Pathway & Network Analysis of Omics Data: Undirected Graphical Models

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

Summer Institute for Statistical Genetics – Australia, 2017

An Overview of Network Reconstruction Methods

Two general classes of network reconstruction methods:

An Overview of Network Reconstruction Methods

Two general classes of network reconstruction methods:

- Methods based on marginal measures of association:
  - Co-expression networks (linear association)
  - Methods based on mutual information (non-linear associations)

An Overview of Network Reconstruction Methods

Two general classes of network reconstruction methods:

- Methods based on marginal measures of association:
  - Co-expression networks (linear association)
  - Methods based on mutual information (non-linear associations)
- Methods based on conditional measures of association:
  - Methods assuming multivariate normality/linearity
  - Generalizations to allow for nonlinear dependencies

Co-Expression/Correlation Networks

Co-Expression/Correlation Networks

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

Co-Expression/Correlation Networks

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

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.

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

Limitations of Co-Expression Networks

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

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!
- 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.

Limitations of Co-Expression Networks

- The estimation is highly dependent on the choice of τ.
  - We can instead test $H_0 : r_{xy} = 0$

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

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

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

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

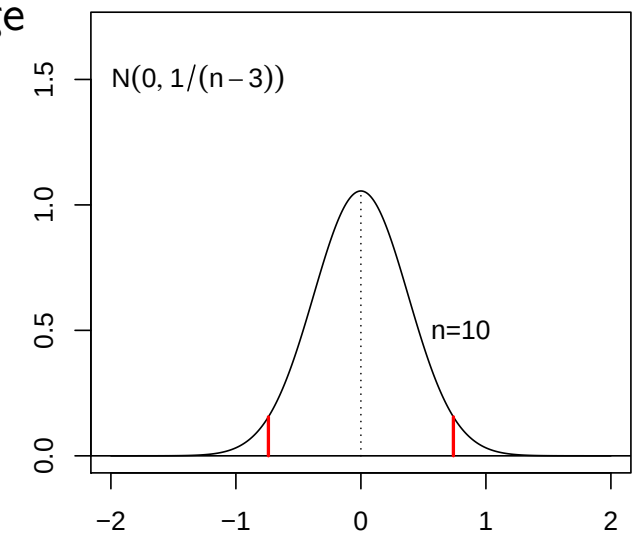
    - Reject $H_0 : r_{xy} = 0$ if $|Z|$ is large

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

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

  - Reject $H_0 : r_{xy} = 0$ if $|Z|$ is large

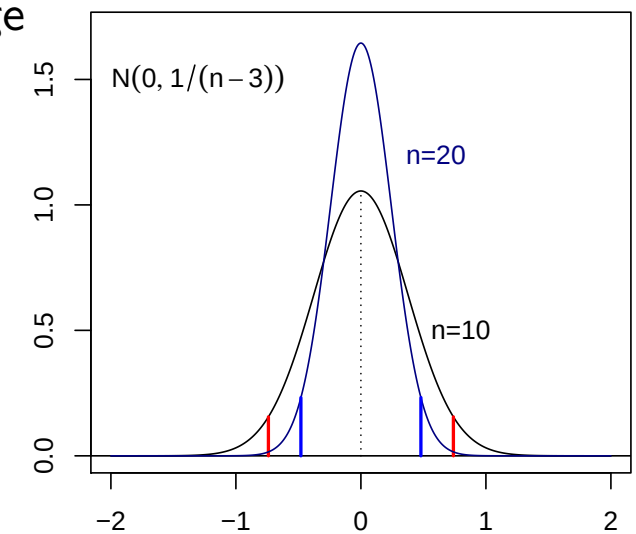


Limitations of Co-Expression Networks

- 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

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

  - Reject $H_0 : r_{xy} = 0$ if $|Z|$ is large

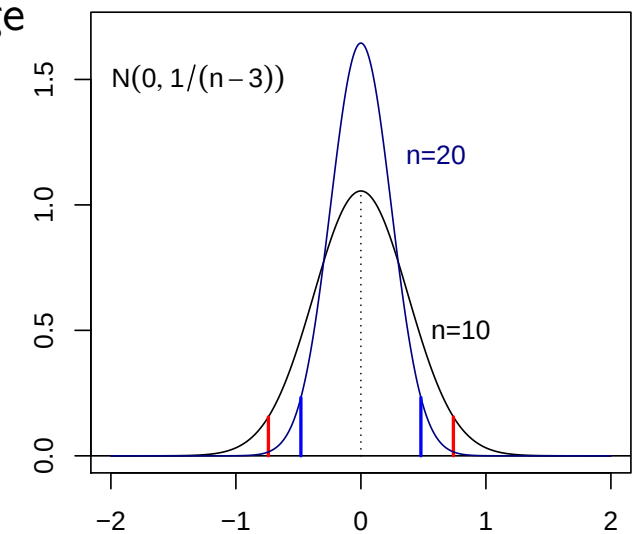


Limitations of Co-Expression Networks

- 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

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

  - Reject $H_0 : r_{xy} = 0$ if $|Z|$ is large



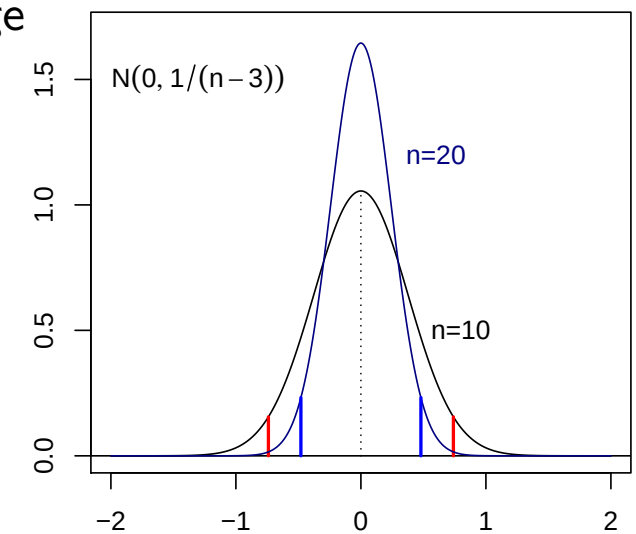
  - Many tests for large p — adjust for multiple comparisons

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

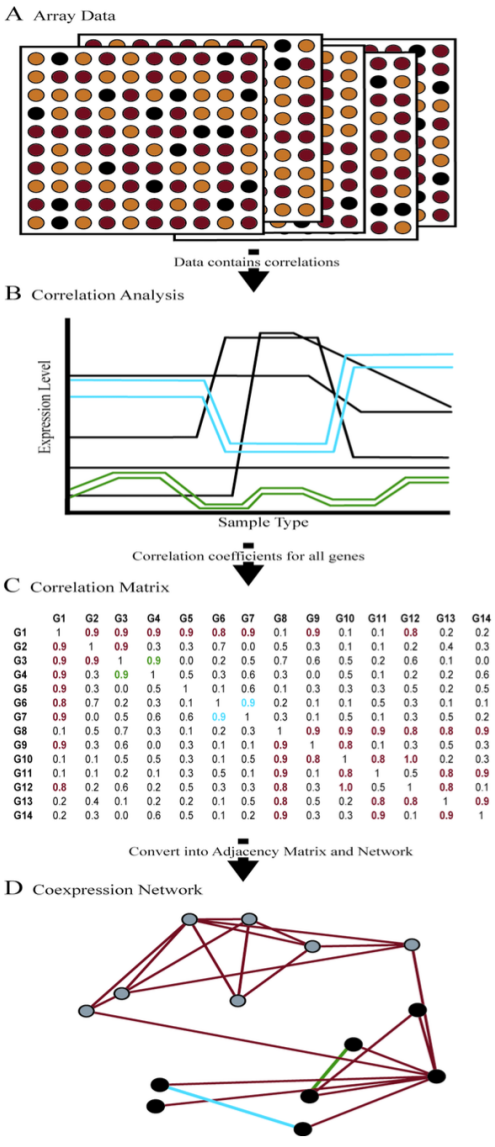
$$Z = \frac{1}{2} \ln \left(\frac{1+r}{1-r} \right) = \operatorname{artanh}(r) \sim_{H_0} N\left(0, \frac{1}{\sqrt{n-3}}\right)$$

  - Reject $H_0 : r_{xy} = 0$ if $|Z|$ is large



  - Many tests for large p — adjust for multiple comparisons
- Alternatively, can use "weighted" co-expression networks

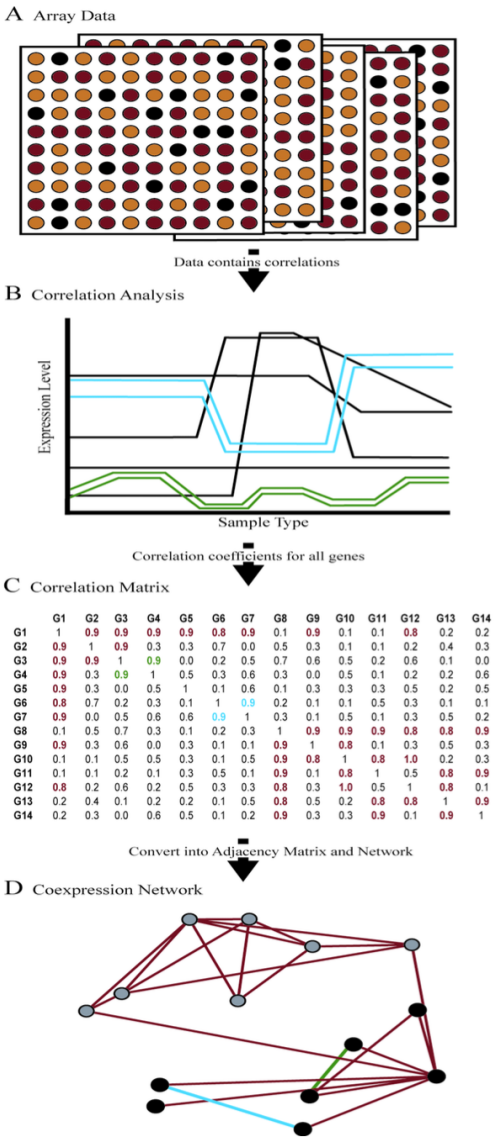
Weighted Gene Co-expression Network Analysis[1]



	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

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

Weighted Gene Co-expression Network Analysis[1]

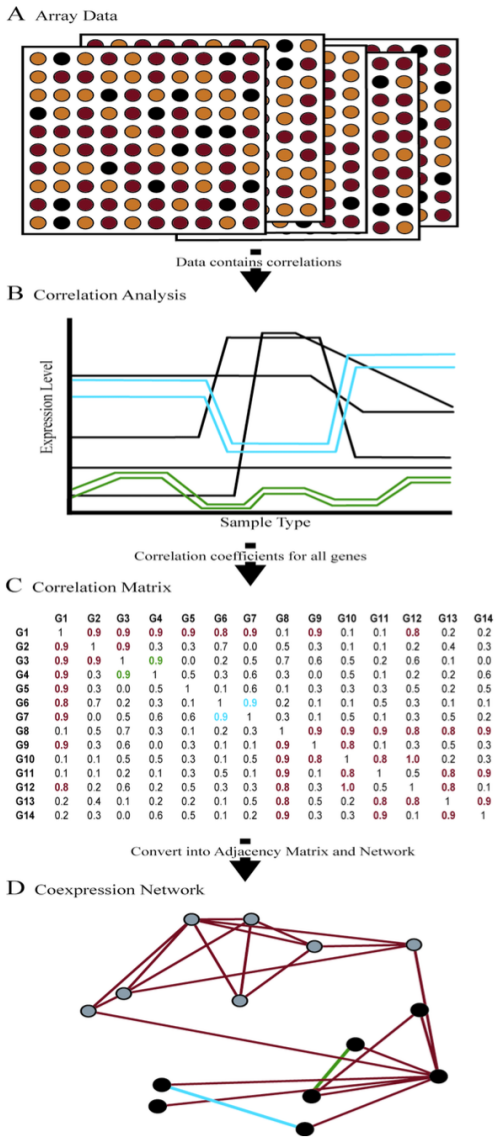


	G1	G2	G3	G4	G5	G6	G7	G8	G9	G10	G11	G12	G13	G14
G1	1	0.9	0.9	0.9	0.9	0.8	0.9	0.1	0.9	0.1	0.1	0.8	0.2	0.2
G2	0.9	1	0.9	0.3	0.3	0.7	0.0	0.5	0.3	0.1	0.1	0.2	0.4	0.3
G3	0.9	0.9	1	0.9	0.0	0.2	0.5	0.7	0.6	0.5	0.2	0.6	0.1	0.0
G4	0.9	0.3	0.9	1	0.5	0.3	0.6	0.3	0.0	0.5	0.1	0.2	0.2	0.6
G5	0.9	0.3	0.0	0.5	1	0.1	0.6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

