

# Pathway & Network Analysis of Omics Data: Introduction

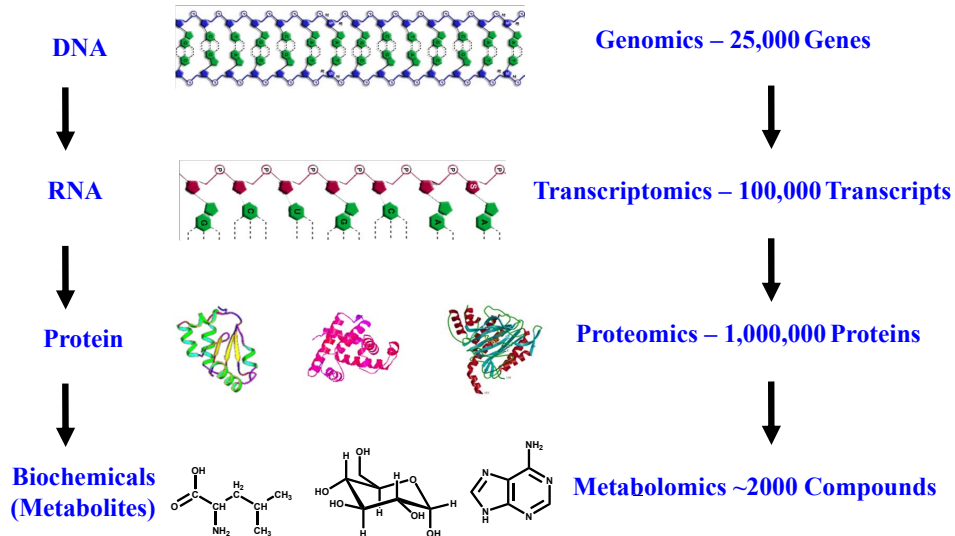
Ali Shojaie  
Department of Biostatistics  
University of Washington  
[faculty.washington.edu/ashojaie](http://faculty.washington.edu/ashojaie)

Summer Institute for Statistical Genetics – 2021

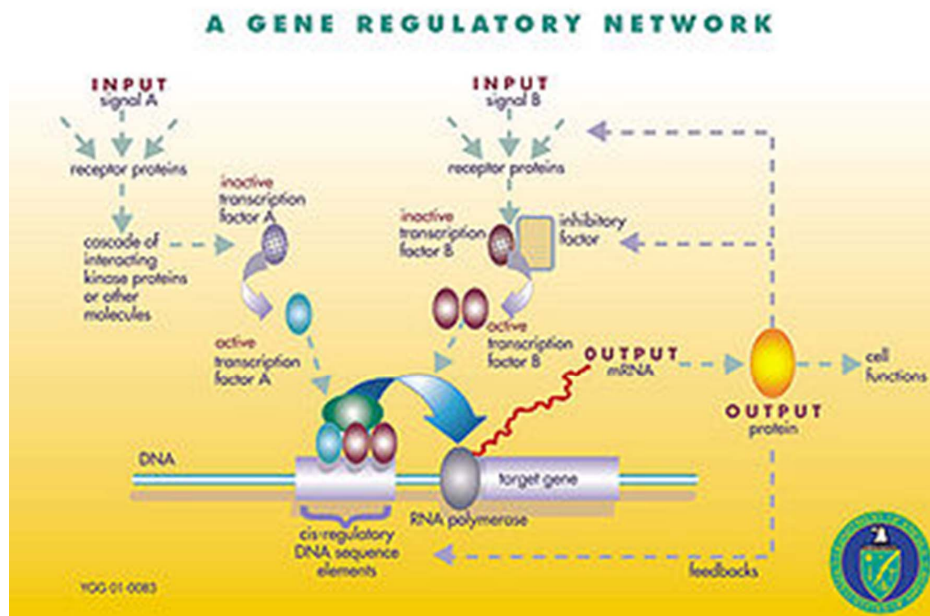
## Why Study Networks?

