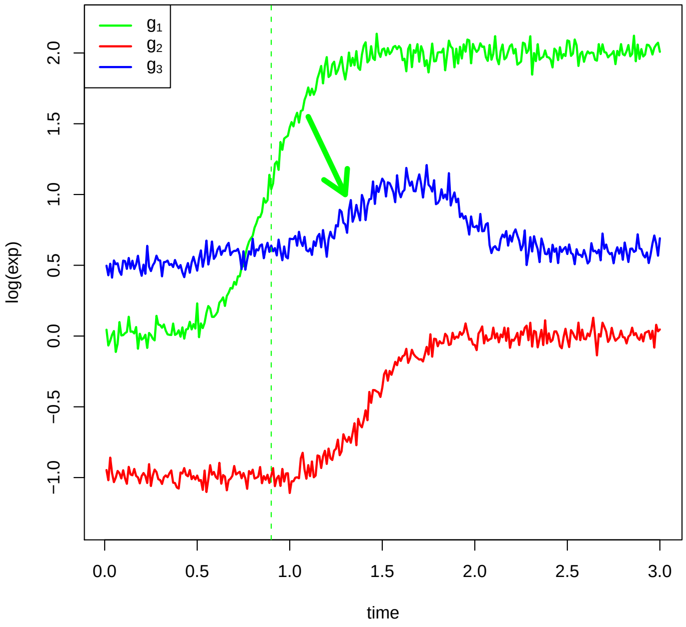




Temporal patterns in Gene Regulatory Networks

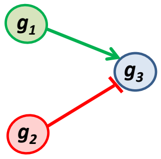
- g_1 : Inducer
- g_2 : Inhibitor
- g_3 : Regulated Gene

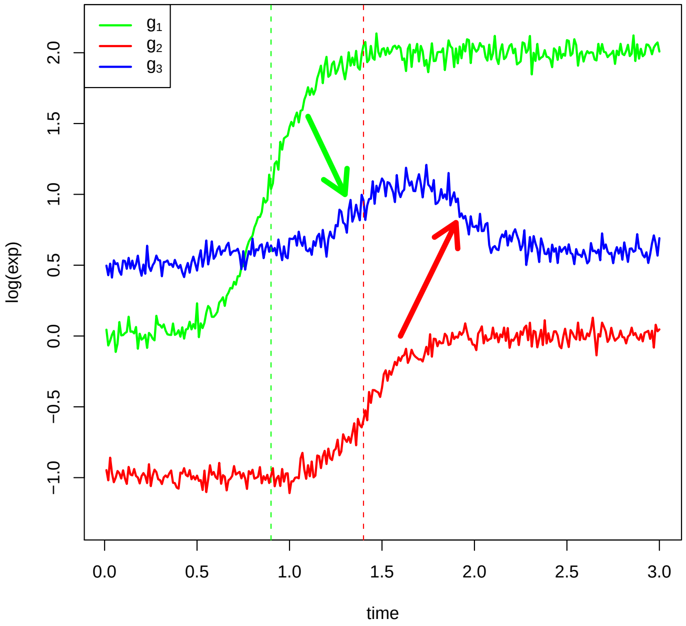




Temporal patterns in Gene Regulatory Networks

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

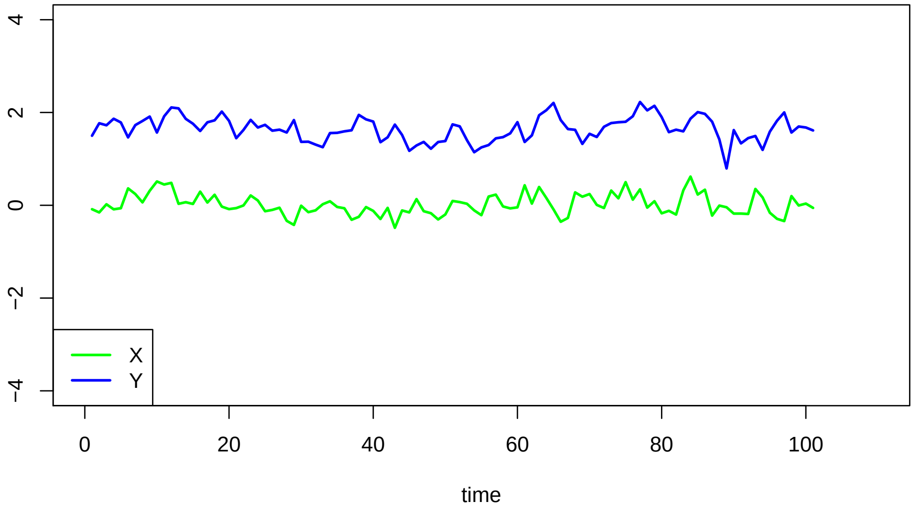




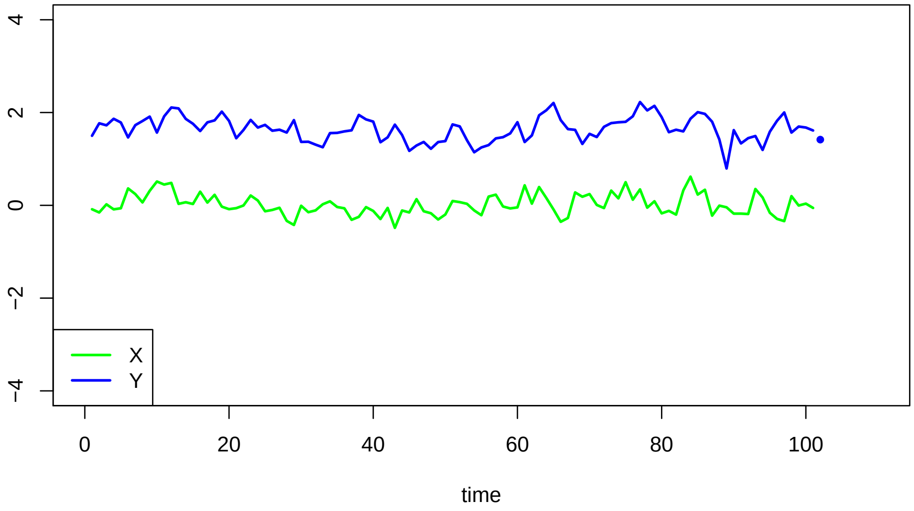
Estimation of Gene Regulatory Networks from Time-Course Data

- The goal is Discover interactions among genes from time-course data
- This is achieved by observing the 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