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

Weighted Gene Co-expression Network Analysis[1]



- Measure concordance of gene expression using Pearson correlation
- Continuously transform the Pearson correlations into an (soft) adjacency function $\rightarrow$ weighted network
  - using the sigmoid adjacency function

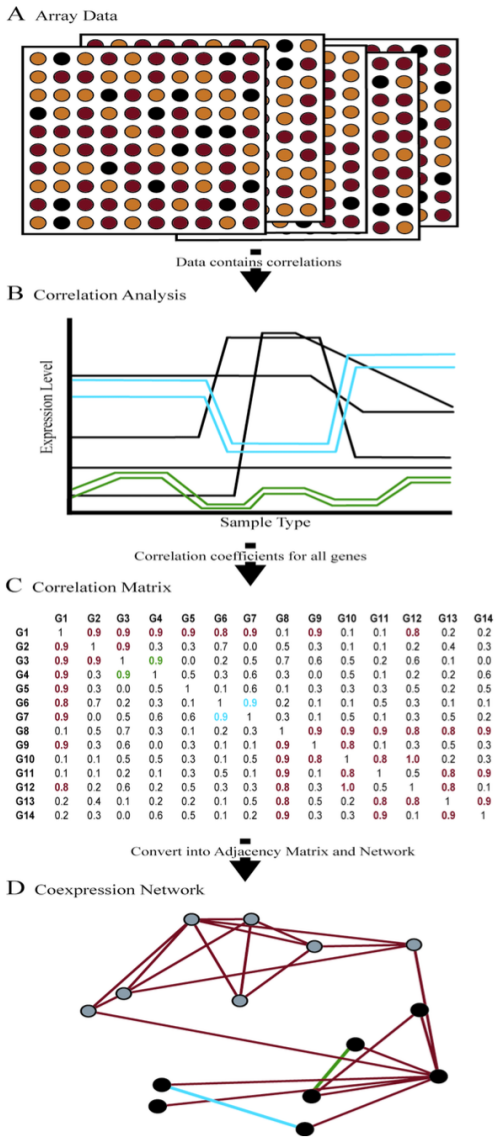
$$A_{ij} = \frac{1}{1 + e^{-\alpha(r_{ij} - \tau_0)}}$$

  - using the power adjacency function

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

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

Weighted Gene Co-expression Network Analysis[1]



- Measure concordance of gene expression using Pearson correlation
- Continuously transform the Pearson correlations into an (soft) adjacency function $\rightarrow$ weighted network
  - using the sigmoid adjacency function

$$A_{ij} = \frac{1}{1 + e^{-\alpha(r_{ij} - \tau_0)}}$$

  - using the power adjacency function

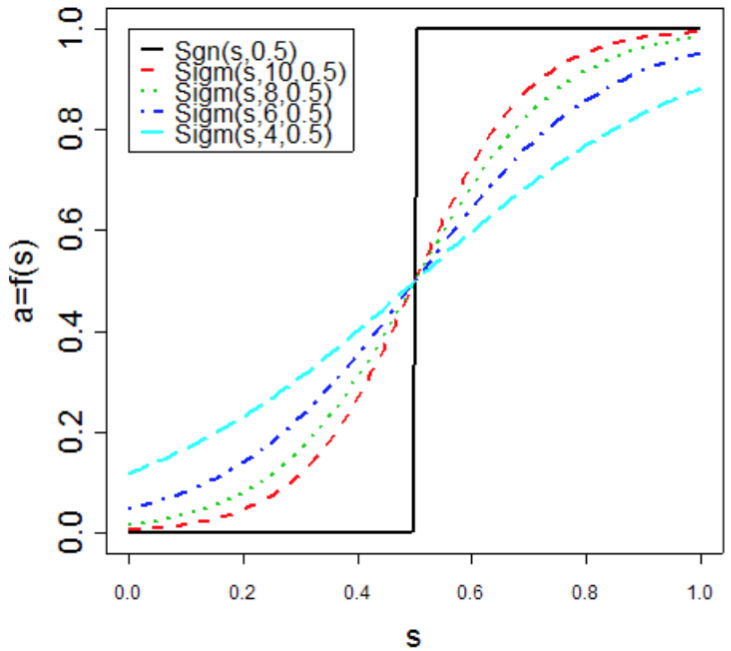
$$A_{ij} = |r_{ij}|^{\beta}$$

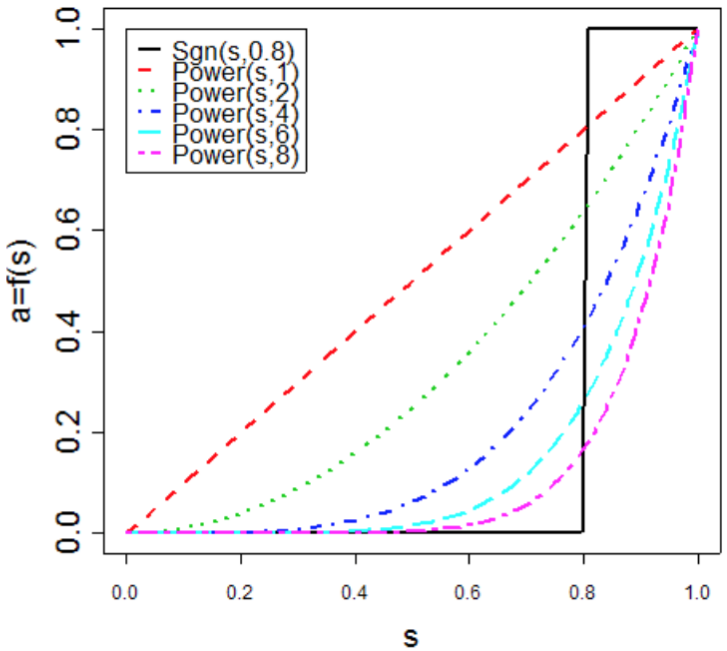
- Perform downstream network analysis (clustering, etc) on weighted networks

[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

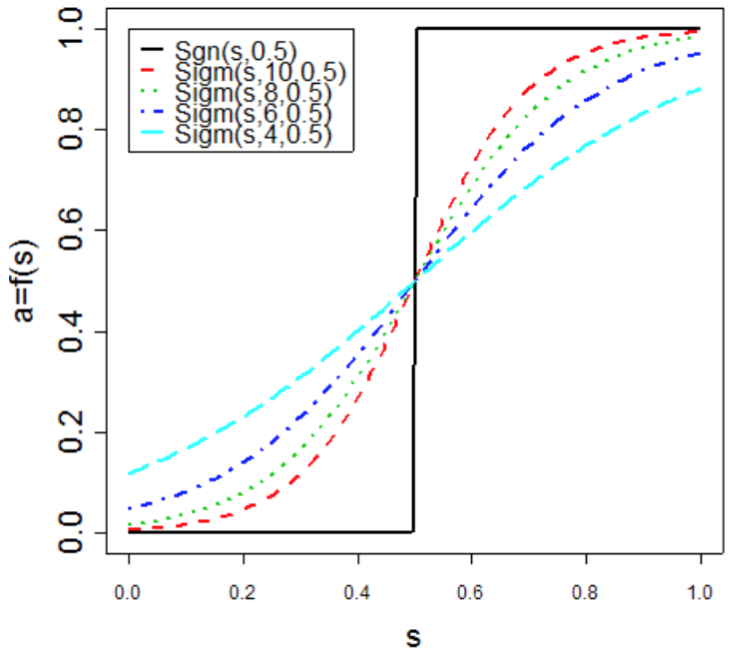
- By changing the tuning parameters, adjacency functions behave similar to hard thresholding

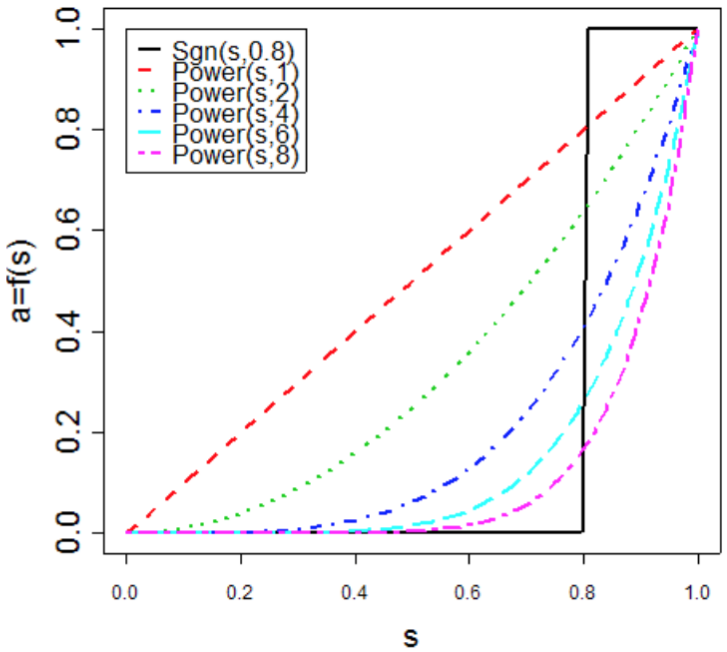




Choice of Parameters

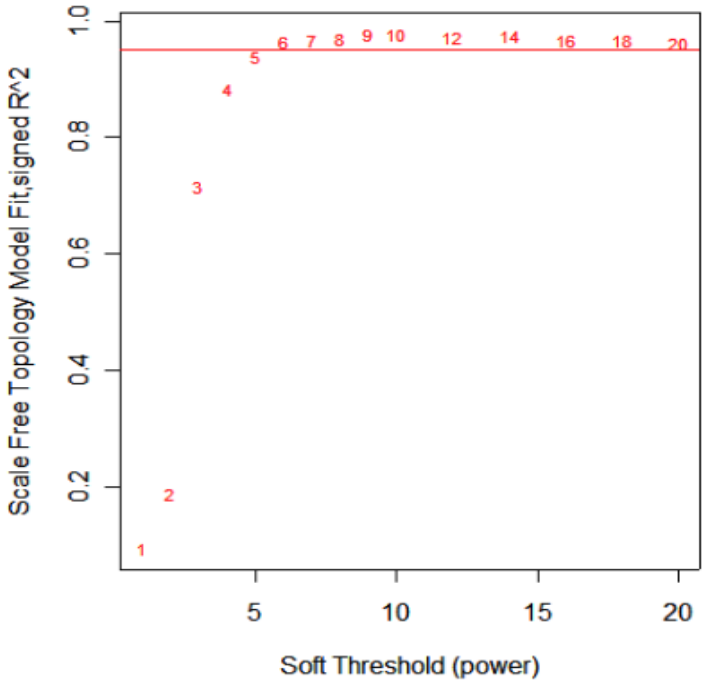
- By changing the tuning parameters, adjacency functions behave similar to hard thresholding