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

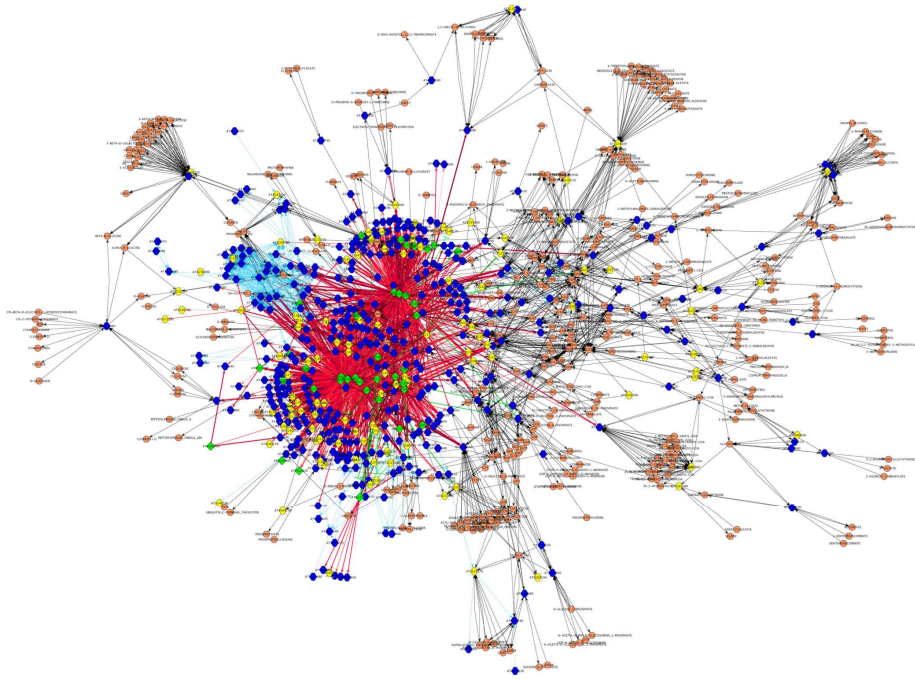
# Central Dogma of Molecular Biology (Extended)



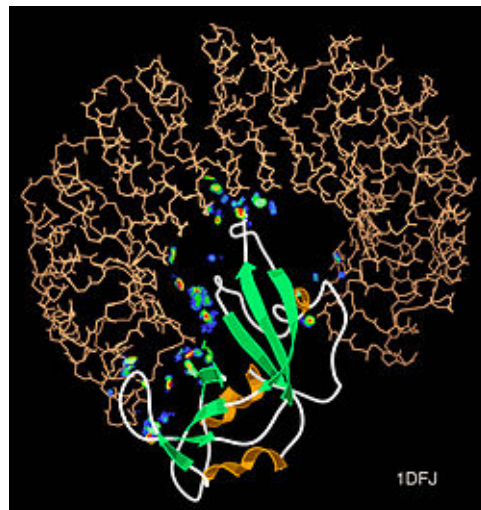
# Networks in Biology: Gene Regulatory Interactions



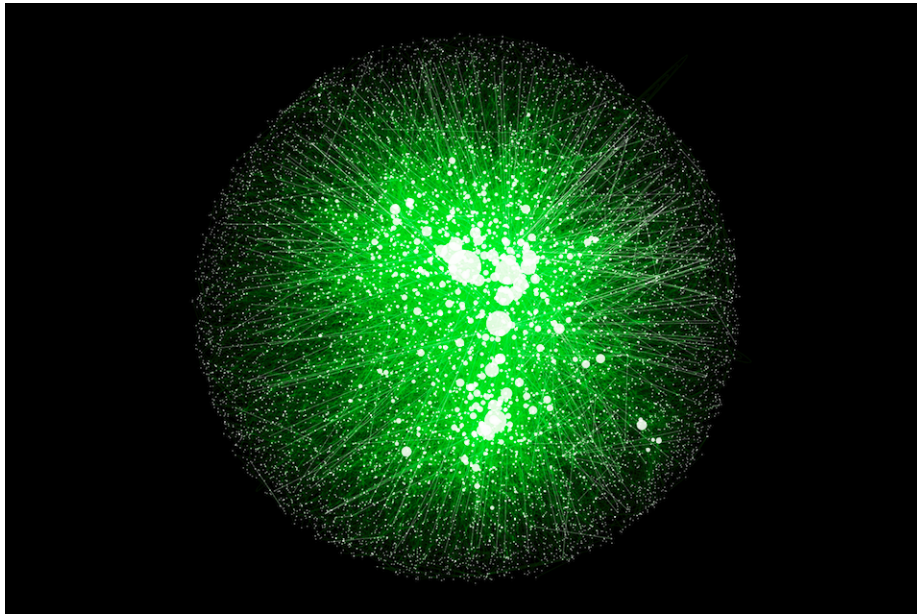
## Networks in Biology: Gene Regulatory Networks



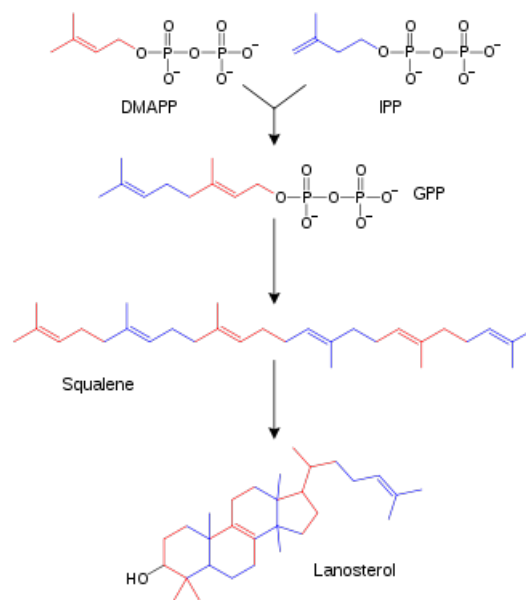
## Networks in Biology: Protein-Protein Interaction