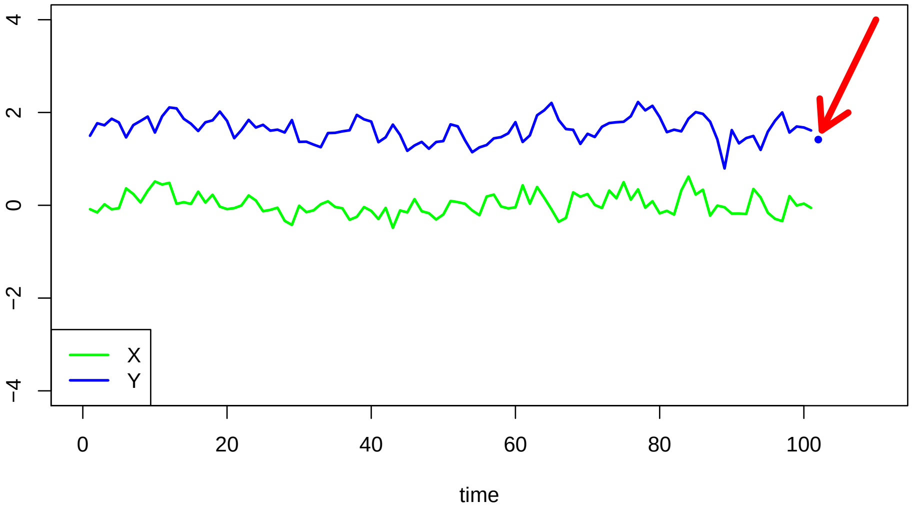
Granger Causality



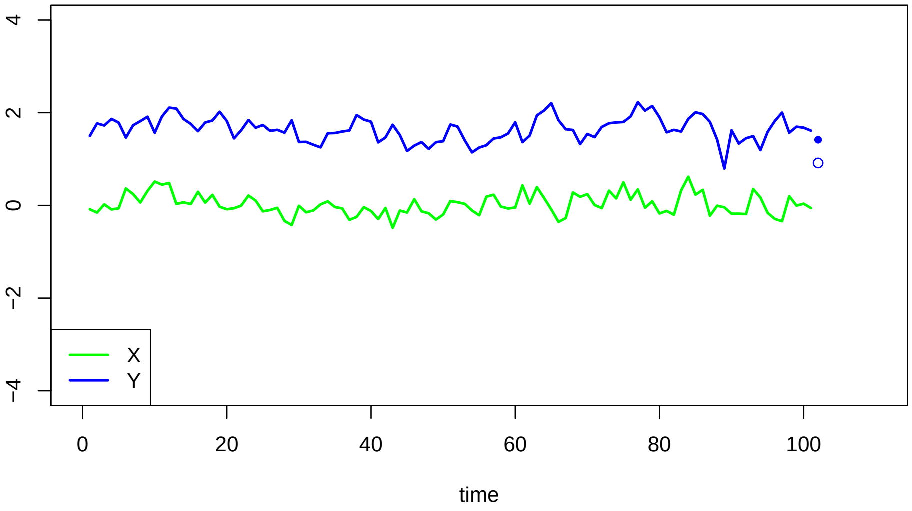
Granger Causality



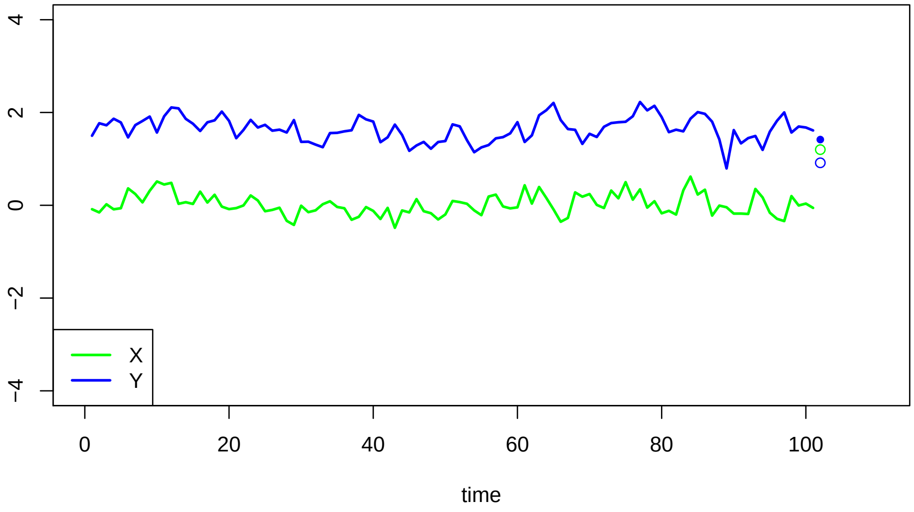
Granger Causality



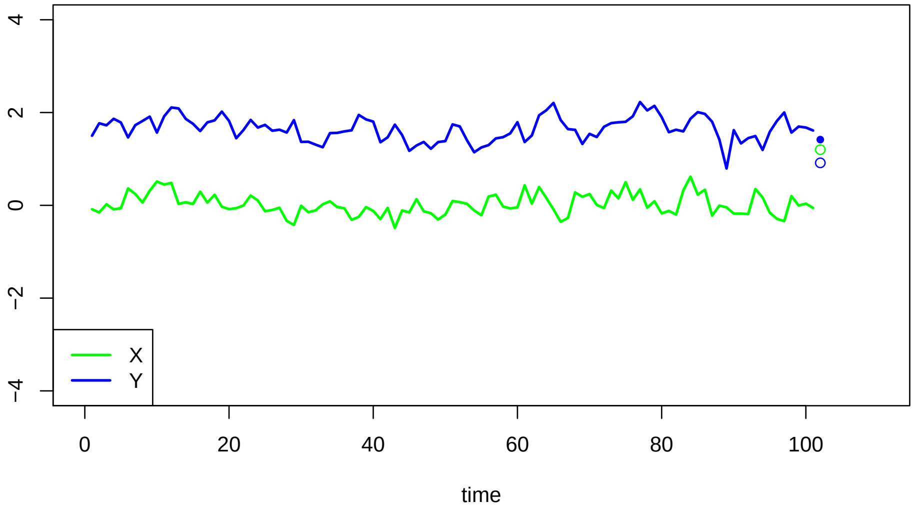
Granger Causality



Granger Causality



Granger Causality



We say X is Granger-causal for Y

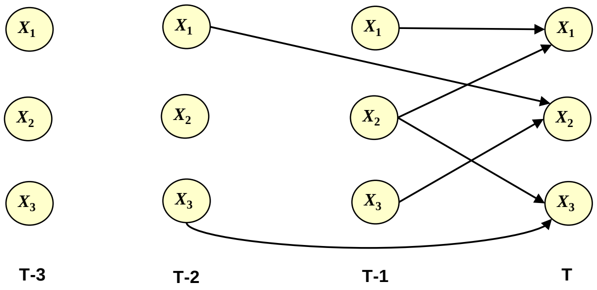
$$Y_t = 0.7Y_{t-1} + 0.4X_{t-1} + 0.2X_{t-2} + \varepsilon_t$$

Granger Causality

- 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 random variables
- We focus on extension of this idea to high dimensional settings, which we refer to as Network Granger Causality

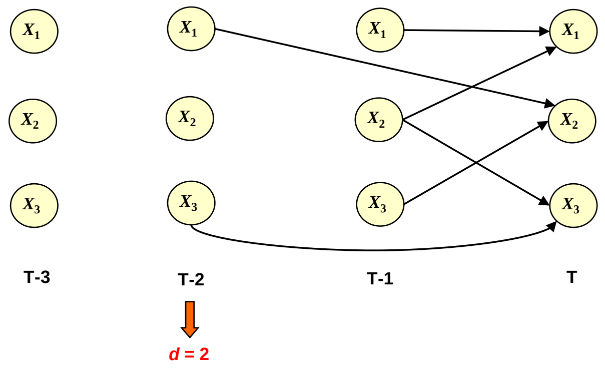
Network Granger Causality: Illustration

p variables observed over T time points



Network Granger Causality: Illustration

p variables observed over T time points



Network Granger Causality: Definition

- $X_1, \ldots, X_p$ stochastic processes and $\mathbf{X}^t = (X_1^t, \ldots, X_p^t)^\mathsf{T}$
- Network Granger Causality Model:

$$\mathbf{X}^T = A^1\mathbf{X}^{T-1} + \cdots + A^d\mathbf{X}^{T-d} + \varepsilon^T$$

- X_j^{T-t} is Granger-causal for X_i^T if $A_{i,j}^t \neq 0$.
- DAG with $(d+1) \times p$ variables
- alternatively, a vector autoregressive model of order d (VAR(d)) with p variables.
- Often $d \ll T$, but not known:
  - usually, d is "guessed", and is set to $d = 1$ (especially in applications of DBN), which can result in loss of information
  - the alternative is to include all previous time points (set $d = T - 1$) but that would result in too many variables
- Recent work has focused on simultaneous estimation of d and network.

Previous work on NGC in high dimensional settings

- The concept of Granger causality has been used in discovering gene regulatory interactions by Fujita et al (2007) and Mukhopadhyay and Chatterjee (2007)
- A number of recent work have considered penalized regression models for estimation of Granger-causal models:
  - lasso regression used in Arnold et al (2007) in a financial application
  - group lasso used in Lozano et al (2009) for grouping effects over time
  - *truncating* lasso Shojaie & Michailidis (2010) to estimate d and network simultaneously
  - lasso w *adaptive thresholding* used in Shojaie, Basu & Michailidis (2012) for improved estimation of d and network

Truncating Lasso Penalty

[1]$\mathcal{X}^t$: data at time t

$$\arg\min_{\theta^t \in \mathbb{R}^p} n^{-1} \|\mathcal{X}_i^T - \sum_{t=1}^{d} \mathcal{X}^{T-t}\theta^t\|_2^2 + \lambda \sum_{t=1}^{d} \Psi^t \sum_{j=1}^{p} |\theta_j^t| w_j^t$$

$$\Psi^1 = 1, \quad \Psi^t = M^{I\{\|A^{(t-1)}\|_0 < p^2\beta/(T-t)\}}, \ t \geq 2$$

where M is a large constant, and β is the user-specified false negative rate (FNR).

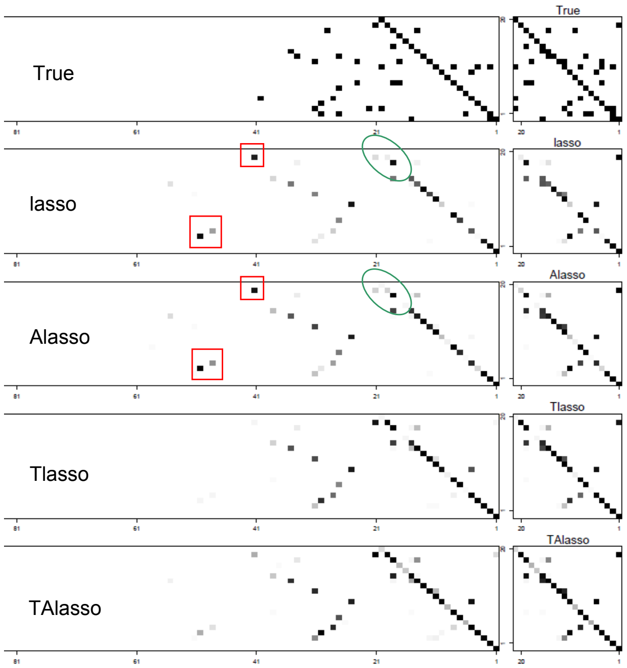
- Can use the following value of λ that controls a version of false positive rate (FPR) at the level α:

$$\lambda(\alpha) = 2n^{-1/2} Z^*_{\frac{\alpha}{2dp^2}}$$

- This method assumes that influences decay over time

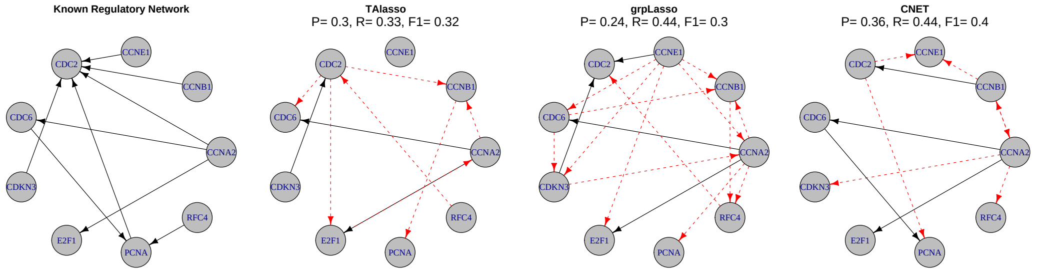
[1] Shojaie & Michailidis (2010)

An Illustrative Example



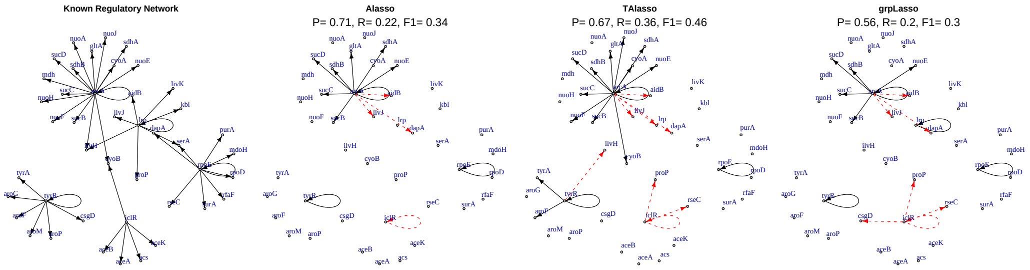
Example I: Gene Network of HeLa Cells

9 genes, 47 time points
$d = 3$

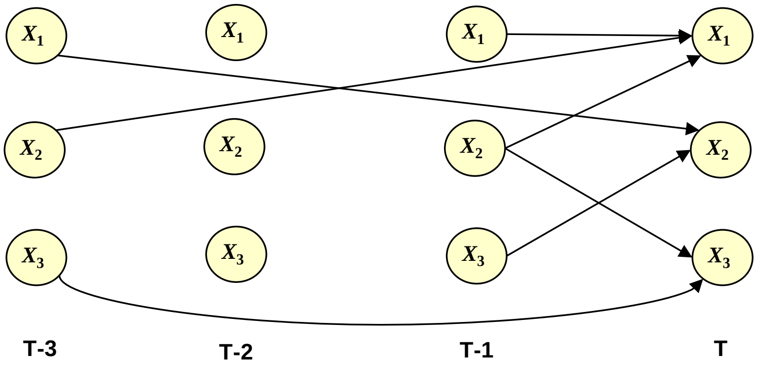


Example II: Gene Regulatory Networks of Yeast

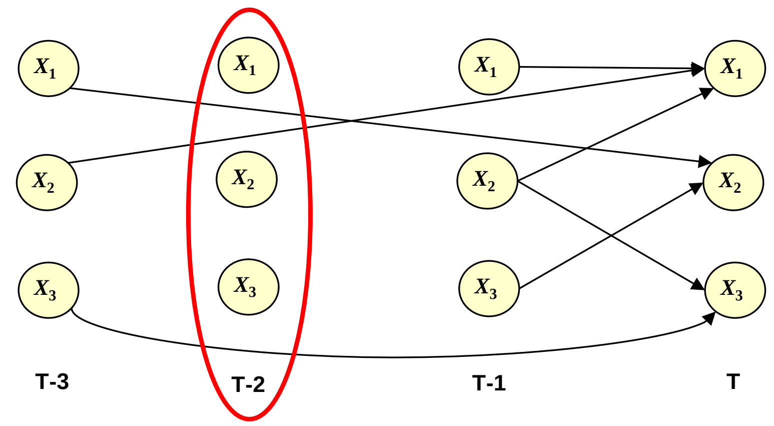
5 Transcription Factors, 37 genes ($p = 42$), 8 time points
$d = 2$