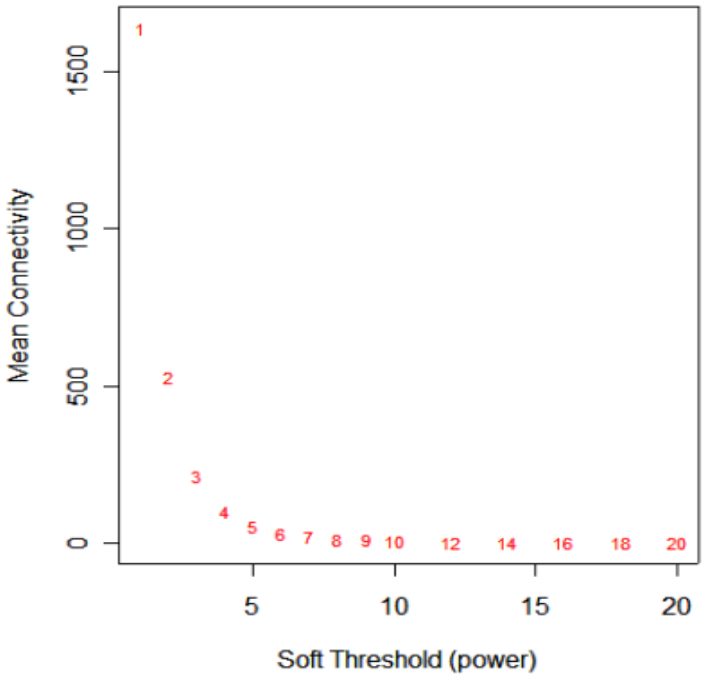




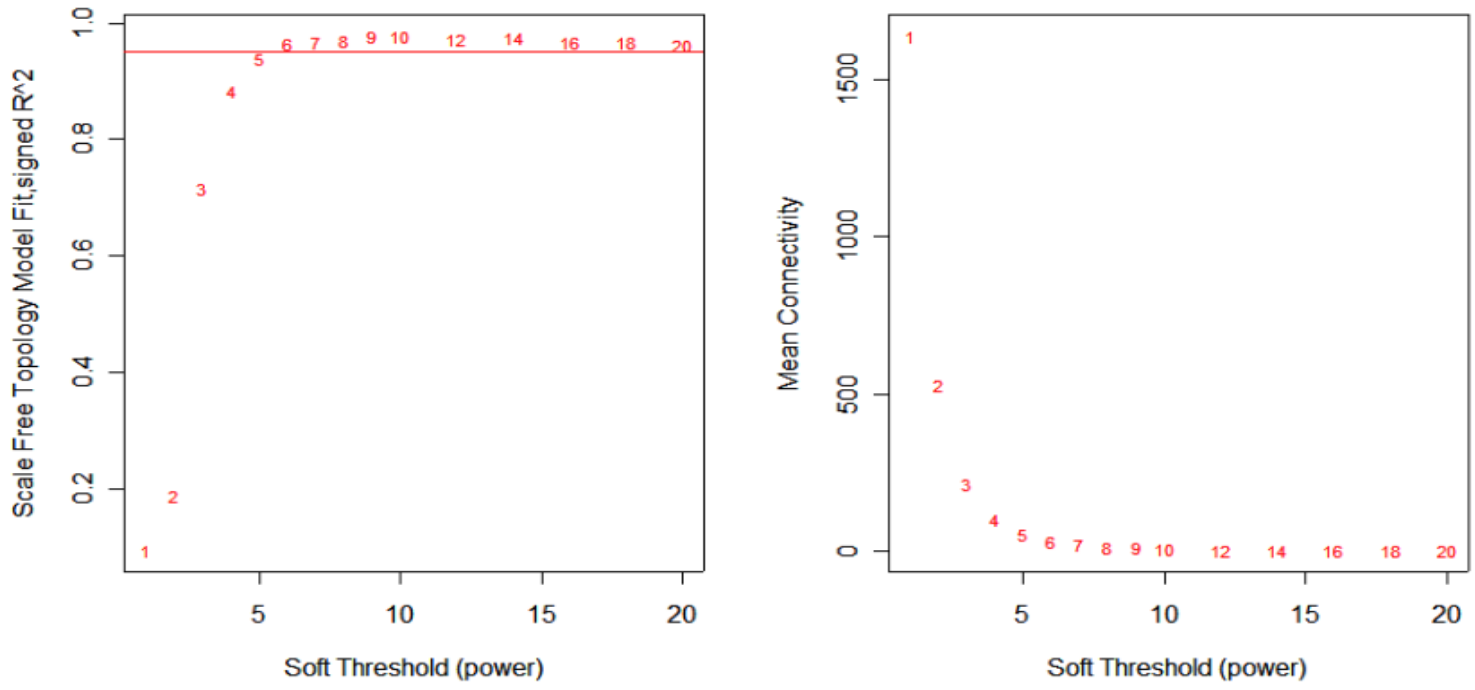
- Power and sigmoid adjacency functions lead to similar results if the parameters are chosen to achieve scale-free topology
- We focus on power adjacency function