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

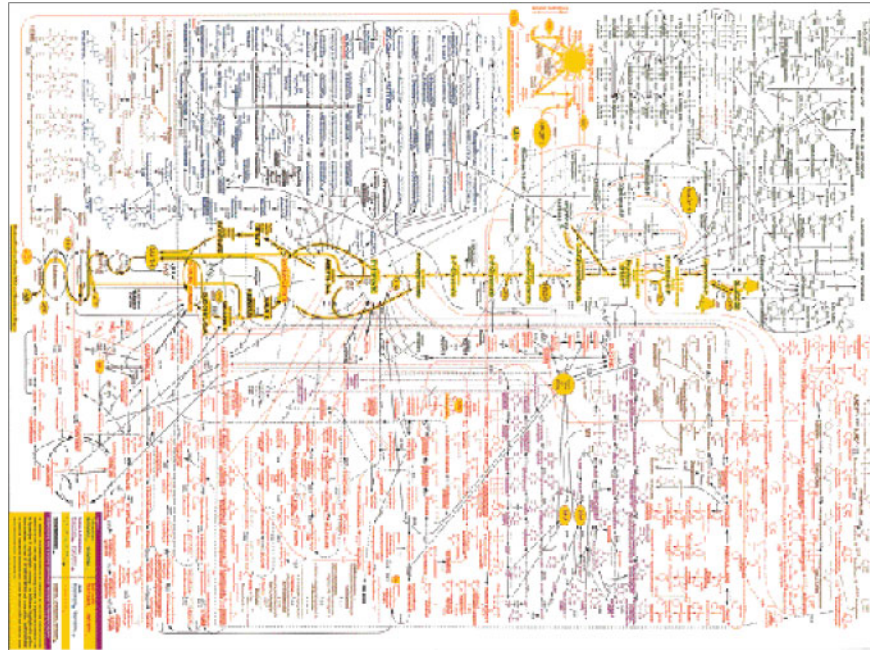


## Networks in Biology: Metabolic Reactions





## 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,<sup>1</sup> Hui Tang,<sup>2</sup> Venugopal Thayanithy,<sup>3</sup> Subbaya Subramanian,<sup>3</sup> Ann L. Oberg,<sup>2</sup> Julie M. Cunningham,<sup>1</sup> James R. Cerhan,<sup>2</sup> Clifford J. Steer,<sup>4</sup> and Stephen N. Thibodeau<sup>1</sup>

<sup>1</sup>Departments of Laboratory Medicine and Pathology and <sup>2</sup>Health Sciences Research, Mayo Clinic, Rochester, Minnesota; and Departments of <sup>3</sup>Laboratory Medicine and Pathology, <sup>4</sup>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?



Cancer Cell  
Article

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

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

<sup>1</sup>Department of Biological Chemistry and Molecular Pharmacology, Harvard Medical School, Boston, MA 02115, USA

<sup>2</sup>Department of Biostatistics and Computational Biology, Dana Farber Cancer Institute, Harvard School of Public Health, Boston, MA 02115, USA

<sup>3</sup>Division of Genetics, Department of Medicine, Brigham and Women's Hospital and Harvard Medical School, Boston, MA 02115, USA

<sup>4</sup>Department of Pathology, Brigham and Women's Hospital and Harvard Medical School, Boston, MA 02115, USA

<sup>5</sup>Harvard/MIT Division of Health Sciences and Technology (HST), Harvard Medical School, Boston, MA 02115, USA

<sup>6</sup>Molecular Oncology Research Institute, Tufts Medical Center, Boston, MA 02111, USA

<sup>7</sup>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<sup>1,5</sup>, Eunjung Lee<sup>2,3,5</sup>, Yu-Tsueng Liu<sup>4</sup>, Doheon Lee<sup>3</sup> and Trey Ideker<sup>1,2,4,\*</sup>

<sup>1</sup> Bioinformatics Program, University of California San Diego, La Jolla, CA, USA, <sup>2</sup> Department of Bioengineering, University of California San Diego, La Jolla, CA, USA,

<sup>3</sup> Department of Bio and Brain Engineering, Korea Advanced Institute of Science and Technology, Daejeon, Korea and <sup>4</sup> Cancer Genetics Program, Moores Cancer Center, University of California San Diego, La Jolla, CA, USA

<sup>5</sup> 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

## Networks: A Short Primer

## Networks: A Short Primer

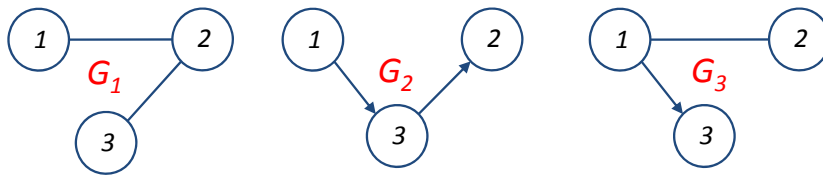
- ▶ A network is a collection of nodes  $V$  and edges  $E$ .