Non-decaying Granger-causal effects



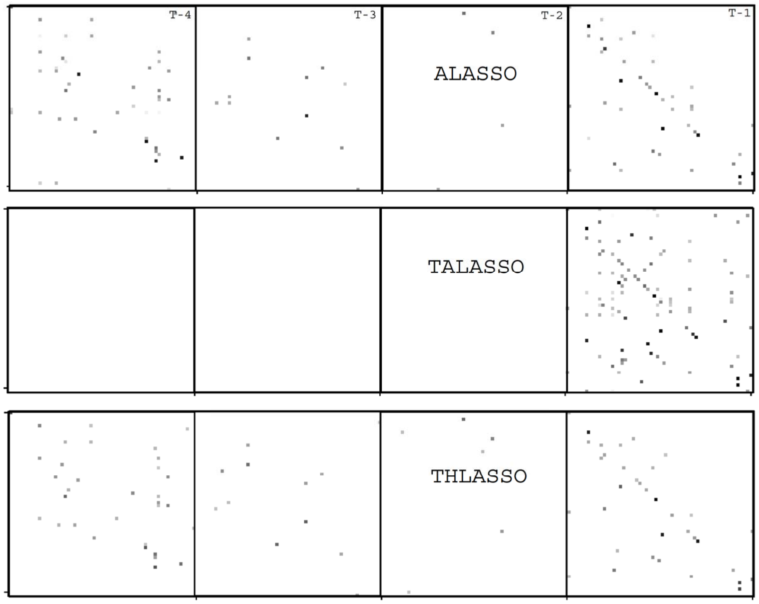
Non-decaying Granger-causal effects



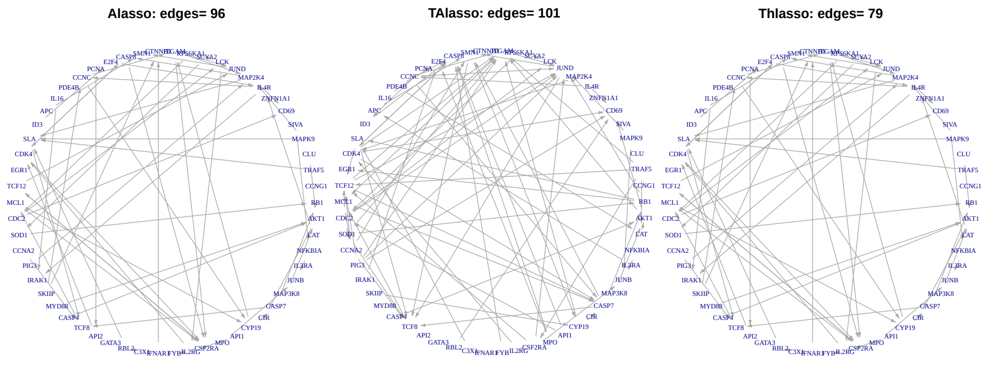
Regulatory Network of T-Cell Activation

- 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



Estimated Regulatory Networks



	Alasso	TAlasso	Thlasso
Alasso	(96)	–	–
TAlasso	99	(101)	–
Thlasso	35	102	(79)

Adaptively Thresholded Lasso Estimate: Main Idea

- Logic: Lasso is in general biased, and cannot achieve structure and norm consistency simultaneously
- In short, the idea is to start with lasso estimates, and then remove "small" values from the adjacency matrix
- Consider two levels of thresholding, one for each element of adjacency matrix, and the second for whole adjacency matrices at a given time point

Method Details

(i) Obtain the regular lasso estimate $\tilde{A}^t(\lambda_n)$ by solving

$$\underset{\theta^t\in\mathbb{R}^p}{\arg\min}\; n^{-1}\|\mathcal{X}_i^T - \sum_{t=1}^{d}\mathcal{X}^{T-t}\theta^t\|_2^2 + \lambda\sum_{t=1}^{T-1}\sum_{j=1}^{p}|\theta_j^t|w_j^t$$

(ii) Let $\Psi^t = \exp\Big(M\mathbf{1}_{\{\|\tilde{A}^t\|_0<p^2\beta/(T-1)\}}\Big)$, and define the thresholded estimate:

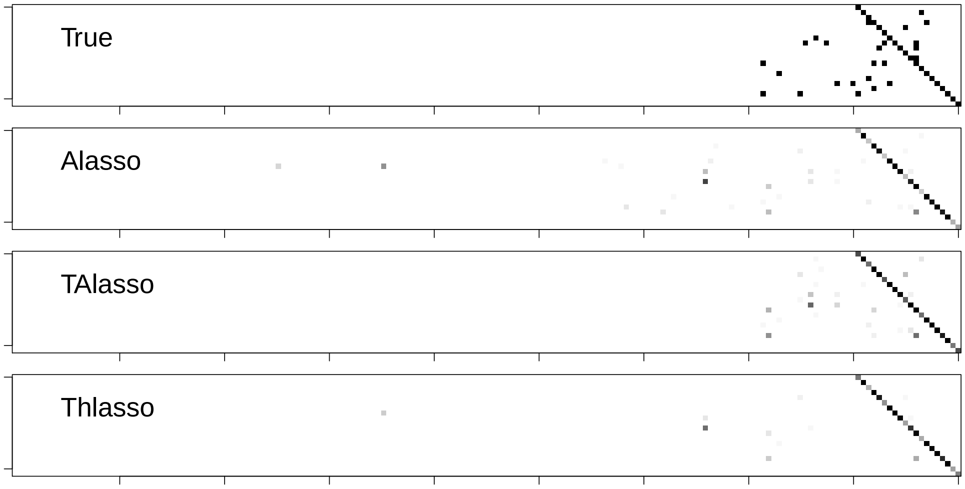
$$\hat{A}_{ij}^t = \tilde{A}_{ij}^t\mathbf{1}_{\{|\tilde{A}_{ij}^t|\geq\tau\,\Psi^t\}}$$

Here M is a large constant and τ is tuning parameter for thresholding.

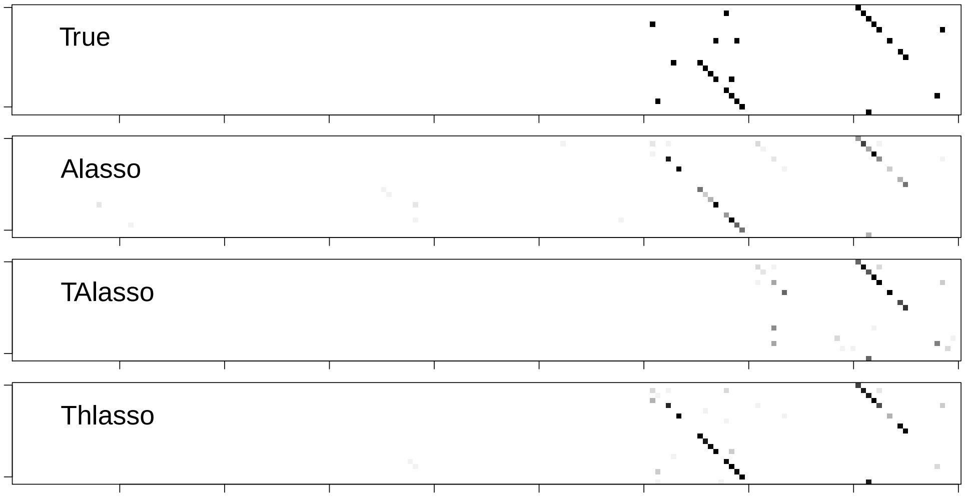
(iii) Estimate the order of the time series by setting

$$\hat{d} = \max_t\{t : \|\hat{A}^t\|_0 \geq p^2\beta/(T-1)\}$$

Illustrative Ex I: Under Decay Assumption



Illustrative Ex II: Decay Assumption Violated



Comments

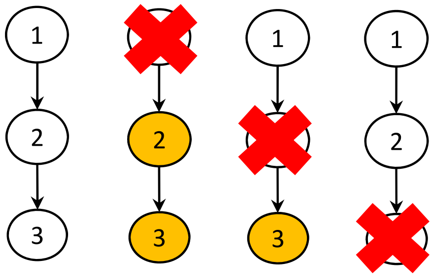
- Benefits:
  - The optimization problem is convex, and can be solved efficiently.
  - Does not require structural assumptions (no decay assumption)
- Drawbacks:
  - Requires more tuning parameters
  - Can be less efficient than truncating lasso if the decay assumption holds
- The tuning parameters can 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.



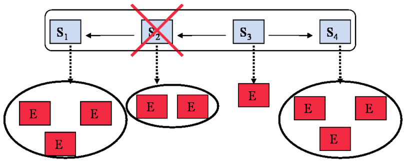
- The *discretized* perturbation data
  (i) do not provide enough information to construct the structure of regulatory networks.
  (ii) provide enough information to determine causal (topological) ordering(s) of nodes.