Choice of Parameters





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'")
```

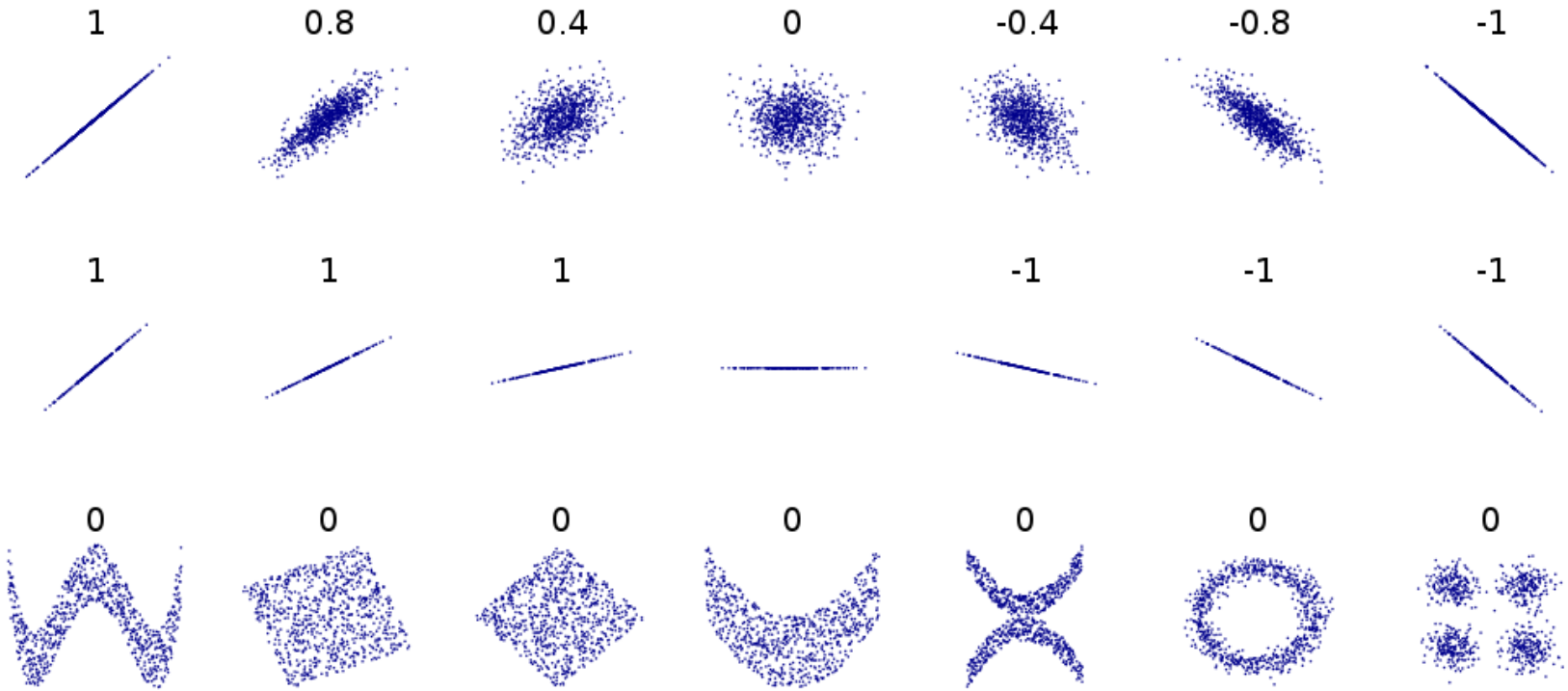
- To determine the power so that the network has scale-free distribution, need to search for multiple powers

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



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

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]

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

ARACNE: Algorithm for the Reconstruction of Accurate Cellular NEtworks[2]

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

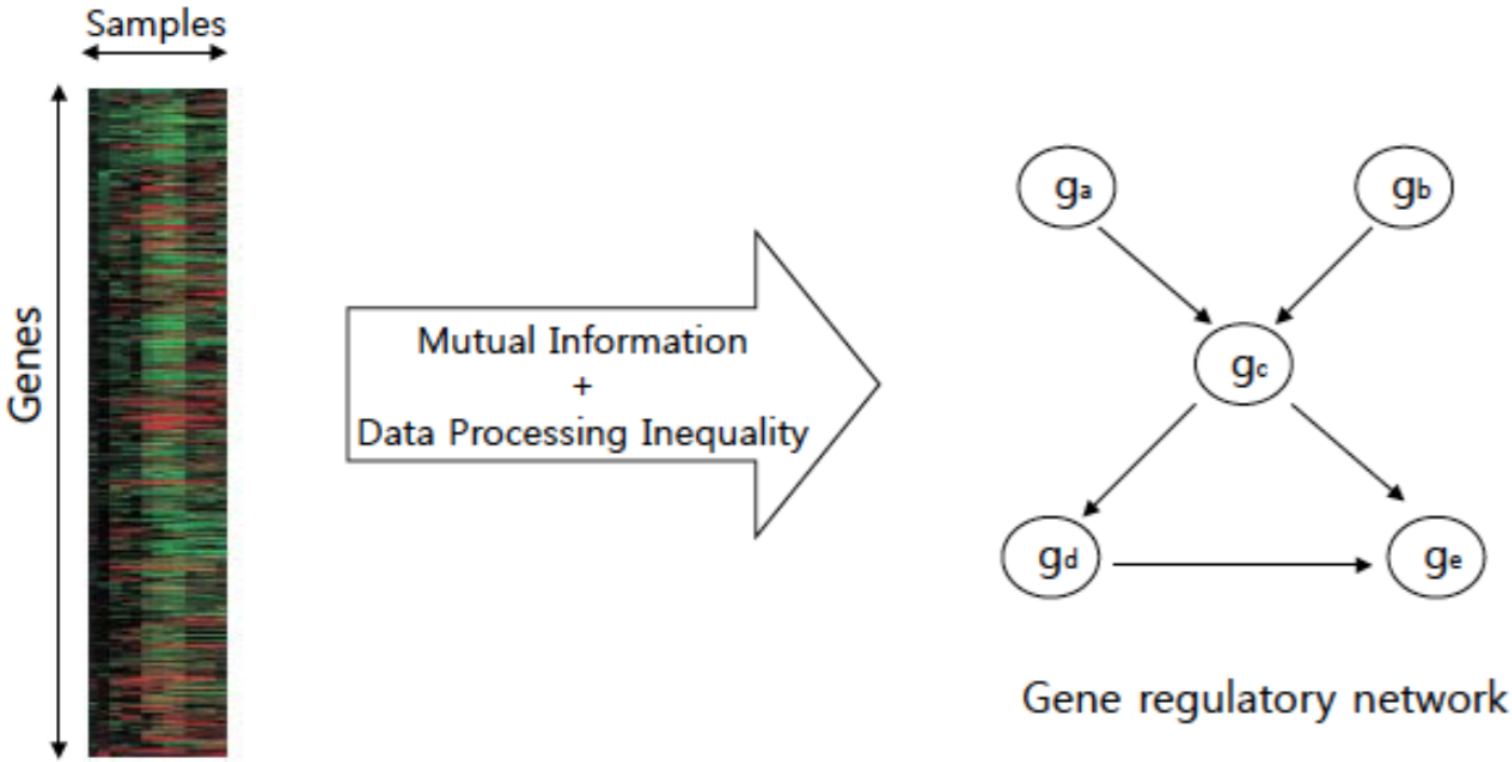
[2] ARACNE: An algorithm for the reconstruction of gene regulatory networks in a mammalian cellular context, Margolin et al, BMC Bioinfo, 2006

ARACNE: Algorithm for the Reconstruction of Accurate Cellular NEtworks[2]

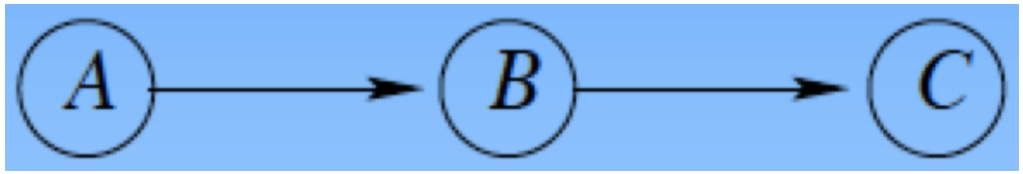
1. Identifies statistically significant gene-gene co-regulation based on mutual information
2. It then eliminates indirect relationships in which two genes are co-regulated through one or more intermediates

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

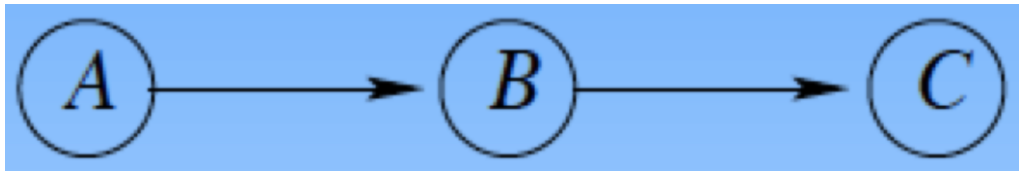
ARACNE


Samples
Genes
Mutual Information
+
Data Processing Inequality
g_a
g_b
g_c
g_d
g_e
Gene regulatory network

Data Processing Inequality (DPI)



Data Processing Inequality (DPI)

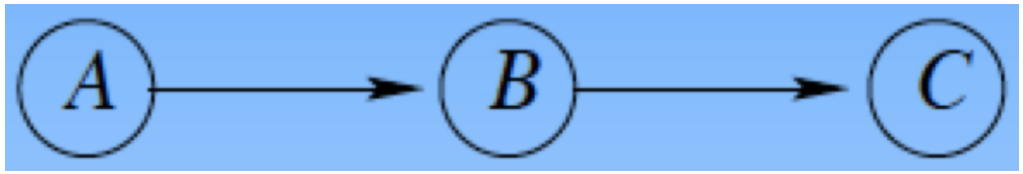


$$I(A, C) \leq min[I(A, B), I(B, C)]$$

where

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

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

Algorithm Details

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

Algorithm Details

- Starts with a network where each triplet of genes is connected by an edge.
- The algorithm then examines each gene triplet for which all pairwise MIs are greater than a cut-off and removes the edge with the smallest value based on DPI.

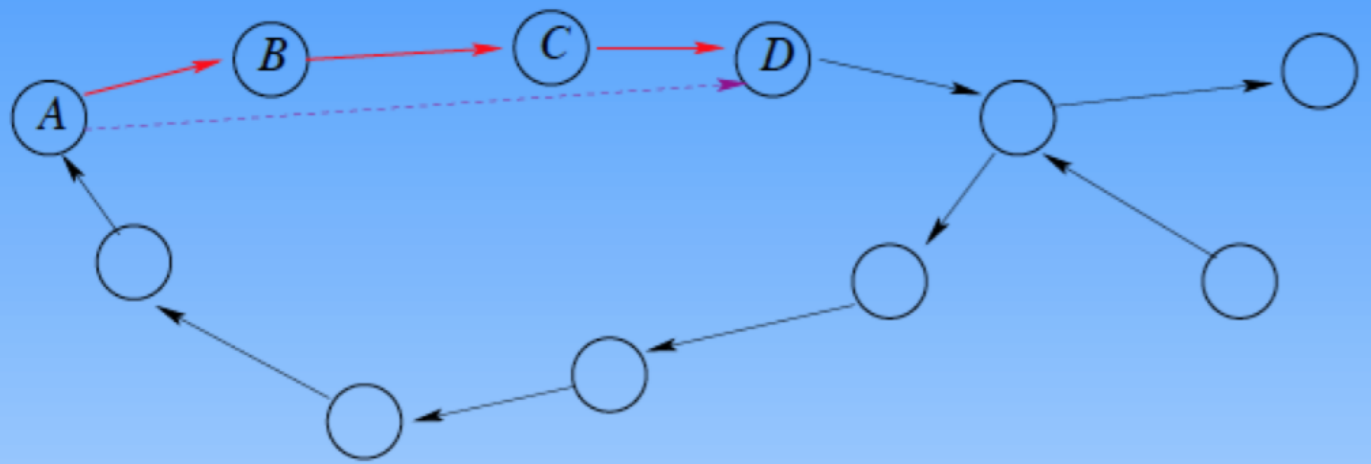
Algorithm Details

- 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

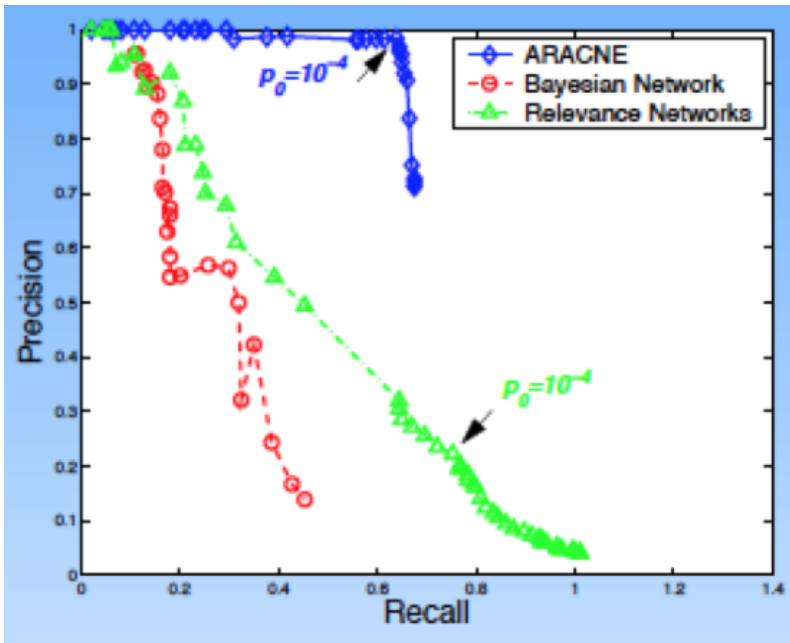
- If MIs can be estimated with no errors, then ARACNE reconstructs the underlying interaction network exactly, provided this network is a tree and has only pairwise interactions.
- The maximum MI spanning tree is a subnetwork of the network built by ARACNE.

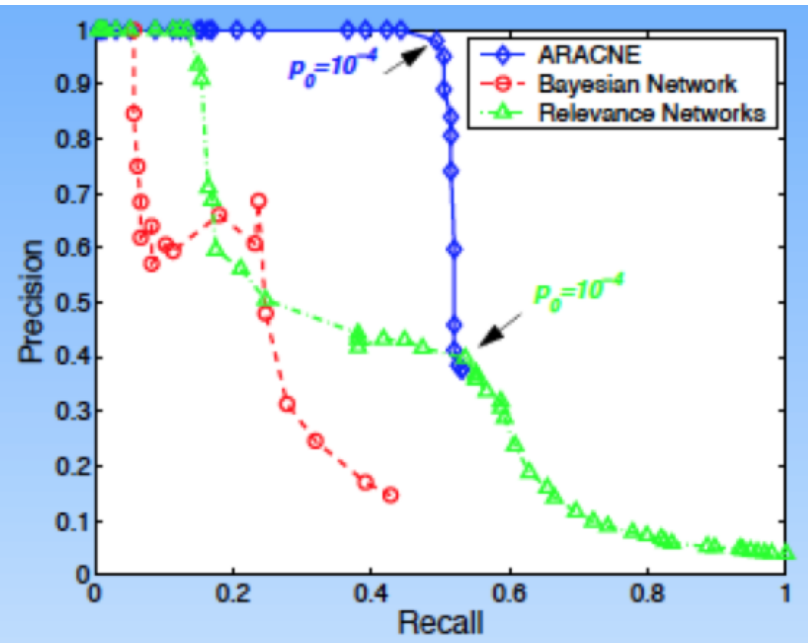
Rationale and Guarantees



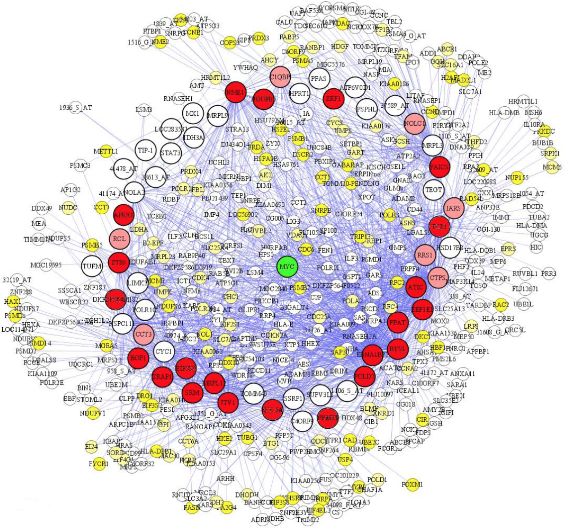
Theorem. Let π_{ik} be the set of nodes forming the shortest path in the network between nodes i and k. Then, if MIs can be estimated without errors, ARACNE reconstructs an interaction network without false positives edges, provided: (a) the network consists only of pairwise interactions, (b) for each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

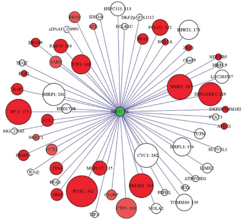
Performance on Synthetic Data





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")
```

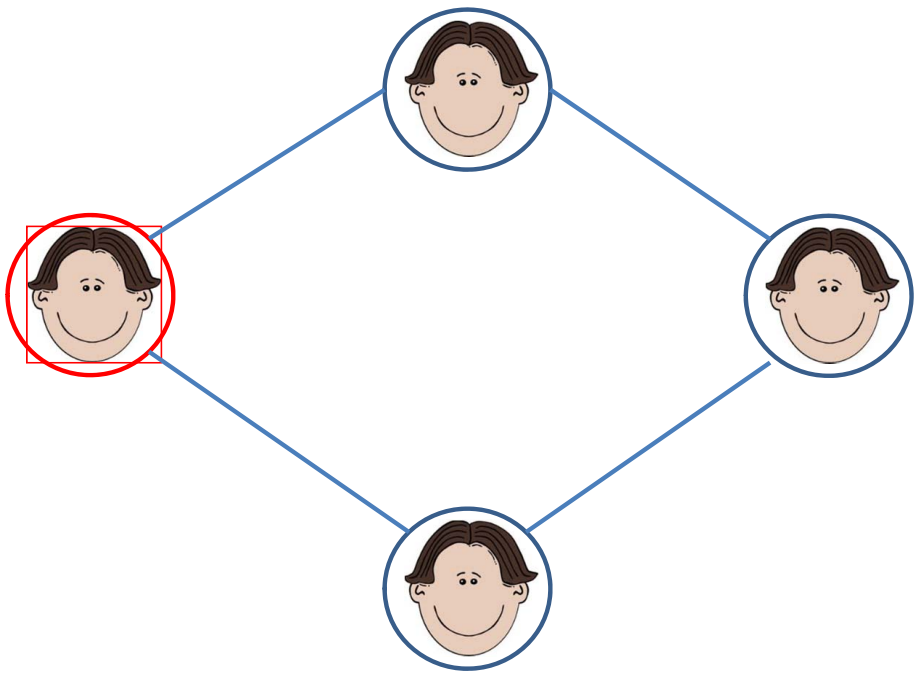
  - eps: threshold for setting an edge to zero, prior to searching over triplets

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!