## Networks: A Short Primer

- ▶ A network is a collection of nodes  $V$  and edges  $E$ .
- ▶ We assume the network has  $p$  nodes, corresponding to random variables  $X_1, \dots, X_p \equiv$  biological measurements.

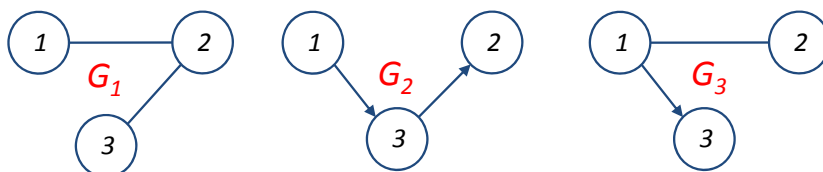
## Networks: A Short Primer

- ▶ A network is a collection of **nodes**  $V$  and **edges**  $E$ .
- ▶ We assume the network has  $p$  nodes, **corresponding to random variables**  $X_1, \dots, X_p \equiv$  biological measurements.
- ▶ Edges can be **directed**  $X \rightarrow Y$  or **undirected**  $X - Y$ .



## Networks: A Short Primer

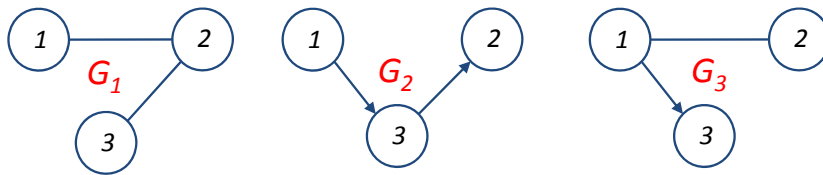
- ▶ A network is a collection of **nodes**  $V$  and **edges**  $E$ .
- ▶ We assume the network has  $p$  nodes, **corresponding to random variables**  $X_1, \dots, X_p \equiv$  biological measurements.
- ▶ Edges can be **directed**  $X \rightarrow Y$  or **undirected**  $X - Y$ .



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

## Networks: A Short Primer

- ▶ A network is a collection of **nodes**  $V$  and **edges**  $E$ .
- ▶ We assume the network has  $p$  nodes, **corresponding to random variables**  $X_1, \dots, X_p \equiv$  biological measurements.
- ▶ Edges can be **directed**  $X \rightarrow Y$  or **undirected**  $X - Y$ .



- ▶ In all these example, the **node set** is  $V = \{1, 2, 3\}$ .
- ▶ The **edges** are:

$$E_1 = \{1 - 2, 2 - 3\}$$

$$E_2 = \{1 \rightarrow 3, 3 \rightarrow 2\}$$

$$E_3 = \{1 - 2, 1 \rightarrow 3\}$$

## Networks: A Short Primer



## Networks: A Short Primer

- ▶ A convenient way to represent the **edges** of the network is to use an **adjacency matrix  $A$**

## Networks: A Short Primer

- ▶ 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)

## Networks: A Short Primer

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

$$A = \begin{bmatrix} \cdot & x & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} \text{ Here, } x \text{ is in row 1 and column 2}$$

## Networks: A Short Primer

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

$$A = \begin{bmatrix} \cdot & x & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} \text{ Here, } x \text{ is in row 1 and column 2}$$

- ▶ Adjacency matrix is a **square** matrix, which has a **1** if there is **an edge** from a **node in one row** to a **node in another column**, and **0** otherwise

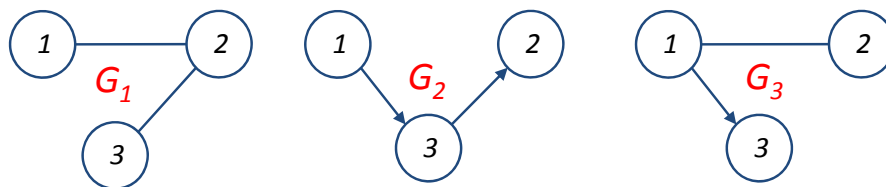
## Networks: A Short Primer

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

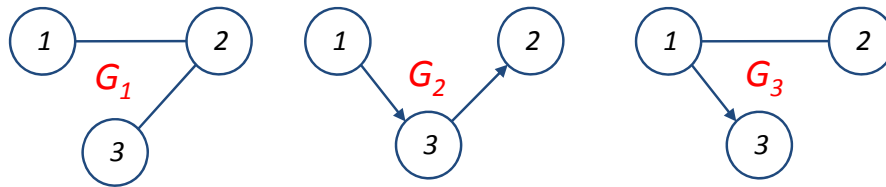
$$A = \begin{bmatrix} \cdot & x & \cdot \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \end{bmatrix} \text{ Here, } x \text{ is in row 1 and column 2}$$