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

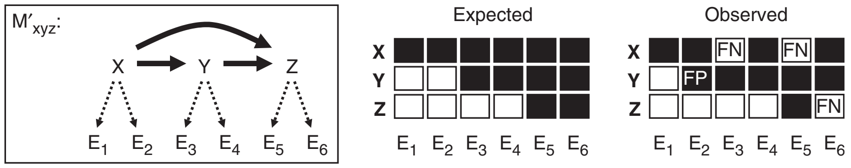
Nested Effect Models

- Motivated by RNAi experiment settings: few knocked-out genes (called *S* genes), and a larger number of 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



Expected

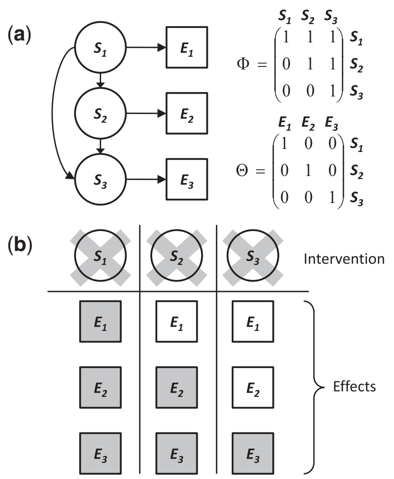
	E_1	E_2	E_3	E_4	E_5	E_6
X	■	■	■	■	■	■
Y	□	□	■	■	■	■
Z	□	□	□	□	■	■

Observed

	E_1	E_2	E_3	E_4	E_5	E_6
X	■	■	FN	■	FN	■
Y	□	FP	■	■	■	■
Z	□	□	□	□	■	FN

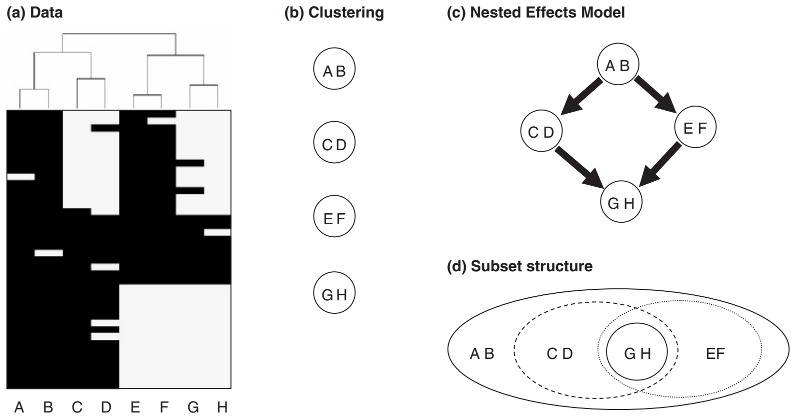
- Works with discretized data: there is either an effect (1) from knocking out of S_i on E_j or not (0)
- Assumes there are positive and negative control samples
- Allows for presence of false positives and false negatives in the discretized data

Nested Effect Models


(a)
S_1
S_2
S_3
E_1
E_2
E_3
$$\Phi = \begin{array}{c|ccc|c} & S_1 & S_2 & S_3 & \\ & 1 & 1 & 1 & S_1 \\ & 0 & 1 & 1 & S_2 \\ & 0 & 0 & 1 & S_3 \end{array}$$
$$\Theta = \begin{array}{c|ccc|c} & E_1 & E_2 & E_3 & \\ & 1 & 0 & 0 & S_1 \\ & 0 & 1 & 0 & S_2 \\ & 0 & 0 & 1 & S_3 \end{array}$$
(b)
Intervention
Effects

- 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 Φ 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)