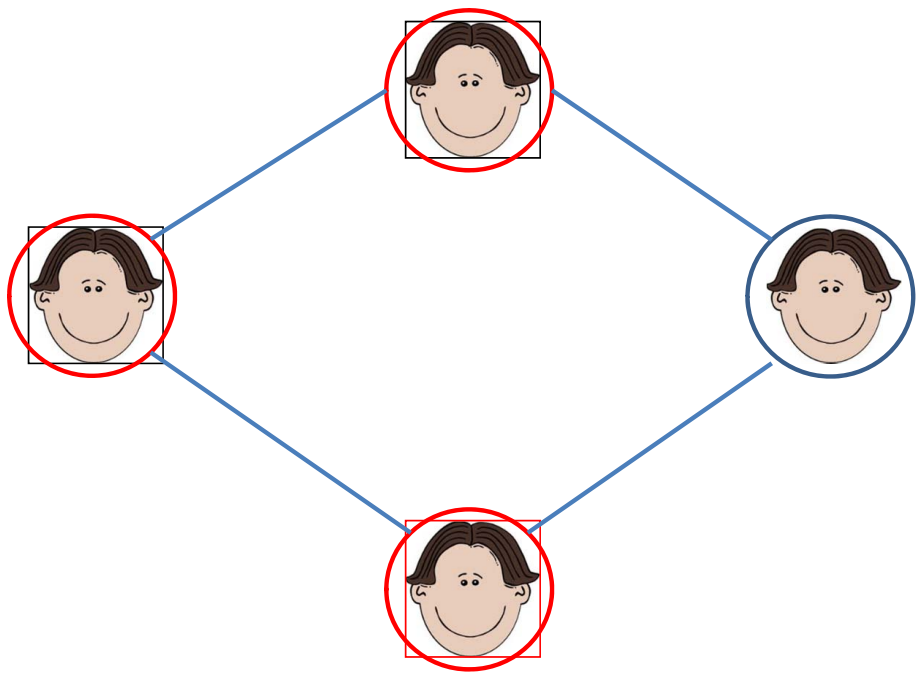
Limitations of Co-Expression Networks

- 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!



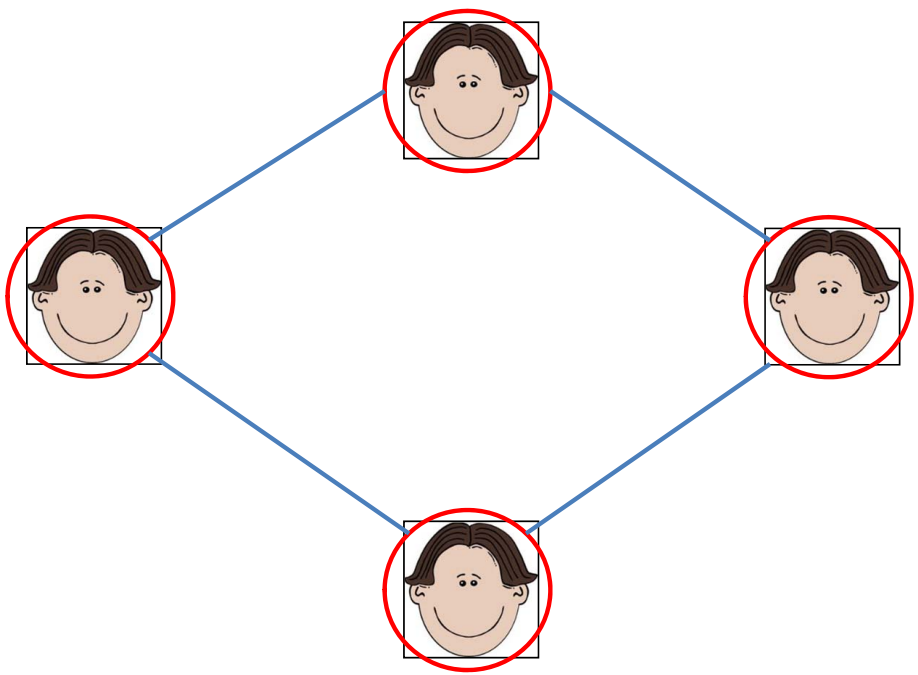
Limitations of Co-Expression Networks

- 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!



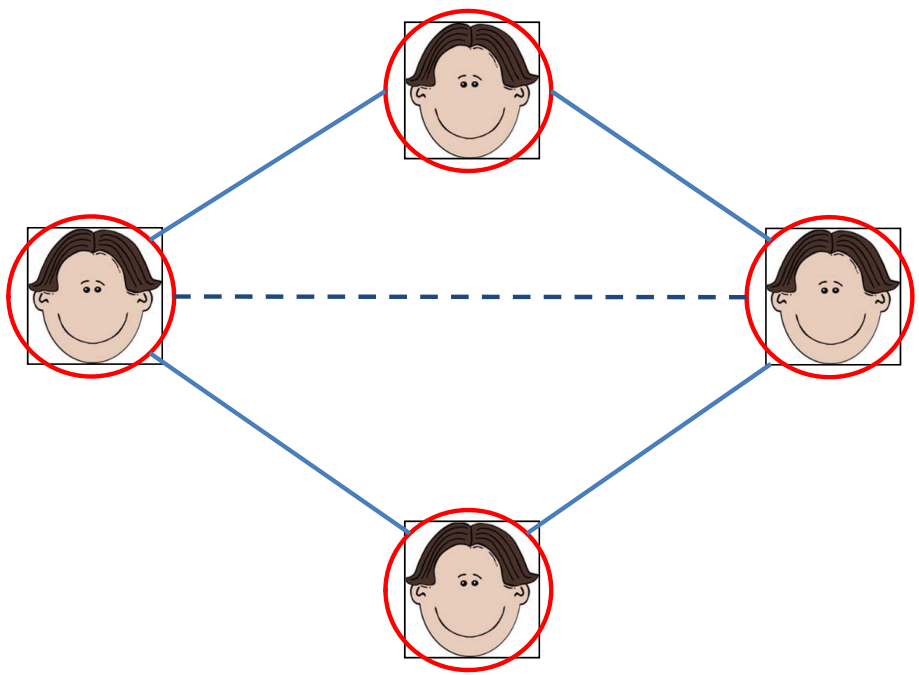
Limitations of Co-Expression Networks

- 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!



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

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

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*.

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.

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

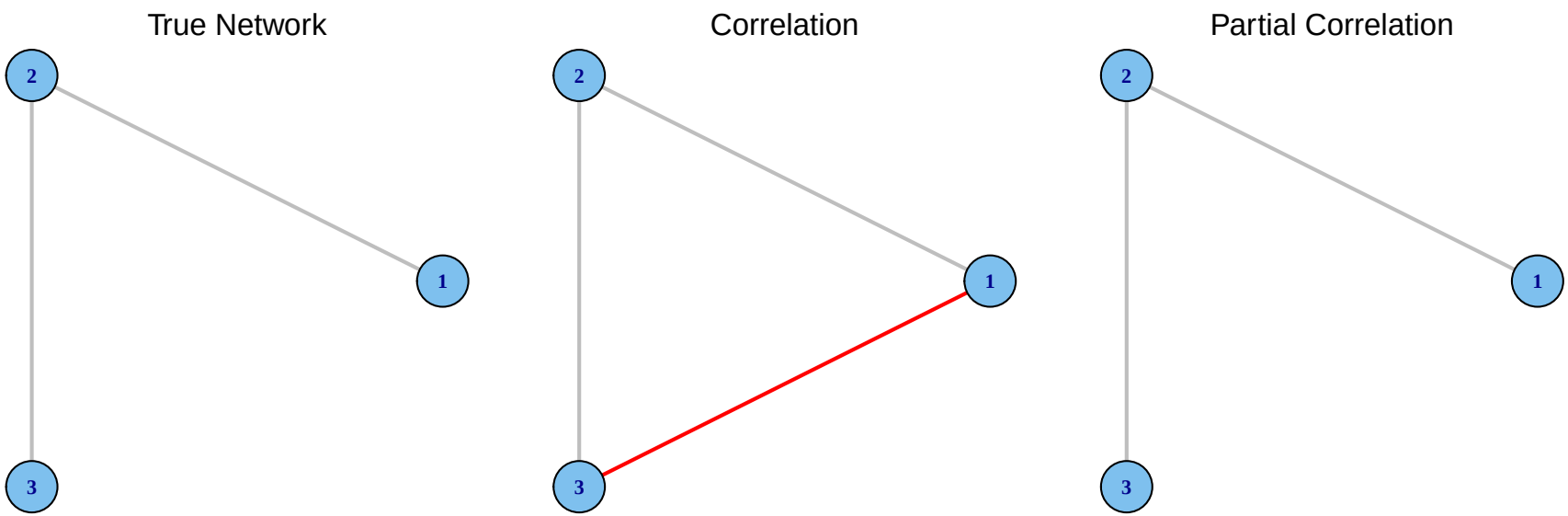
- Partial correlation is also symmetric
- Partial correlation is also a number between -1 and 1
- In partial correlation networks, we draw an edge between X and Y, if the partial correlation between them is large
- Calculation of partial correlation is more difficult
- Again, we can determine this using 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

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

A simple example

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



A larger example

A larger example

- A network with 10 nodes and 20 edges

A larger example

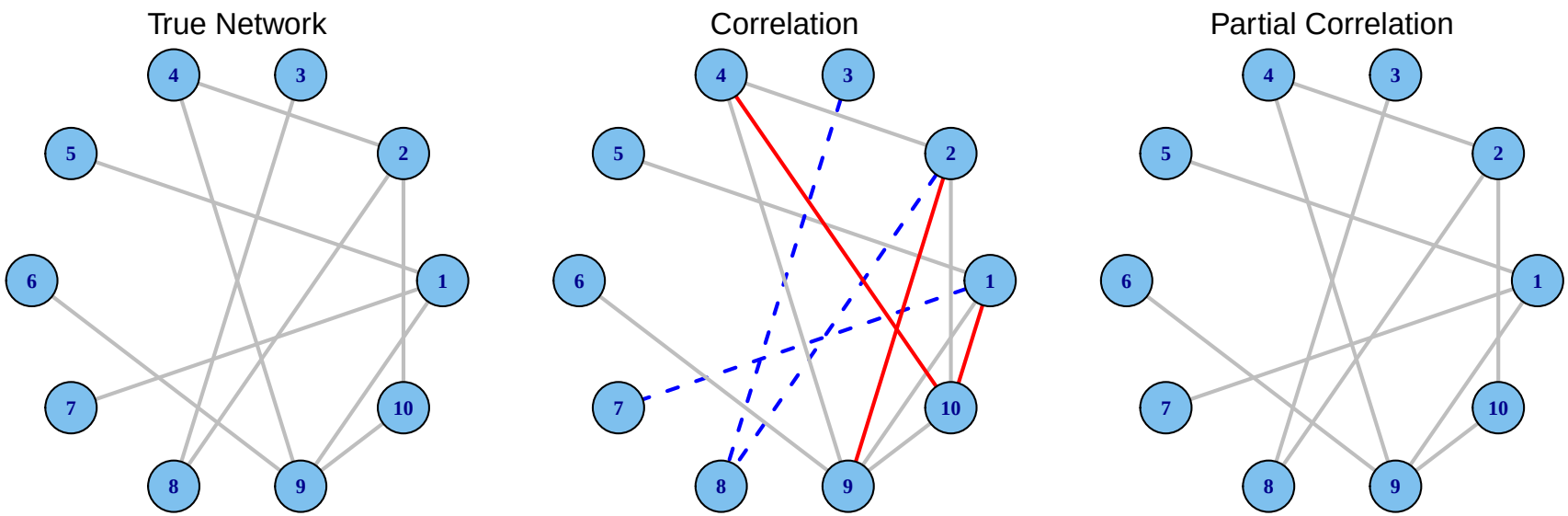
- A network with 10 nodes and 20 edges
- $n = 100$ observations