- ▶ 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 Primer



## Networks: A Short Primer



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

## 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.

## 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.

## What Do Edges in Biological Networks Mean?

- ▶ In **gene regulatory networks**, an edge from gene  $i$  to gene  $j$  often means that  $i$  affects the expression of  $j$ ; i.e. as  $i$ 's expression changes, we expect that expression of  $j$  to increase/decrease.
- ▶ In **protein-protein interaction networks**, an edge between proteins  $i$  and  $j$  often means that *the two proteins bind together and form a protein complex*. Therefore, we expect that these proteins are generated at similar rates.
- ▶ In **metabolic networks**, an edge between compound  $i$  and  $j$  often means that *the two compounds are involved in the same reaction*, meaning that they are generated at relative rates.

## What Do Edges in Biological Networks Mean?

- ▶ In **gene regulatory networks**, an edge from gene  $i$  to gene  $j$  often means that  $i$  **affects the expression of  $j$** ; i.e. as  $i$ 's expression changes, we expect that expression of  $j$  to increase/decrease.
- ▶ In **protein-protein interaction networks**, an edge between proteins  $i$  and  $j$  often means that *the two proteins bind together and form a protein complex*. Therefore, we expect that these proteins are generated at similar rates.
- ▶ In **metabolic networks**, an edge between compound  $i$  and  $j$  often means that *the two compounds are involved in the same reaction*, meaning that they are generated at relative rates.
- ▶ Thus, edges represent some type of **association among genes, proteins or metabolites**, defined generally to include *linear or nonlinear* associations; more later....

## Statistical Models for Biological Networks



## Statistical Models for Biological Networks

- ▶ We use the framework of **graphical models**

## Statistical Models for Biological Networks

- ▶ We use the framework of **graphical models**
- ▶ In this setting, **nodes correspond to “random variables”**

## Statistical Models for Biological Networks

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

## Statistical Models for Biological Networks

- ▶ We use the framework of **graphical models**
- ▶ In this setting, **nodes correspond to “random variables”**
- ▶ In other words, each node of the network represents one of the variables in the study
  - ▶ In gene regulatory networks, **nodes  $\equiv$  genes**
  - ▶ In PPI networks, **nodes  $\equiv$  proteins**
  - ▶ In metabolic networks, **nodes  $\equiv$  metabolites**
- ▶ In practice, we observe  $n$  measurements of each of the variables (genes/proteins/ metabolites) for say different individuals, and want to determine which variables are connected, or use their connection for statistical analysis

## Our Plan

We will cover the following topics

- ▶ Methods for **detecting signal on known networks**
  - ▶ Network analysis based on **centrality and clustering**
  - ▶ **Topology-based pathway enrichment analysis**
- ▶ Methods for **learning undirected networks**
  - ▶ Co-expression networks
  - ▶ ARACNE
  - ▶ Conditional independence graphs
    - ▶ Gaussian observations (glasso, etc)
    - ▶ Non-Gaussian and non-linear data (nonparanormal, etc)
- ▶ Methods for **learning directed networks**
  - ▶ Bayesian Networks (basic concepts, reconstruction algorithm)
  - ▶ Learning directed networks from time-course data (dynamic Bayesian networks)

## Pathway & Network Analysis of Omics Data: Analysis of Network-Structured Data

Ali Shojaie  
Department of Biostatistics  
University of Washington  
[faculty.washington.edu/ashojaie](http://faculty.washington.edu/ashojaie)

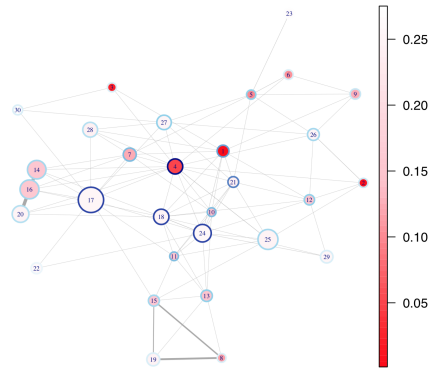
Summer Institute for Statistical Genetics – 2021

### Introduction

Suppose we observe **activities of individual nodes** (genes, proteins, brain regions, etc) on a **network** (gene regulatory network, structural connectivity network, etc)

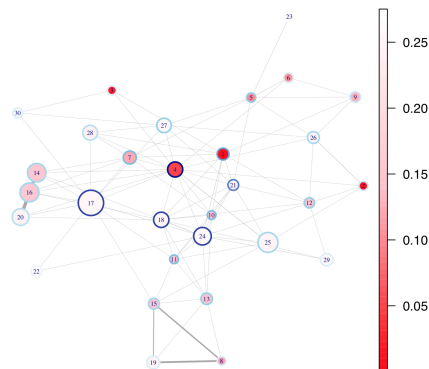
## Introduction

Suppose we observe **activities of individual nodes** (genes, proteins, brain regions, etc) on a **network** (gene regulatory network, structural connectivity network, etc)