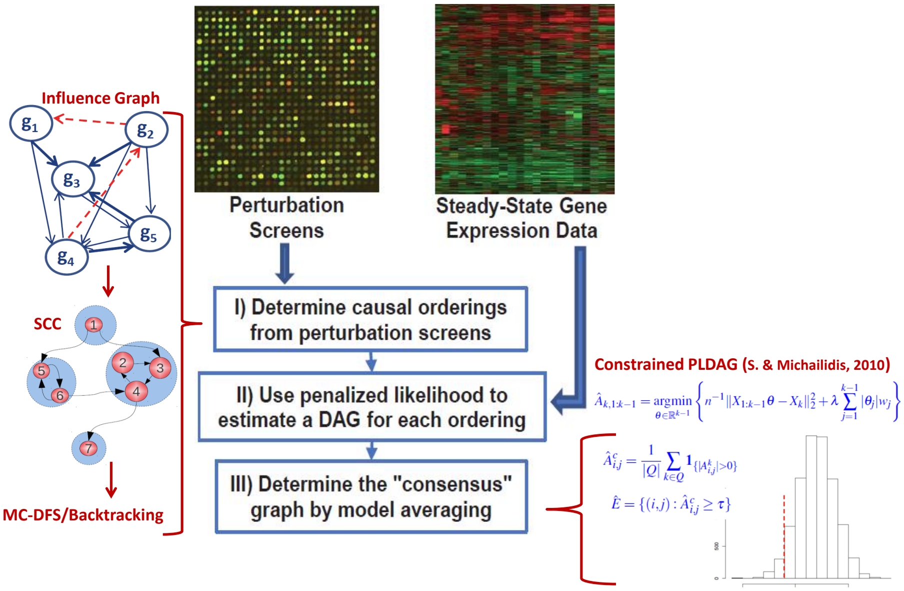
Nested Effect Models

```
> 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")

nem(D, ...)
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 ~ 100 S genes (though very slow), but may not work for larger experiments

The RIPE Algorithm[2]



Perturbation Screens

Steady-State Gene Expression Data

I) Determine causal orderings from perturbation screens

II) Use penalized likelihood to estimate a DAG for each ordering

III) Determine the "consensus" graph by model averaging

Constrained PLDAG (S. & Michailidis, 2010)

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

$$\hat{A}^c_{i,j} = \frac{1}{|Q|} \sum_{k \in Q} \mathbf{1}_{\{|A^k_{i,j}|>0\}}$$

$$\hat{E} = \{(i,j) : \hat{A}^c_{i,j} \geq \tau\}$$

[2] Regulatory Network Inference from joint Perturbation and Expression data (Shojaie et al, 2014), package `ripe` on github

The RIPE Algorithm

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

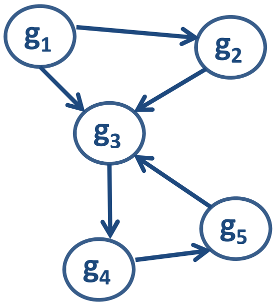
I) Use perturbation data to determine causal ordering(s) among nodes

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

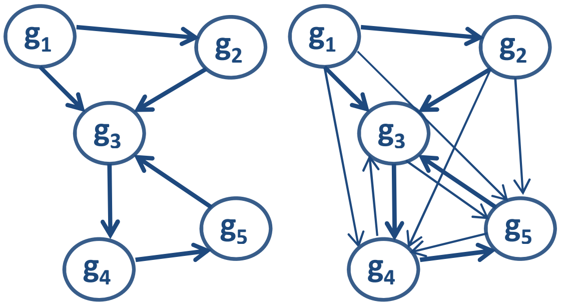
Step I) Determining Causal Orderings

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



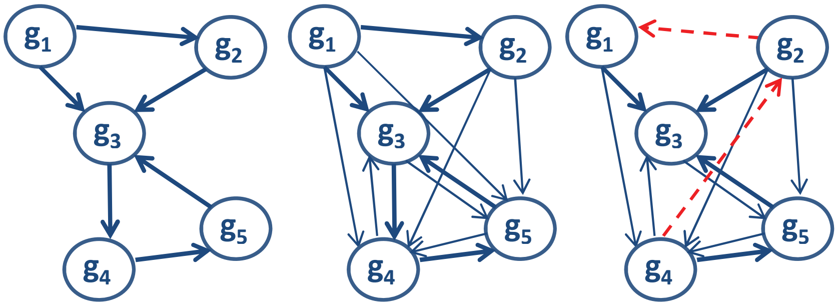
Step I) Determining Causal Orderings

- 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



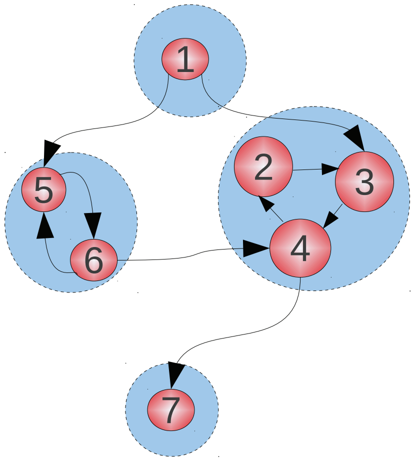
Step I) Determining Causal Orderings

- 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

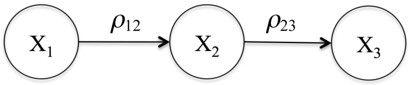
- Create a hyper-graph of strong connected components (SCC), where each node is a collection of ≥ 1 nodes that cannot be further ordered (i.e. there is a cycle).
- Find an ordering (topological sorting) of the SCC graph (note, this is by construction a DAG) using Depth First Search algorithm (DFS).
- 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

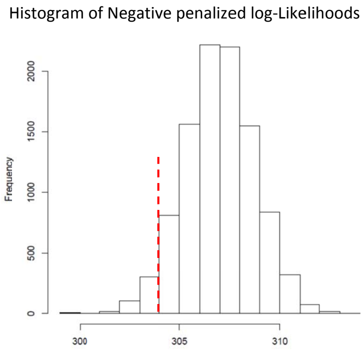
- Given a topological ordering of nodes, the nodes of the graph can be rearranged to form a DAG
- For each ordering, estimate (the structure of) one DAG using the penalized likelihood method of the previous lecture, (by solving $p-1$ lasso regression problems):

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



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...).

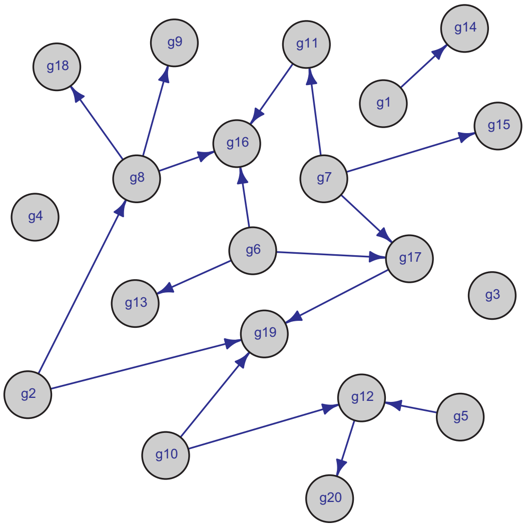


- Solution: average over edges with the best scores:

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

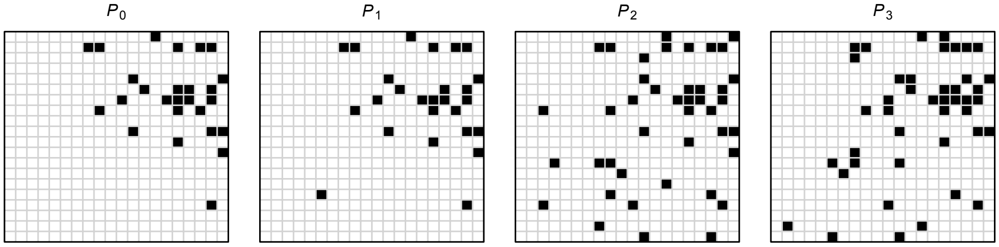
  - L_q: lower qth quantile of (penalized) negative log-likelihoods
  - $Q = \{o \in \mathcal{O} : \ell(o) \leq L_q\}$ set of orderings for these likelihoods

Simulate Network: DAG of size $p = 20$



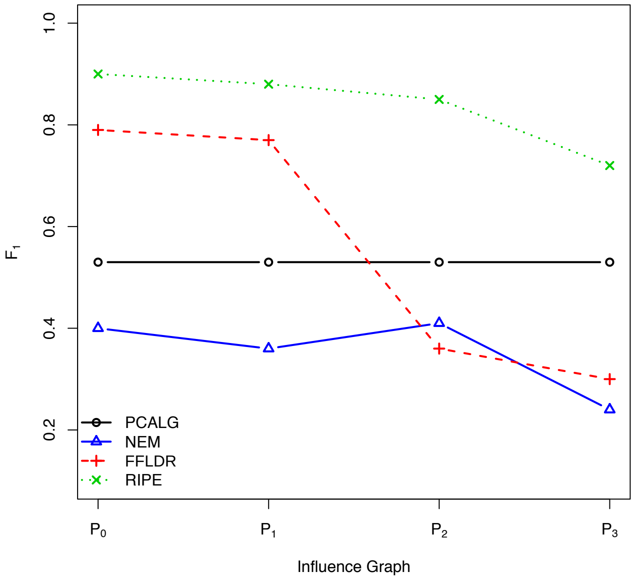
Data Generation

Perturbation data: Adjacency matrices of true and noisy influence graphs



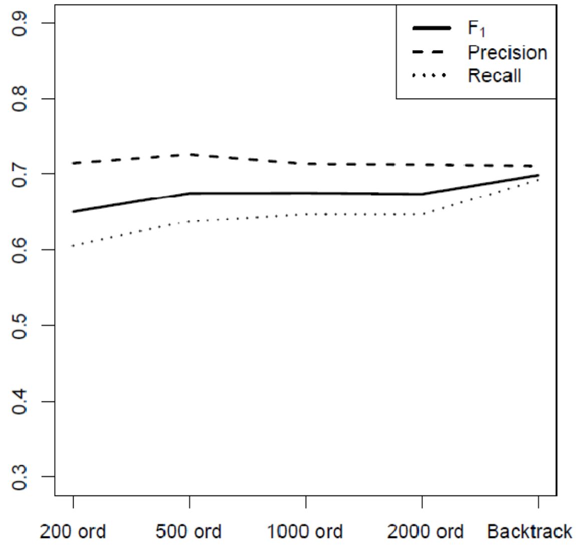
Steady-state expression data: generated $n = 50$ Gaussian observations according to the true DAG.

Comparison of F_1 measures



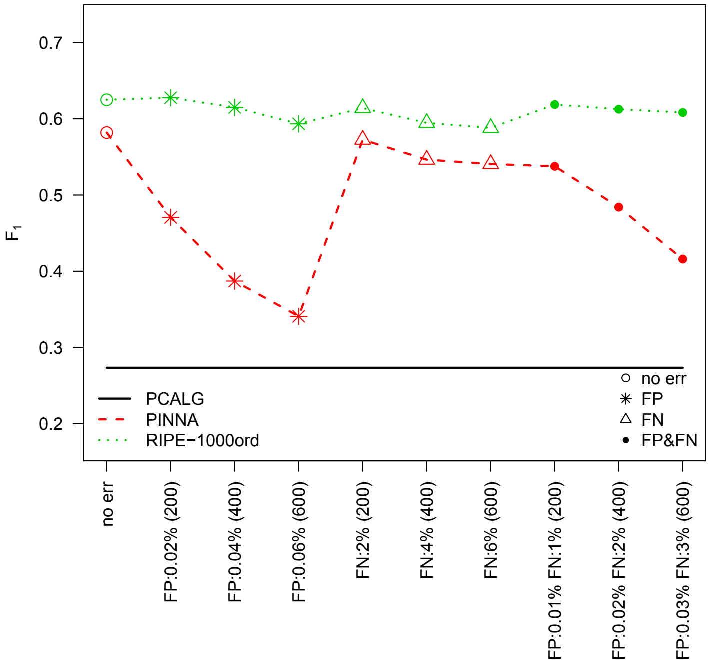
How Many Orderings?

For P_3, there are a total of 3962 orderings using the backtracking algorithm.