A larger example

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

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

Partial Correlation for Gaussian Random Variables

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

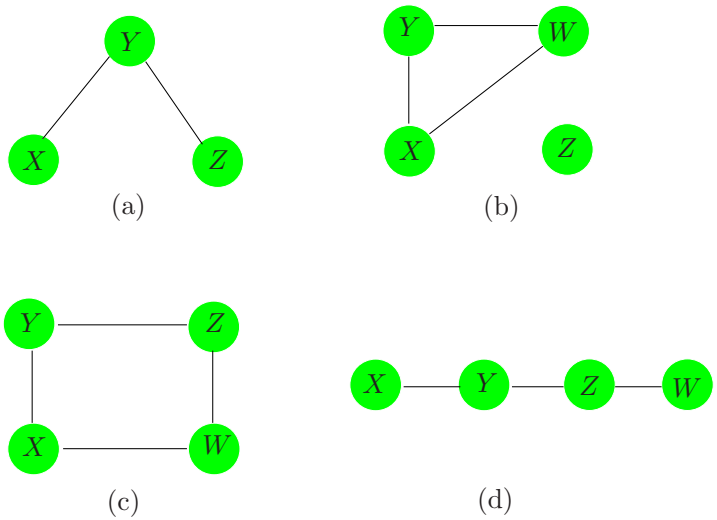
Partial Correlation for Gaussian Random Variables

- It turns out, we can calculate the partial correlation between X_i and X_j given all other variables, by calculating the inverse of the empirical covariance matrix S.
- In other words, the (i, j) entry in Σ^{-1} gives the partial correlation between X_i and X_j given all other variables $X_{\setminus i,j}$.

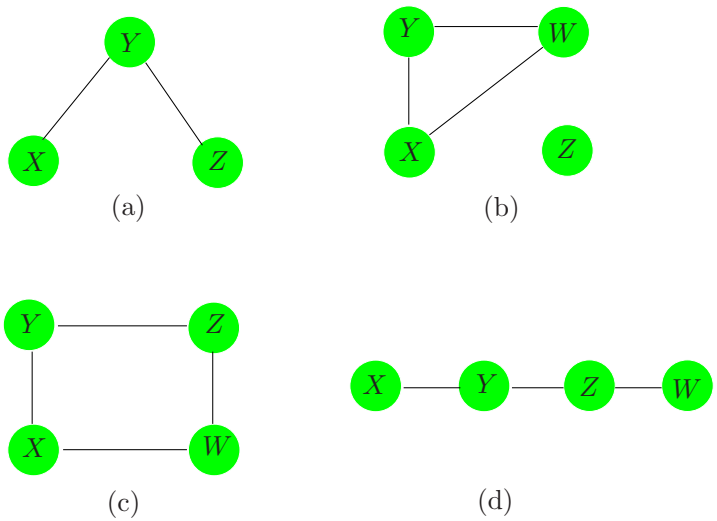
Partial Correlation for Gaussian Random Variables

- 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 $X_{\setminus i,j}$.
- Now suppose the variables are connected by a graph G, then if $X \sim N(0, \Sigma)$, the nonzero entries in the inverse covariance matrix correspond to the edges of G: $(i, j) \in E$ iff $\Sigma_{ij}^{-1} \neq 0$

Partial Correlation for Gaussian Random Variables



Partial Correlation for Gaussian Random Variables



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

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

Estimation

Therefore, to estimate the edges in the graph G,

Estimation

Therefore, to estimate the edges in the graph G,

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

Estimation

Therefore, to estimate the edges in the graph G,

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

Estimation

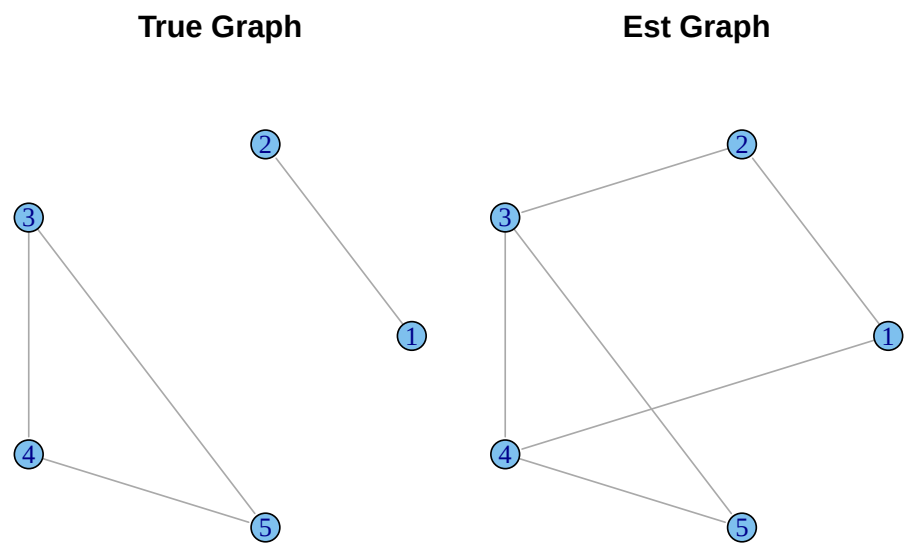
Therefore, to estimate the edges in the graph G,

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

Estimation

Therefore, to estimate the edges in the graph G,

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

Difficulties in HD

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

Difficulties in HD

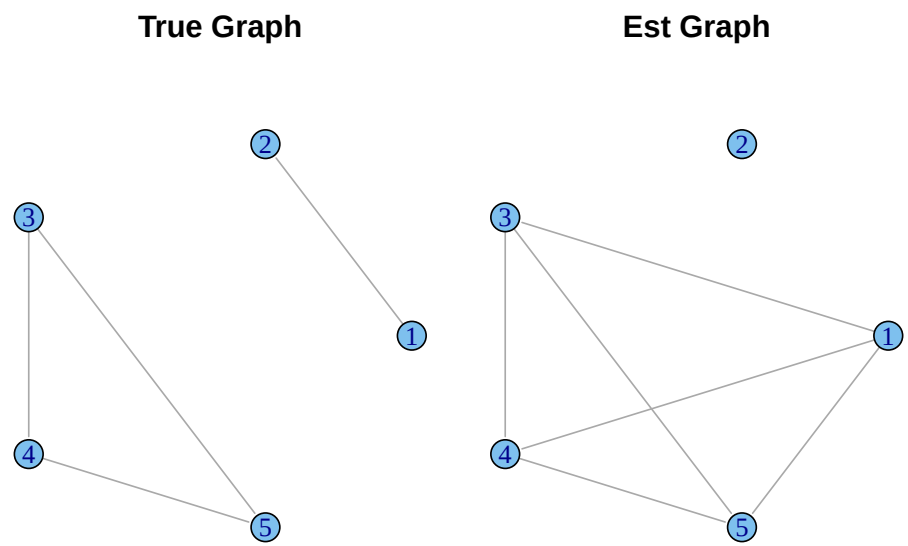
- A number of problems arise in high dimensional settings, especially when $p \gg n$.
- First, S is not invertible if $p > n$!

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 more false positives and false negatives.

Difficulties in HD

- A number of problems arise in high dimensional settings, especially when $p \gg n$.
- First, S is not invertible if $p > n$!
- Even if $p < n$, but n is not very large, we may still get poor estimates, and more false positives and false negatives.



Estimation in High Dimensions – Method 1

Estimation in High Dimensions – Method 1

- Idea: estimating partial conditions is equivalent to regressing each variable X_j on all others!

$$X_1 \sim \beta_2 X_2 + \beta_3 X_3 + \cdots + \beta_p X_p$$

Estimation in High Dimensions – Method 1

- Idea: estimating partial conditions is equivalent to regressing each variable X_j on all others!

$$X_1 \sim \beta_2 X_2 + \beta_3 X_3 + \cdots + \beta_p X_p$$

- Problem: but when $p > n$ we the usual regression is not defined!

Estimation in High Dimensions – Method 1

- Idea: estimating partial conditions is equivalent to regressing each variable X_j on all others!

$$X_1 \sim \beta_2 X_2 + \beta_3 X_3 + \cdots + \beta_p X_p$$

- Problem: but when $p > n$ we the usual regression is not defined!
- Solution: use penalized regression...

Digression: A (Very Short) Intro to Penalized Estimation

- Consider a linear regression of an outcome y on a set of variables $X_1, \ldots, X_p$

$$y = \sum_{j=1}^{p} X_j \beta_j + \epsilon_j$$

Digression: A (Very Short) Intro to Penalized Estimation

- Consider a linear regression of an outcome y on a set of variables $X_1, \ldots, X_p$

$$y = \sum_{j=1}^{p} X_j \beta_j + \epsilon_j$$

- The 'classical' method for estimating the coefficients β_j is the least squares (LS) method, which minimizes

$$\text{minimize}_{\beta_1,\ldots,\beta_p} \|y - \sum_j X_j \beta_j\|^2 = \sum_{i=1}^{n} \left(y_i - \sum_j X_{j,i} \beta_j \right)^2$$

Digression: A (Very Short) Intro to Penalized Estimation

- Consider a linear regression of an outcome y on a set of variables $X_1, \ldots, X_p$

$$y = \sum_{j=1}^{p} X_j \beta_j + \epsilon_j$$

- The 'classical' method for estimating the coefficients β_j is the least squares (LS) method, which minimizes

$$\text{minimize}_{\beta_1,\ldots,\beta_p} \|y - \sum_j X_j \beta_j\|^2 = \sum_{i=1}^{n} \left(y_i - \sum_j X_{j,i} \beta_j\right)^2$$

- Unfortunately, LS does not work when $p > n$ – we cannot obtain sensible estimates of β_js as LS results in overfitting

Digression: A (Very Short) Intro to Penalized Estimation

- Consider a linear regression of an outcome y on a set of variables $X_1, \ldots, X_p$

$$y = \sum_{j=1}^{p} X_j \beta_j + \epsilon_j$$

- The 'classical' method for estimating the coefficients β_j is the least squares (LS) method, which minimizes

$$\text{minimize}_{\beta_1,\ldots,\beta_p} \|y - \sum_j X_j \beta_j\|^2 = \sum_{i=1}^{n} \left(y_i - \sum_j X_{j,i} \beta_j\right)^2$$

- Unfortunately, LS does not work when $p > n$ – we cannot obtain sensible estimates of β_js as LS results in overfitting
- In penalized regression, a penalty is added to the LS objective to prevent overfitting and control the model complexity

$$\text{minimize}_{\beta_1,\ldots,\beta_p} \|y - \sum_j X_j \beta_j\|^2 + \text{Penalty}(\beta_1, \ldots, \beta_p)$$

A Useful Penalty: The Lasso

- In Lasso the penalty is sum of absolute values of coefficients

$$\text{minimize}_{\beta_1,\ldots,\beta_p} \|y - \sum_j X_j \beta_j\|^2 + \lambda \sum_j |\beta_j|$$

where λ is a nonnegative tuning parameter that controls the amount of penalty and hence the model complexity.