## Introduction

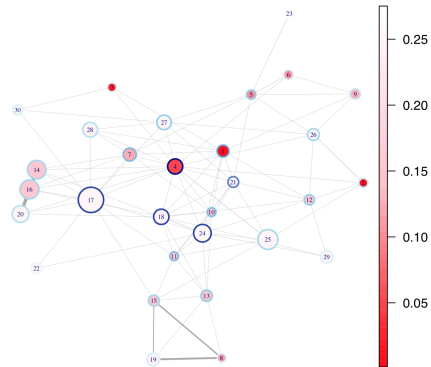
Suppose we observe **activities of individual nodes** (genes, proteins, brain regions, etc) on a **network** (gene regulatory network, structural connectivity network, etc)



How can we identify the **important nodes**?

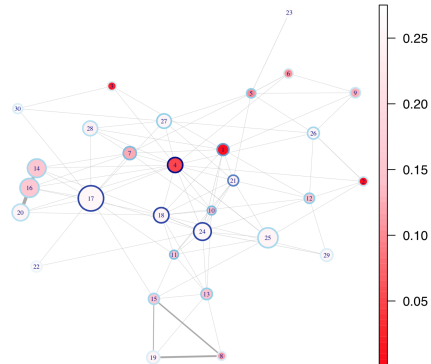
## Introduction

Suppose we observe **activities of individual nodes** (genes, proteins, brain regions, etc) on a **network** (gene regulatory network, structural connectivity network, etc)



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

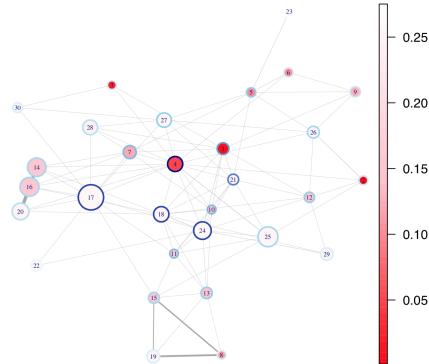
## Identifying Important Nodes



How can we identify the **important nodes**?



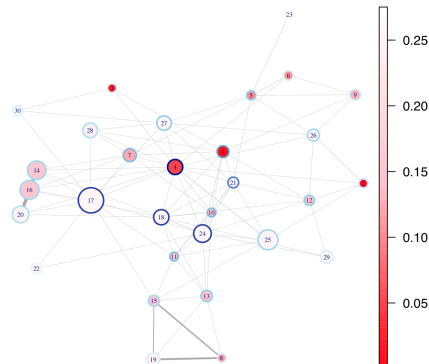
## Identifying Important Nodes



How can we identify the **important nodes**?

- ▶ We can select the **significant nodes** based on p-values, after adjusting for multiple comparisons (FDR, etc)

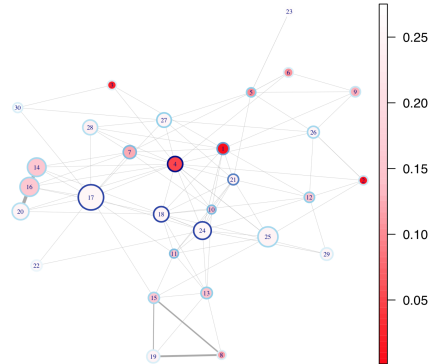
## Identifying Important Nodes



How can we identify the **important nodes**?

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

## Identifying Important Nodes



How can we identify the **important nodes**?

- ▶ We can select the **significant nodes** based on p-values, after adjusting for multiple comparisons (FDR, etc)
- ▶ But the signal is often weak for lots of tests
- ▶ If we believe the network is informative, it may make sense to **use the network to guide our selection**

## Identifying Important Nodes

Possible strategies:

- ▶ Identify **individual nodes** associated with the outcome by incorporating the network (signal detection on network)
- ▶ Test if (pre-specified) **subnetworks** are associated with the outcome (**topology-based pathway enrichment analysis**)
- ▶ Identify **collections of (connected) nodes** that are associated with the outcome (*de-novo identification of enriched modules*)

# Signal Detection on Networks

# Signal Detection on Networks

How can we identify the **important nodes in a network?**

## Signal Detection on Networks

How can we identify the **important nodes in a network?**

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

## Signal Detection on Networks

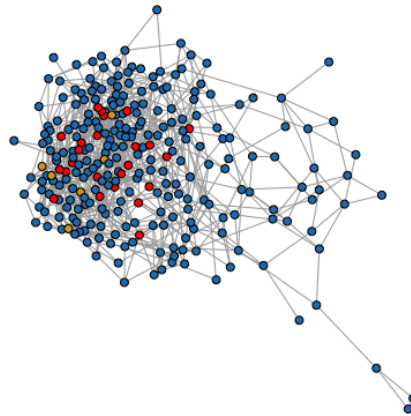
How can we identify the **important nodes in a network?**

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

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

## Example: Functional Relevance of Hub Nodes

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



## Other Measures of Centrality

- ▶ **Closeness:** Total distance of each node to other nodes:

$$cl_j = \left( \sum_{k \in V} d(j, k) \right)^{-1}$$

where  $d(j, k)$  is the (shortest path) distance between  $j$  and  $k$ .