High Dimensional Cyclic Graphs ($p = 1000$)

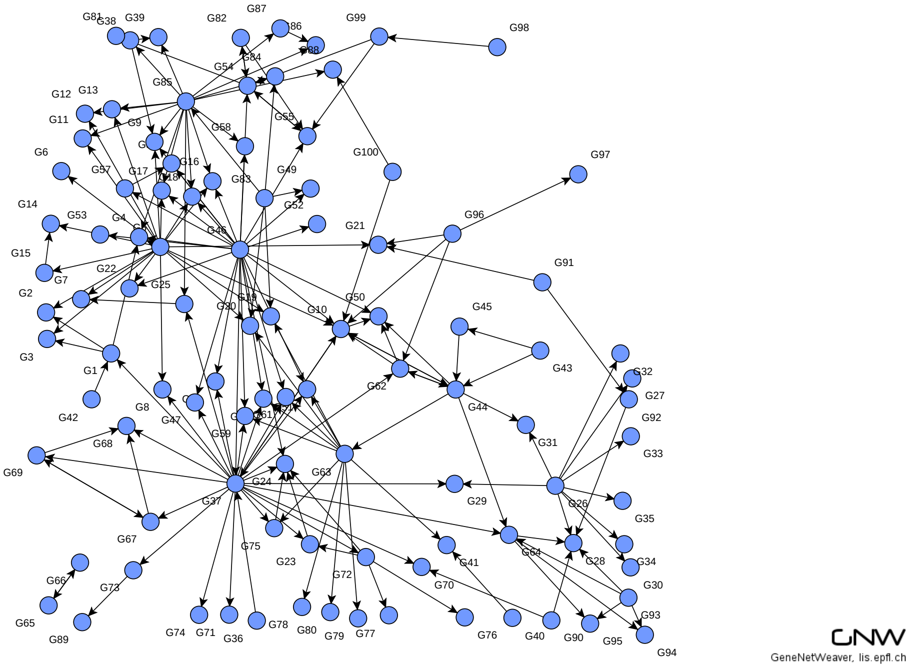
Effect of FP and FN errors



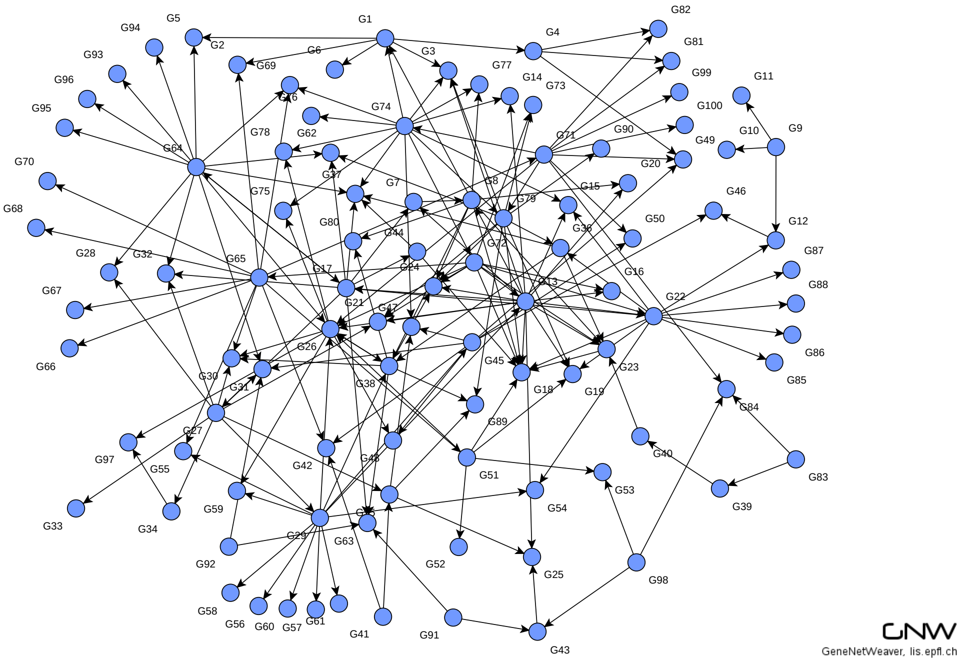
A More Complicated Example: DREAM-4 Challenge

- 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
- The algorithm of Pinna et al (PINNA) was the winner of the high dimensional reconstruction challenge (on networks of size 100)

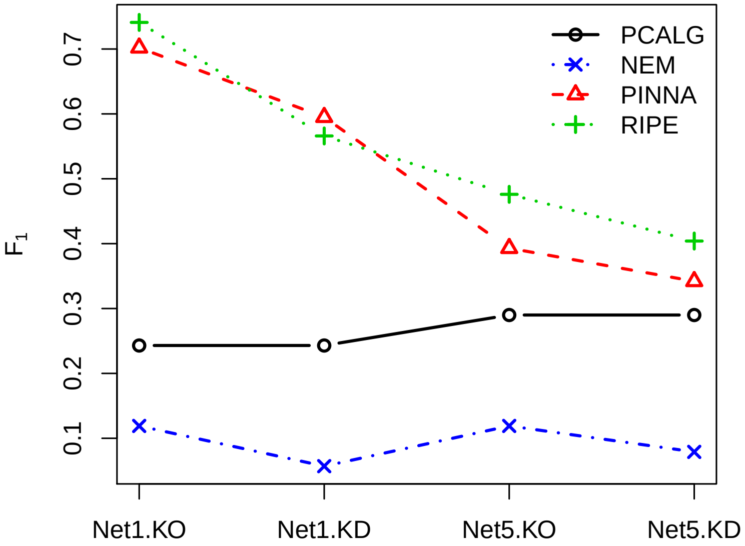
DREAM Network 1 (Simplest)



DREAM Network 5 (Most Difficult!)



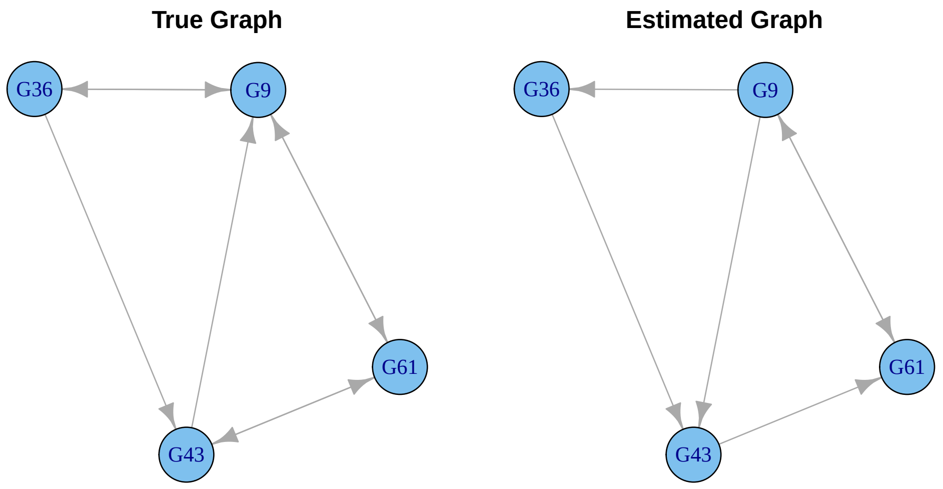
Comparison of F_1 Measures



Example of estimated modules

Largest cyclic component in DREAM1 network

When the perturbation data includes cycles, the consensus graph will be cyclic.

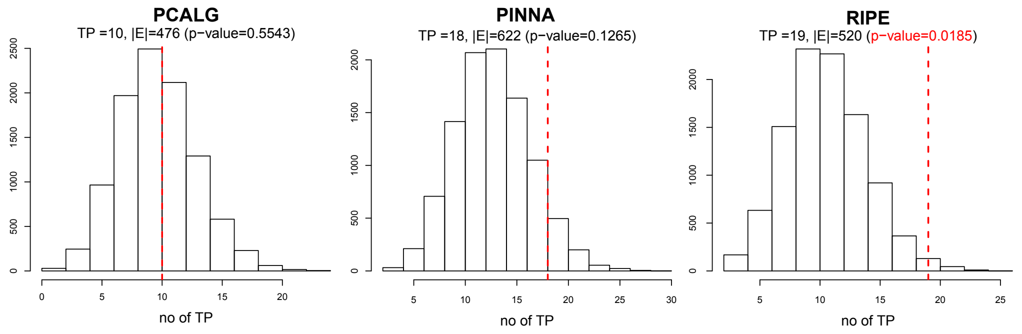


Network of Yeast Transcription Factors

- 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

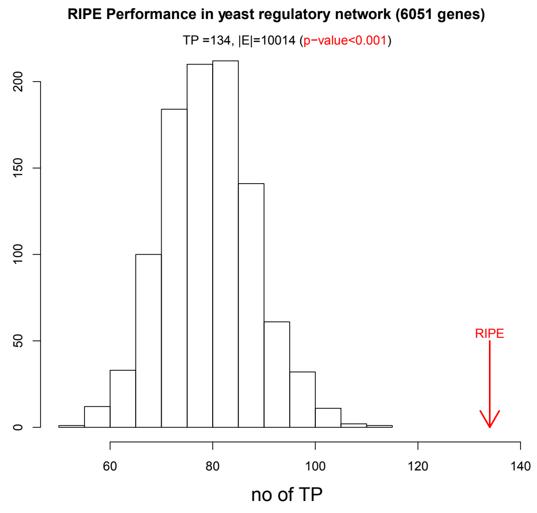
Network of Yeast Transcription Factors

- 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



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...

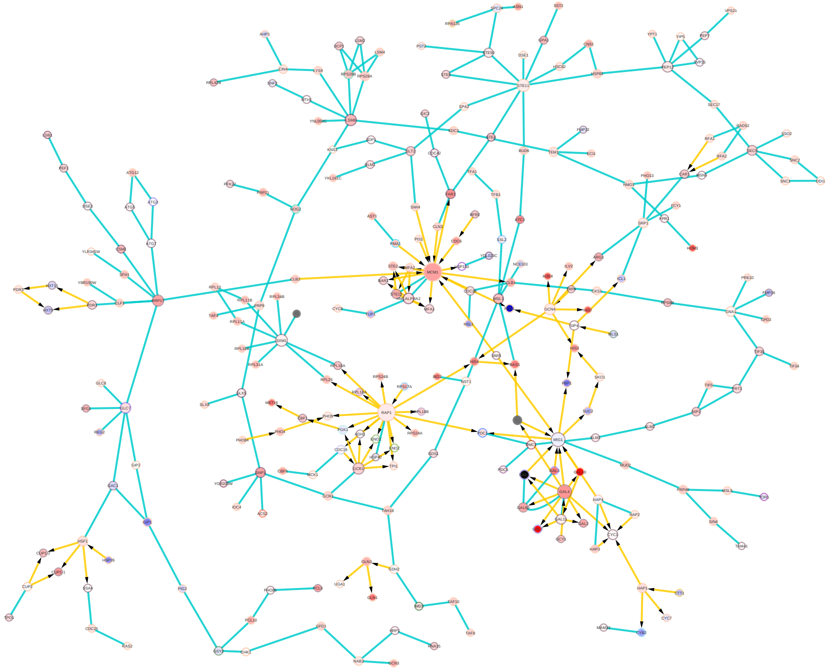
Pathway & Network Analysis of Omics Data: Network-Based Pathway Enrichment Analysis

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

Summer Institute for Statistical Genetics – 2016

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

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

- Test the significance of *a priori* defined gene sets
- Preserve the correlation among genes in the gene set
- Based on a competitive null hypothesis, where activity of each pathway is compared with other pathways, often using a permutation test
- Competitive tests of enrichment assume that a small number of genes have differential activity, and are very sensitive to the choice of gene sets, they also problem with
- Self-contained tests address these issues, but may be less efficient or sensitive to model assumptions (*Goemen & Buhlmann* (2007), *Ackermann & Strimmer* (2009))

Signaling Pathway Impact Analysis (SPIA)

- Combines classical 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)
- Models each pathway separately (ignores connections among pathways)
- Implemented in the Bioconductor package `SPIA`

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 $P_{NDE} = P(X \geq N_{DE} \mid H_0)$

The SPIA Methodology

SPIA combines two types of evidence

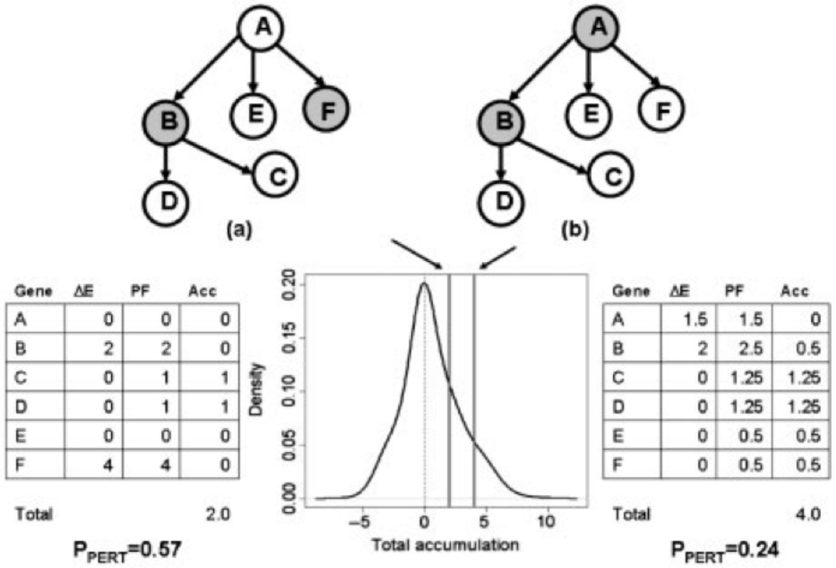
(ii) the abnormal perturbation of the pathway

- 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)
    - β_{ij} is the magnitude of effect of gene j on gene i; currently, $beta_{ij} = 1$ if $j \to i$
    - $\Delta E(g_i)$ 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 $ACC(g_i) = B \cdot (I - B)^{-1} \Delta E$
    - B is the normalized matrix of β's: $B_{ij} = \beta_{ij}/N_{DS}(g_j)$
    - ΔE 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



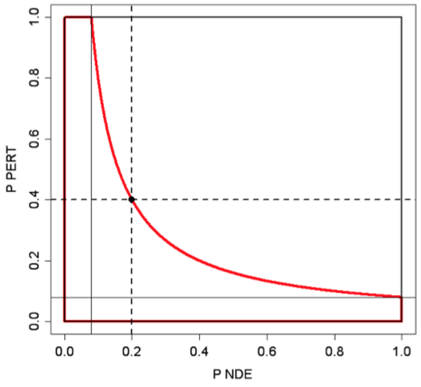
Gene	ΔE	PF	Acc
A	0	0	0
B	2	2	0
C	0	1	1
D	0	1	1
E	0	0	0
F	4	4	0
Total			2.0

Gene	ΔE	PF	Acc
A	1.5	1.5	0
B	2	2.5	0.5
C	0	1.25	1.25
D	0	1.25	1.25
E	0	0.5	0.5
F	0	0.5	0.5
Total			4.0

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

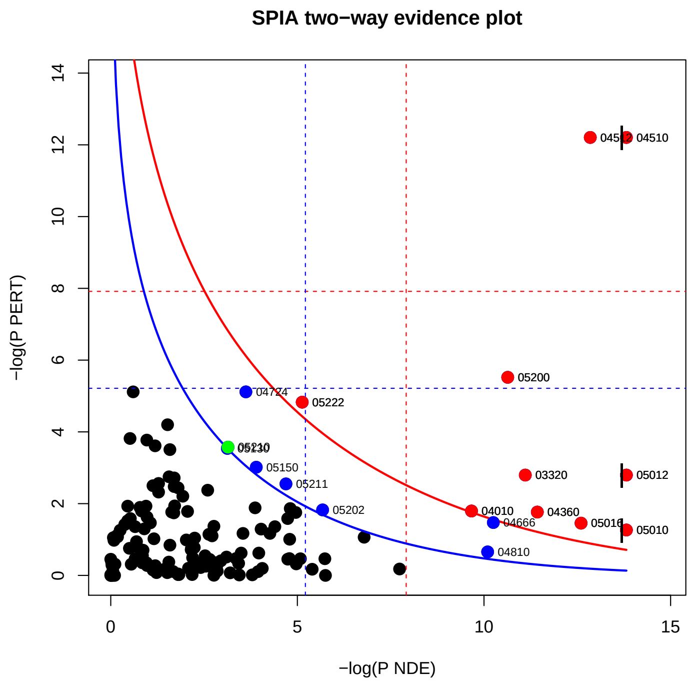
```
data(colorectalcancer)