A Useful Penalty: The Lasso

- In Lasso the penalty is sum of absolute values of coefficients

$$\text{minimize}_{\beta_1,\ldots,\beta_p} \|y - \sum_j X_j \beta_j\|^2 + \lambda \sum_j |\beta_j|$$

where λ is a nonnegative tuning parameter that controls the amount of penalty and hence the model complexity.

  - When $\lambda = 0$, we get least squares.
  - When λ is very large, we get $\hat{\beta}_j = 0$ for all j.

A Useful Penalty: The Lasso

- In Lasso the penalty is sum of absolute values of coefficients

$$\text{minimize}_{\beta_1,\ldots,\beta_p} \|y - \sum_j X_j \beta_j\|^2 + \lambda \sum_j |\beta_j|$$

  where λ is a nonnegative tuning parameter that controls the amount of penalty and hence the model complexity.
  - When $\lambda = 0$, we get least squares.
  - When λ is very large, we get $\hat{\beta}_j = 0$ for all j.
- The reason that lassi is useful is that for intermediate values of λ some coefficients would be exactly zero!

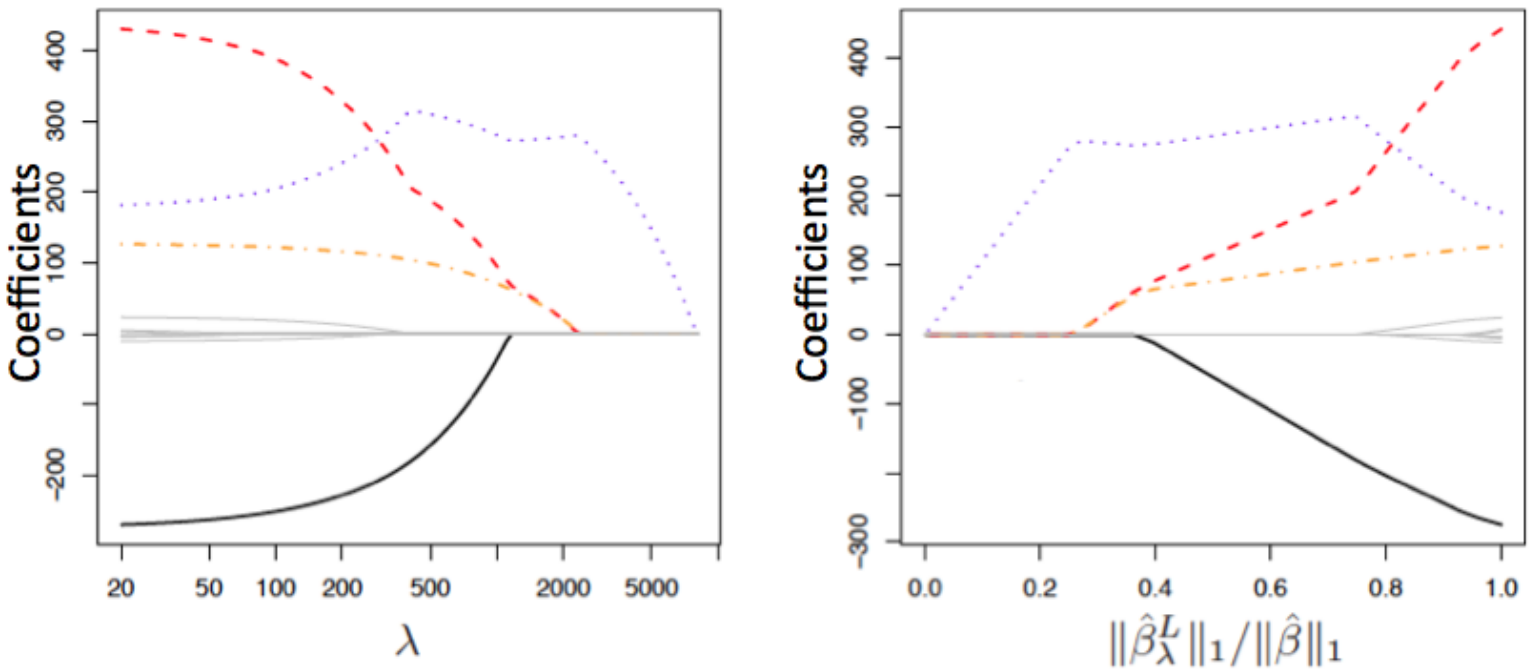
A Useful Penalty: The Lasso

- In Lasso the penalty is sum of absolute values of coefficients

$$\text{minimize}_{\beta_1,\ldots,\beta_p} \|y - \sum_j X_j \beta_j\|^2 + \lambda \sum_j |\beta_j|$$

  where λ is a nonnegative tuning parameter that controls the amount of penalty and hence the model complexity.
  - When $\lambda = 0$, we get least squares.
  - When λ is very large, we get $\hat{\beta}_j = 0$ for all j.
- The reason that lassi is useful is that for intermediate values of λ some coefficients would be exactly zero!
  - Lasso selects a subset of variables!!

Lasso As λ Varies



Lasso In Practice

- Perform lasso for a very fine grid of λ values.
- Use held-out data (e.g. cross-validation) to select the optimal value of λ – that is, the best level of model complexity.
- Perform the lasso on the full data set, using that value of λ.

Lasso In Practice

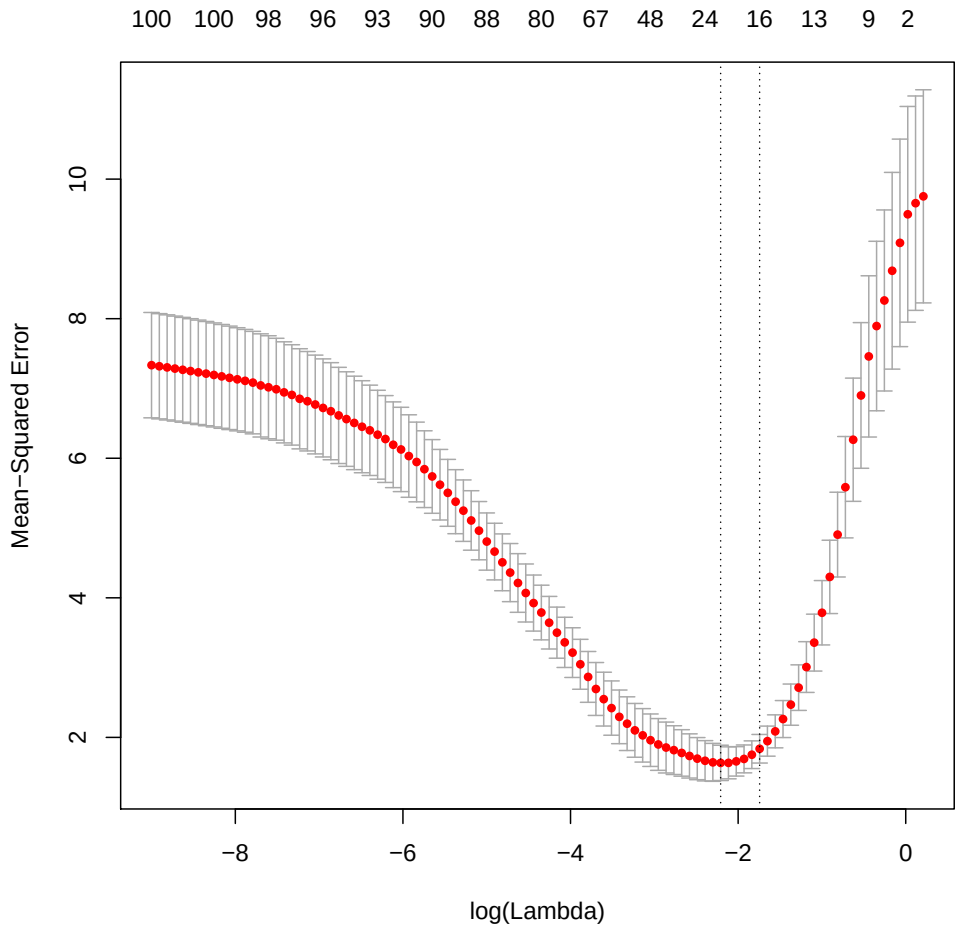
- Perform lasso for a very fine grid of λ values.
- Use held-out data (e.g. cross-validation) to select the optimal value of λ – that is, the best level of model complexity.
- Perform the lasso on the full data set, using that value of λ.
- The idea of lasso is more general and can be used in many other problems

$$\text{Loss}(\beta; X) + \lambda \sum_j |\beta_j|$$

Example in R

```
xtr <- matrix(rnorm(100*100),ncol=100)
beta <- c(rep(1,10),rep(0,90))
ytr <- xtr%*%beta + rnorm(100)
library(glmnet)
cv.out <- cv.glmnet(xtr,ytr,alpha=1,nfolds=5)
print(cv.out$cvm)
plot(cv.out)
cat("CV Errors", cv.out$cvm,fill=TRUE)
cat("Lambda with smallest CV Error",
cv.out$lambda[which.min(cv.out$cvm)],fill=TRUE)
cat("Coefficients", as.numeric(coef(cv.out)),fill=TRUE)
cat("Number of Zero Coefficients",sum(abs(coef(cv.out))<1e-8),
fill=TRUE)
```

R Output


100 100 98 96 93 90 88 80 67 48 24 16 13 9 2
Mean−Squared Error
2
4
6
8
10
−8
−6
−4
−2
0
log(Lambda)

Back to Estimating CIGs in High Dimensions – Method 1

Back to Estimating CIGs in High Dimensions – Method 1

- The idea in the first method, called neighborhood selection, is to estimate the graph by fitting a penalized regression of each variable on all other variables.

Back to Estimating CIGs in High Dimensions – Method 1

- The idea in the first method, called neighborhood selection, is to estimate the graph by fitting a penalized regression of each variable on all other variables.
- In other words, for $j = 1, \ldots, p$, we solve

$$\|X_j - \sum_{k \neq j} X_k \beta_k\|^2 + \lambda \sum_{k \neq j} |\beta_k|$$

Back to Estimating CIGs in High Dimensions – Method 1

- The idea in the first method, called neighborhood selection, is to estimate the graph by fitting a penalized regression of each variable on all other variables.
- In other words, for $j = 1, \ldots, p$, we solve

$$\|X_j - \sum_{k \neq j} X_k \beta_k\|^2 + \lambda \sum_{k \neq j} |\beta_k|$$

- The final estimate of the graph is obtained by getting all of the edges found from these individual regression problems.

Estimating CIGs in High Dimensions – Method 2

Estimating CIGs 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

Estimating CIGs 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
- The log likelihood function of (zero-mean) Gaussian random variables can be written as

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

  where Θ is the $p \times p$ inverse covariance (aka precision) matrix.
- Here, logdet is the logarithm of determinant of matrix; tr is the trace of the matrix, or sum of its diagonal values; and λ is the tuning parameter.

Estimating CIGs 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
- The log likelihood function of (zero-mean) Gaussian random variables can be written as

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

  where Θ is the $p \times p$ inverse covariance (aka precision) matrix.
- Here, logdet is the logarithm of determinant of matrix; tr is the trace of the matrix, or sum of its diagonal values; and λ is the tuning parameter.
- Can estimate Θ by maximizing the penalized log-likelihood:

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

Comparing the Two Approaches

Comparing the Two Approaches

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

Comparing the Two Approaches

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

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

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.