- ▶ **Betweenness:** The number of *paths* that go through a node:

$$bw_j = \sum_{i \neq j \neq k} \frac{\pi_{ik}(j)}{\pi_{ik}}$$

where  $\pi_{ik}(j)$  is the number of paths between  $i$  and  $k$  that go through  $j$ , and  $\pi_{ik}$  is the total number of paths between them.

## Identifying “Central” Nodes

Calculating centrality measures using igraph:

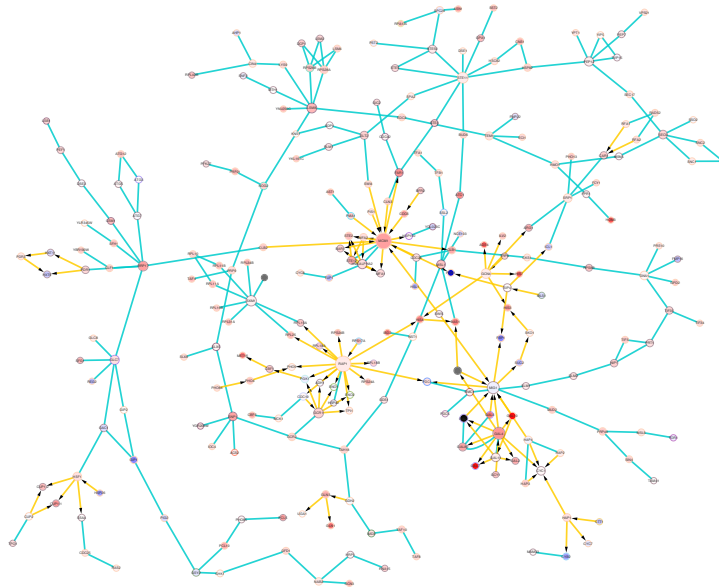
- ▶ Hub nodes: `hub_score(graph)`
- ▶ Closeness: `closeness(graph, vids)`
  - ▶ use `estimate_closeness()` for larger networks)
- ▶ Betweenness: `betweenness(graph, vids)`
  - ▶ use `estimate_betweenness()` for larger networks)

# Topology-Based Pathway Enrichment Analysis



## Yeast GAL Pathway

Ideker et al, 2001



## Topology-Based Pathway Enrichment Analysis

Test for **changes in activities of node** (genes, brain ROIs, etc) in **pre-specified subnetworks**, while **incorporating network information**

Two possible null hypotheses:

## Topology-Based Pathway Enrichment Analysis

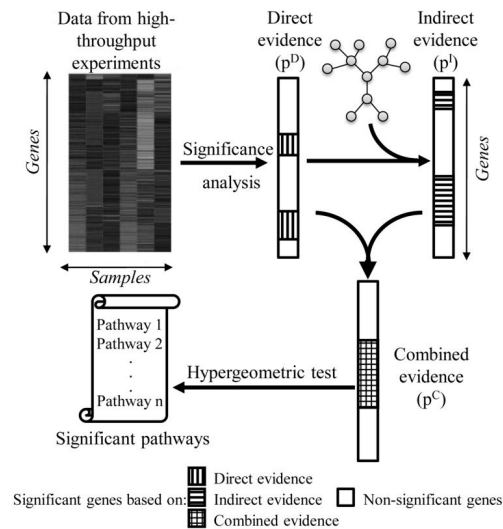
Test for **changes in activities of node** (genes, brain ROIs, etc) in **pre-specified subnetworks**, while **incorporating network information**

Two possible null hypotheses:

- ▶ **Competitive** null hypothesis: activity of each pathway is **compared with other pathways**, often using a **permutation test**
  - ▶ Assume few genes are differentially connected, and may be sensitive to the choice of gene sets
- ▶ **Self-contained** null hypothesis: activity of each pathway is compared **against the null** distribution
  - ▶ More rigorous, but may be sensitive to modeling assumptions (*Goemen & Buhlmann (07)*, *Ackermann & Strimmer (09)*)

## PathNet<sup>1</sup>

A simple topology-based pathway enrichment method:



<sup>1</sup>Dutta et al (2012)

## PathNet: Details

- ▶ Each gene's  $p$ -value from differential expression is combined with  $p$ -values of its neighbors using **Fisher's methods**