#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



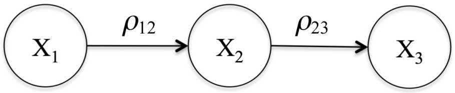
Network-Based Gene Set Analysis (NetGSA)

- 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
- Edges in the network can be directed $j \to k$ or undirected $j \leftrightarrow k$
- Edges defines 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 $k \to j$ & for undirected edges, $A_{jk} = A_{kj}$
- Pathways defined *a priori* based on common biological functions, etc

The Latent Variable Model: Main Idea



$$
\begin{aligned}
X_1 &= \gamma_1 \\
X_2 &= \rho_{12} X_1 + \gamma_2 = \rho_{12}\gamma_1 + \gamma_2 \\
X_3 &= \rho_{23} X_2 + \gamma_3 = \rho_{23}\rho_{12}\gamma_1 + \rho_{23}\gamma_2 + \gamma_3
\end{aligned}
$$

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 ith sample in the expression data
- Let $Y = X + \varepsilon$, with X the signal and $\varepsilon \sim N_p(0, \sigma_\varepsilon^2 I_p)$ the noise
- The influence matrix Λ 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

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

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

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

where $\boldsymbol{\beta}$ and $\boldsymbol{\gamma}$ are fixed and random effect parameters and

$$\boldsymbol{\varepsilon} \sim N_{np}(\mathbf{0}, R(\theta_\varepsilon)), \qquad \boldsymbol{\gamma} \sim N_{np}(\mathbf{0}, \sigma^2_\gamma \mathbf{I_{np}})$$

- Temporal Correlation incorporated through R

In general, the design matrices, $\boldsymbol{\Psi}$ and $\boldsymbol{\Pi}$ depend on the experimental settings (similar to ANOVA), and are functions of Λ

Estimation of MLM Parameters

MLE for β:

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

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

$\hat{\beta}$ depends on estimates of σ^2_γ and θ^2_ε (estimated using restricted maximum likelihood (REML)).

Inference using MLM

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

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

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

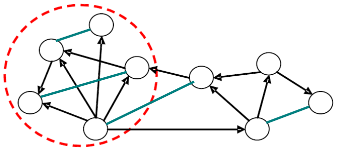
$$T = \frac{\ell\hat{\beta}}{\sqrt{\ell\hat{C}\ell'}}$$

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

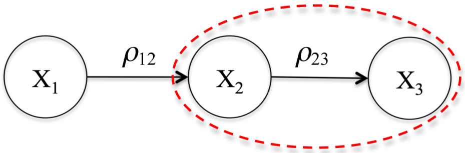
- One intuitive choice is to use the indicator vector for the members of pathway **b**, but this only reflects changes in the mean vector
- Need to *de-couple the effect of each subnetwork* from other nodes



- Can be shown that $(\mathbf{b}\Lambda \cdot \mathbf{b})\gamma$ 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:

$$\ell^* = \left(-\mathbf{b} \cdot \mathbf{b}\Lambda^C, \mathbf{b} \cdot \mathbf{b}\Lambda^T\right)$$

"Optimal" Choice of Contrast Vector



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

Consider the set, $\mathbf{b} = (0, 1, 1)$; then

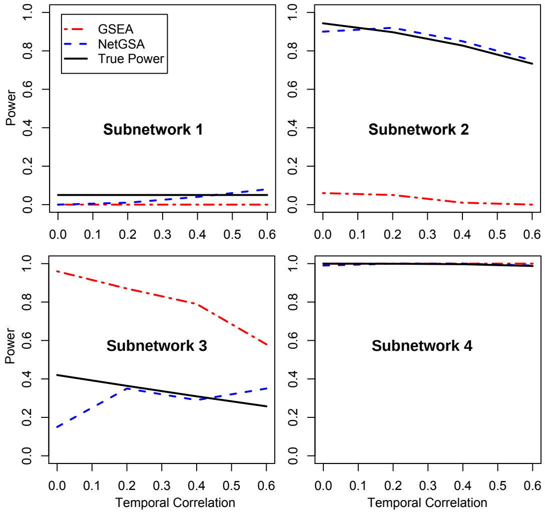
$$(\mathbf{b}\Lambda) = (\rho_{12} + \rho_{12}\rho_{23}, 1 + \rho_{23}, 1)$$

On the other hand,

$$(\mathbf{b}\Lambda \cdot \mathbf{b}) = (0, 1 + \rho_{23}, 1)$$

Comparison in Simulated Data

Subnetwork	Mean	Network Influence
1	$\mu_1 = \mu_2 = 1$	$\rho_1 = \rho_2 = 0.2$
2	$\mu_1 = 1, \mu_2 = 2$	$\rho_1 = \rho_2 = 0.2$
3	$\mu_1 = \mu_2 = 1$	$\rho_1 = 0.2, \rho_2 = 0.7$
4	$\mu_1 = 1, \mu_2 = 2$	$\rho_1 = 0.2, \rho_2 = 0.7$



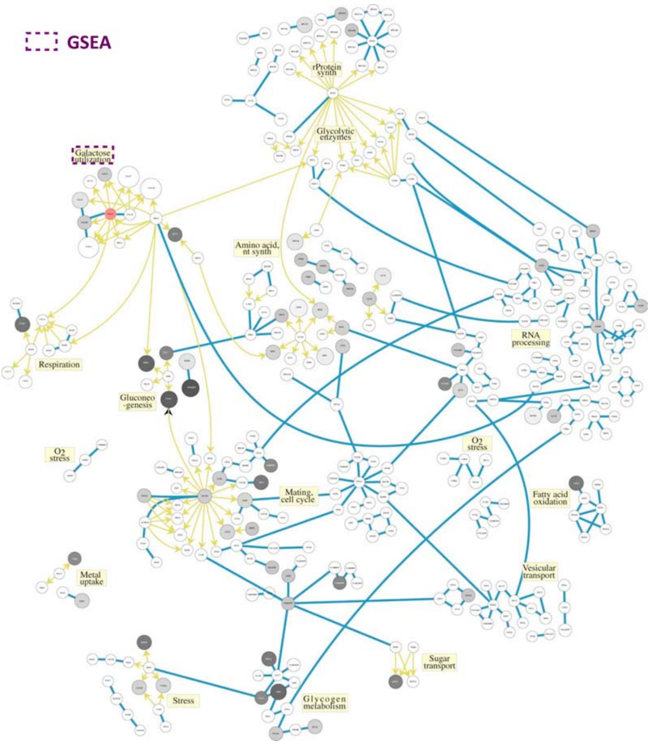
Yeast Galactose Utilization Pathway

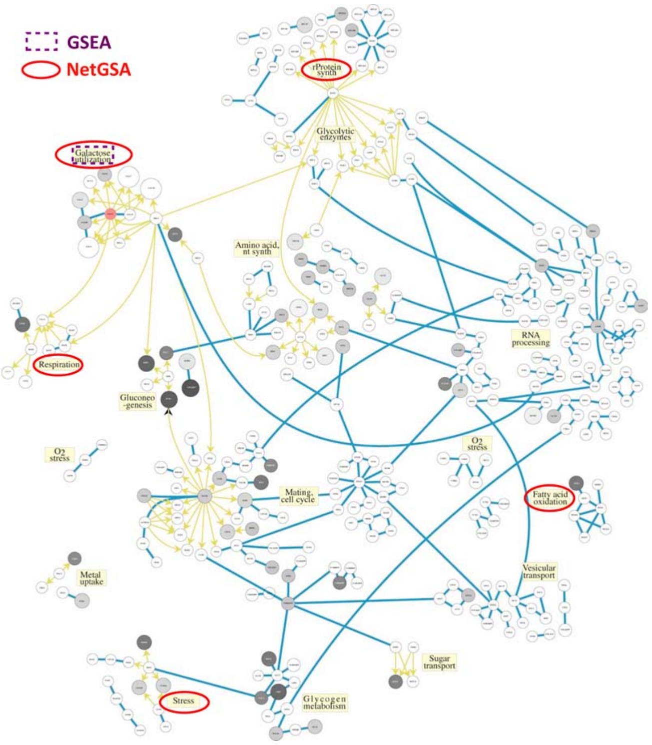
Ideker et al (2001) data on yeast Galactose Utilization Pathway

- 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–


GSEA
rProtein synth
Glycolytic enzymes
Galactose utilization
Amino acid, nt synth
RNA processing
Respiration
Gluconeo -genesis
O2 stress
O2 stress
Mating, cell cycle
Fatty acid oxidation
Metal uptake
Vesicular transport
Sugar transport
Stress
Glycogen metabolism


GSEA
NetGSA
rProtein synth
Glycolytic enzymes
Galactose utilization
Amino acid, nt synth
RNA processing
Respiration
Gluconeo -genesis
O2 stress
O2 stress
Mating, cell cycle
Fatty acid oxidation
Metal uptake
Vesicular transport
Sugar transport
Stress
Glycogen metabolism

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

Gasch et al (2000)

- Gene Expression Data

Experiment	Obs. Time (after 33C)
Mild heat shock (*29C to 33C*), no sorbitol	5, 15, 30 min
Mild Heat Shock, 1M sorbitol at 29C & 33C	5, 15, 30 min
Mild Heat Shock, 1M sorbitol at 29C	5, 15, 30 min