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.

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

A Real Example

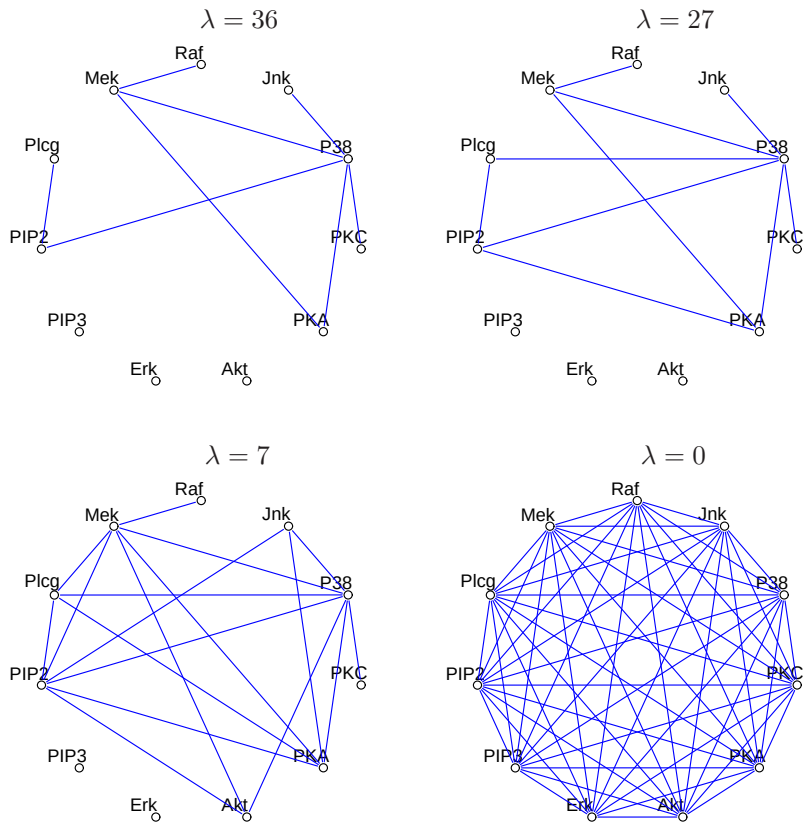
- Flow cytometry allows us to obtain measurements of proteins in individual cells, and hence facilitates obtaining datasets with large sample sizes.

A Real Example

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

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



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)
```

Choice of tuning parameter

Choice of tuning parameter

- Choosing the right λ is very difficult.

Choice of tuning parameter

- Choosing the right λ is very difficult.
- As λ gets larger, we get sparser graphs. However, there is no systematic way of choosing the right λ.

Choice of tuning parameter

- Choosing the right λ is very difficult.
- 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 (ongoing research).

Choice of tuning parameter

- Choosing the right λ is very difficult.
- 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 (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

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.

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**.

Some Comments

- The penalized estimation methods discussed above allow estimation of graphical models in the $p \gg n$ settings, e.g. when p is in 1000's and n is in 100's.
- However, both of these methods, and most other methods for estimation of conditional independence networks, work when the network is **sparse**.
- Sparsity means that there are not many edges in the network, and the network is far from fully connected.

Some Comments

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

Computation

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

Computation

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

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.
- Idea: solve the problem by iterating over p regression problems, one for each column of the precision matrix.

Computation

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

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.
- Idea: solve the problem by 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`.

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.
- Idea: solve the problem by 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, better to use the empirical correlation matrix

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

Marginal vs Conditional Associations

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

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 (especially using neighborhood selection).

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 (especially using neighborhood selection).
- Estimation and inference using marginal associations can be done with much smaller samples

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 (especially using neighborhood selection).
- 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.

Non-linear associations

Non-linear associations

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

Non-linear associations

- Recall that correlation is a measure of linear dependence, this is also true about partial correlation.
- However, many real-world associations are non-linear
- Therefore, (partial) correlation may miss non-linear associations among variables

Non-linear associations

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

Linearity and Normality

[3] Khatri & Rao (1976) & Fisk (1970)

Linearity and Normality

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

[3] Khatri & Rao (1976) & Fisk (1970)

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 $\Rightarrow$ linear relationships
  - linear dependencies (+ extra mild assumptions) $\Rightarrow$ MVN[3]

[3] Khatri & Rao (1976) & Fisk (1970)

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 $\Rightarrow$ linear relationships
  - linear dependencies (+ extra mild assumptions) $\Rightarrow$ MVN[3]
- Both of these are strong assumptions and may not hold in real-world applications!

[3] Khatri & Rao (1976) & Fisk (1970)

Conditional Independence Graphs

Conditional Independence Graphs

- In case of Gaussian variables, $\Theta_{jk} = 0$ implies that X_j and X_k are conditionally independent.

Conditional Independence Graphs

- In case of Gaussian variables, $\Theta_{jk} = 0$ implies that X_j and X_k are conditionally independent.
- Conditional dependence is a general notion that defines the class of conditional independent graphs (CIG). In CIG,

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 neighbors(X): $Z \notin \text{neighbors(X)}$, then $X \perp\!\!\!\perp Z \mid \text{neighbors(X)}$

Nonparanormal (Gaussian Copula) Models

Nonparanormal (Gaussian Copula) Models

- Suppose $X \nsim N(0, \Sigma)$, but there exists monotone functions $f_j, j = 1, \ldots p$ such that $[f_1(X_1), \ldots f_p(X_p)] \sim N(0, \Sigma)$

Nonparanormal (Gaussian Copula) Models

- Suppose $X \approx N(0, \Sigma)$, but there exists monotone functions $f_j, j = 1, \ldots p$ such that $[f_1(X_1), \ldots f_p(X_p)] \sim N(0, \Sigma)$
  - We say that X has a nonparanormal distribution $X \sim NPN_p(f, \Sigma)$.
  - f and 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

Nonparanormal (Gaussian Copula) Models

- Suppose $X \approx N(0, \Sigma)$, but there exists monotone functions $f_j, j = 1, \ldots p$ such that $[f_1(X_1), \ldots f_p(X_p)] \sim N(0, \Sigma)$
  - We say that X has a nonparanormal distribution $X \sim NPN_p(f, \Sigma)$.
  - f and 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: $[f_1(X_1), \ldots f_p(X_p)]$
  ii) estimate the network of the transformed data (e.g. calculate the empirical covariance matrix of the transformed data, and apply glasso or neighborhood selection)

A Related Procedure

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 $x_j^1, \ldots, x_j^n$ and $\bar{r}_j = (n+1)/2$ be the average rank
  - Calculate Spearman's ρ or Kendall's τ

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

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

A Related Procedure

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

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

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

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

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 $x_j^1, \ldots, x_j^n$ and $\bar{r}_j = (n+1)/2$ be the average rank
  - Calculate Spearman's ρ or Kendall's τ

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

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

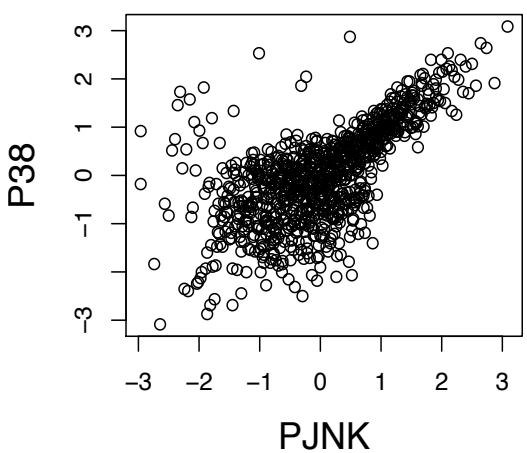
- If $X \sim NPN_p(f, \Sigma)$, then $\Sigma_{jk} = 2\sin(\rho_{jk}\pi/6) = \sin(\tau_{jk}\pi/2)$
- Therefore, we can estimate Σ^{-1} by plugging in rank-based correlations into graphical lasso (R-package `huge`)

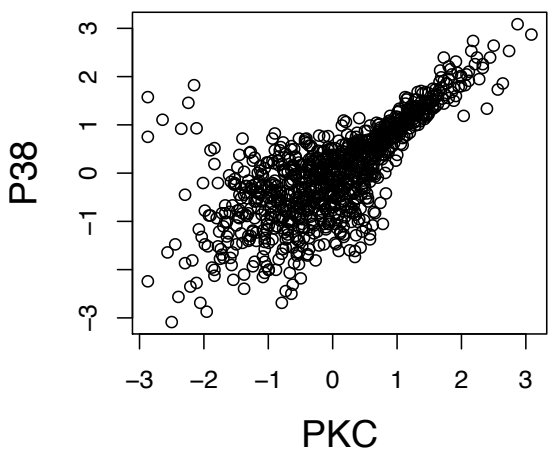
A Real Data Example

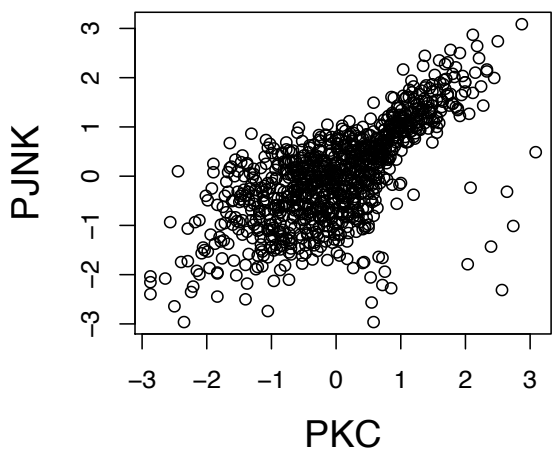
- Protein cytometry data for cell signaling data (Sachs et al, 2005)
- Transform the data using Gaussian copula (Liu et al, 2009), giving marginal normality

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

Graphical Models for Discrete Random Variables

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

Graphical Models for Discrete Random Variables

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

Pairwise Markov Random Fields

Pairwise Markov Random Fields

- The idea of pairwise MRFs is to "assume" that only two-way interactions among variables exist
  - The pairwise MRF associated with the graph G over the random vector X is the family of probability distributions $P(X)$ that can be written as

$$P(X) \propto \exp \sum_{(j,k)\in E} \phi_{jk}(x_j, x_k)$$

  - For each edge $(j, k) \in E$, 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

Graphical Models for Binary Random Variables

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

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

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

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

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

  - 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

[4] Ravikumar et al (2010)

Graphical Models for Binary Random Variables

- We can consider a neighborhood selection[4] approach with an ℓ_1 penalty to find the neighborhood of each node $N(j) = \{k \in V : (j, k) \in E\}$

[4] Ravikumar et al (2010)

Graphical Models for Binary Random Variables

- We can consider a neighborhood selection[4] 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)

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

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

[4] Ravikumar et al (2010)

Graphical Models for Binary Random Variables

- We can consider a neighborhood selection[4] 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)

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

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

[4] Ravikumar et al (2010)

Other Non-Gaussian Distributions

- 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[5]
  - We can even learn networks with multiple types of nodes (gene expression, SNPs, and CNVs)[6]

[5] Yang et al (2012)

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

More Flexible Graphical Models

More Flexible Graphical Models

- As an alternative to parametric graphical models (Gaussian, Ising, Poisson, etc), we can estimate graphical models assume non-parametrically, i.e. without making specific assumptions about the form of the distributions
- A commonly used approach is to assume that interactions are captured by conditional means, and estimate those means non-parametrically. Two examples are
  - SpaCE JAM: models the conditional means using sparse additive models using a group lasso penalty (R-package `spacejam`)
  - GraFo: uses instead random forests to model the conditional means

Summary

Summary

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

Summary

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

Summary

- 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 also difficult.

Summary

- 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 also difficult.
- It is often difficult to validate the estimates; however, in case of biological networks, we can compare our findings with known interactions from literature.