- 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

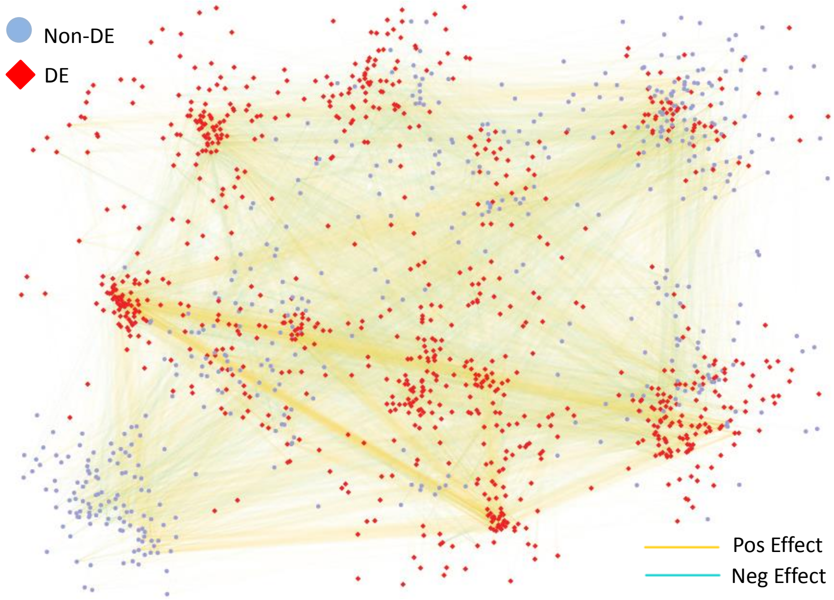
Model and Results

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

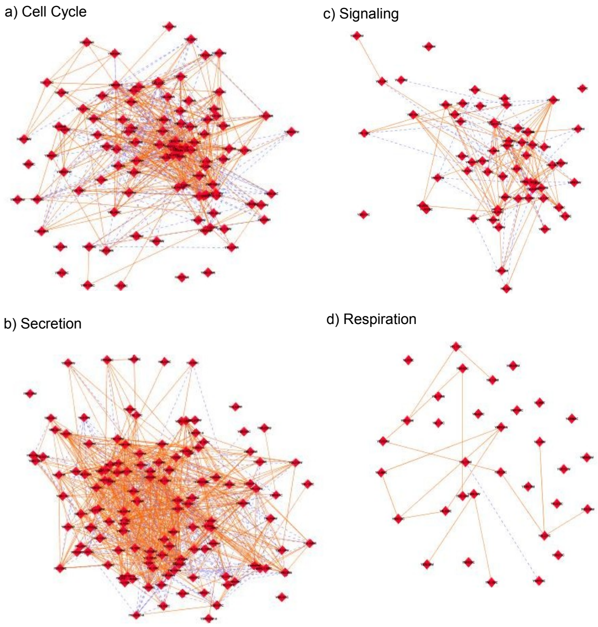
$$\mathbb{E}Y_{11} = \Lambda\mu, \qquad \mathbb{E}Y_{jk} = \Lambda(\mu + \alpha_j + \delta_k) \quad j, k = 2, 3$$

- 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

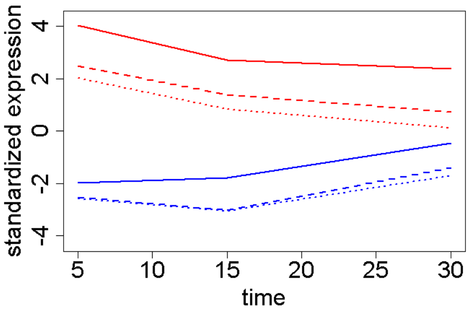


Significant subnetworks



Expression Profiles

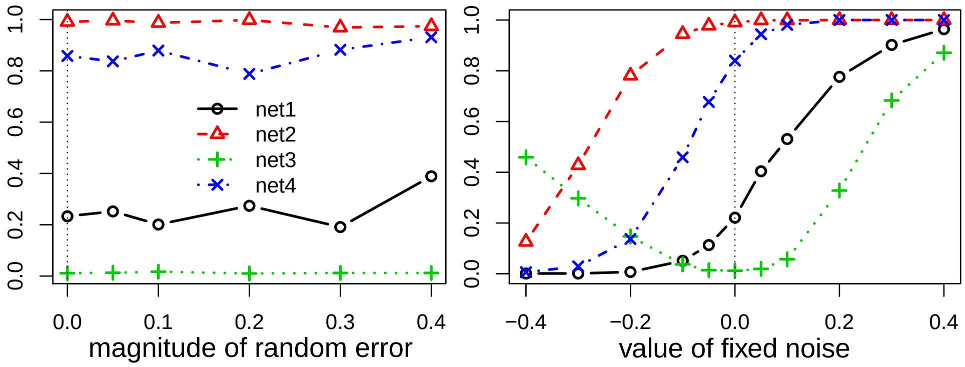
Average Standardized Expression Levels of Pathways



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

Effect of Noise In Network Information

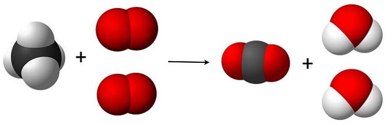
- Let $\tilde{A}$ be observed network information, and A be the truth.
- It can be shown that, if $\|\tilde{A} - A\|$ is small then, NetGSA still works (is *asymptotically most powerful unbiased test*)



Metabolic Profiling in Bladder Cancer

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

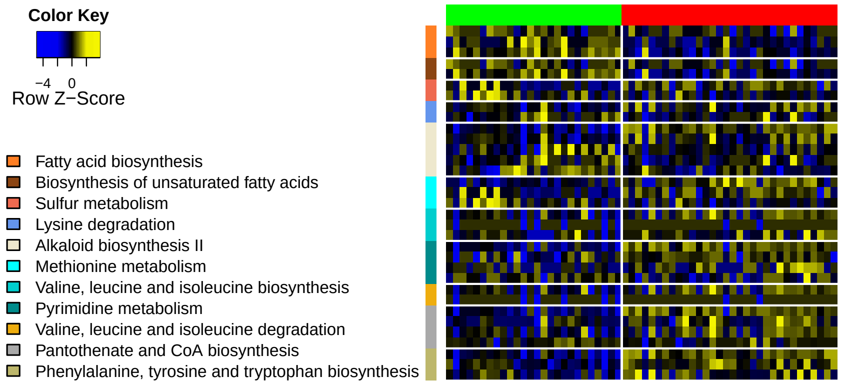
- 58 bladder cancer and adjacent benign samples
- Pathways information obtained from KEGG



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

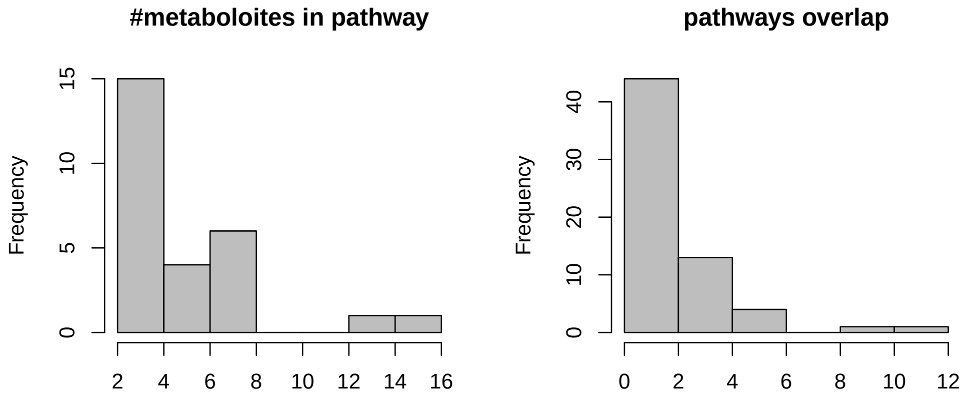
Metabolic Profiling in BCa

- 63 metabolites identified, mapped to 70 pathways
- 27 pathways with at least 3 members



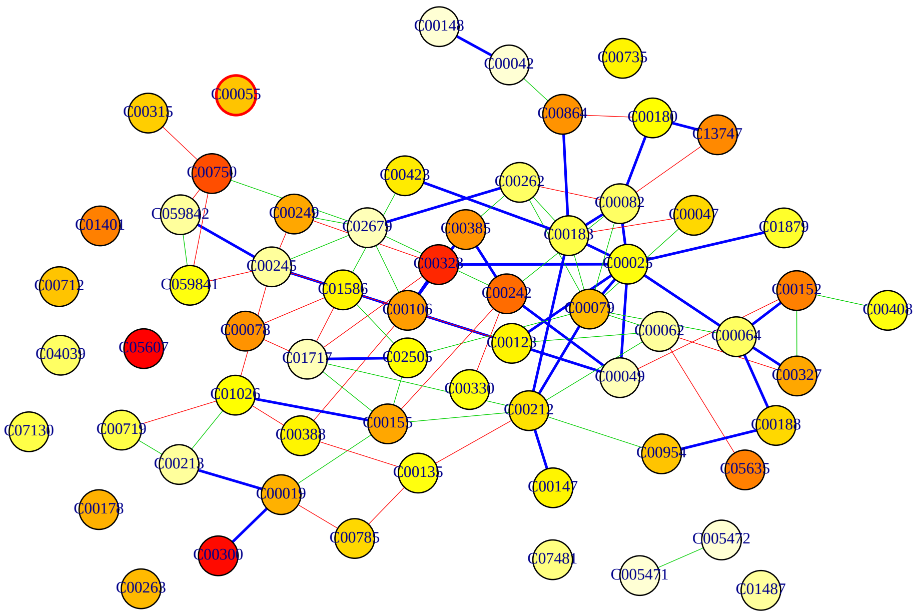
Metabolic Profiling in BCa

- Small pathway sizes & significant overlap among pathways



- Existing methods may not work well...

Metabolic Interaction Network



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, also observed by *Putluri et al* (2012)

R package `netgsa`

- Basic usage:

```
NetGSA(A1, A2, x, y, B)
```

- `A1,A2`: $p \times p$ weighted adjacency matrices for condition 1 and 2 (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
- In the current version, only two conditions are supported (e.g. cancer vs. normal); extension for multiple conditions will be released (hopefully) soon
- The code above takes weighted `A1, A2` as input. However, the package includes functions that allow you to enter a (partial) edge list as input, and estimate `A1, A2` for the case of undirected 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]

[1] Ideker & Krogan (2012)