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

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

## topologyGSA<sup>2</sup>

- ▶ topologyGSA (Gene Set Analysis Exploiting Pathway Topology) assumes that data are normally distributed:

$$X^1 \sim N(\mu^1, \Sigma^1), \quad X^2 \sim N(\mu^2, \Sigma^2)$$

- ▶ It obtains estimates of  $\Sigma^1$  and  $\Sigma^2$  based on the networks (think graphical lasso, but with **known nonzero entries**)
- ▶ It then **performs two tests**:
  - ▶ equality of covariance matrices:  $H_0^c : \Sigma^1 = \Sigma^2$
  - ▶ equality of means  $H_0^m : \mu^1 = \mu^2$  — it uses different methods depending on the result of  $H_0^c$
- ▶ Implemented in R-package topologyGSA (also in graphite)

<sup>2</sup>Massa et al (2010)

## Signaling Pathway Impact Analysis (SPIA)<sup>3</sup>

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

---

<sup>3</sup>Tarca et al (2009)

## The SPIA Methodology

## The SPIA Methodology

SPIA combines two types of evidence

## The SPIA Methodology

SPIA combines two types of evidence

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

## The SPIA Methodology

SPIA combines two types of evidence

- (i) the **overrepresentation** of DE genes in a given pathway
  - ▶ measured by the p-value for the given number of DE genes

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

## The SPIA Methodology

SPIA combines two types of evidence

## The SPIA Methodology

SPIA combines two types of evidence

- (ii) the **abnormal perturbation of the pathway**

## The SPIA Methodology

SPIA combines two types of evidence

- (ii) the **abnormal perturbation of the pathway**
  - ▶ the **perturbation for each gene** in the pathway is defined as

$$PF(g_i) = \Delta E(g_i) + \sum_{j=1}^p \beta_{ij} \frac{PF(g_j)}{N_{DS}(g_j)}$$

## The SPIA Methodology

SPIA combines two types of evidence

(ii) the **abnormal perturbation of the pathway**

▶ the **perturbation for each gene** in the pathway is defined as

$$PF(g_i) = \Delta E(g_i) + \sum_{j=1}^p \beta_{ij} \frac{PF(g_j)}{N_{DS}(g_j)}$$

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

## The SPIA Methodology



## The SPIA Methodology

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

## The SPIA Methodology

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

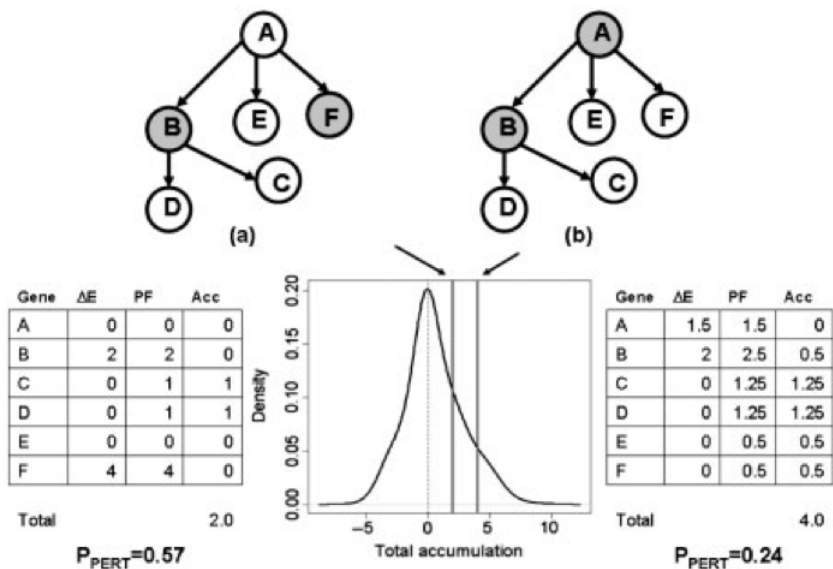
## The SPIA Methodology

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

## The SPIA Methodology

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

## The SPIA Methodology



## The SPIA Methodology

## The SPIA Methodology

SPIA combines two types of evidence

## The SPIA Methodology

SPIA combines two types of evidence

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

## The SPIA Methodology

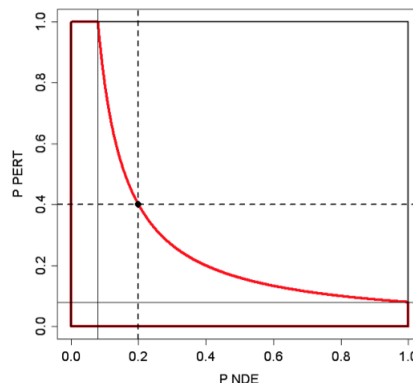
SPIA combines two types of evidence

- ▶ The **final p-value for each pathway** is calculated based on the p-values from parts (i) and (ii):
  - ▶  $P_G(i) = c_i - c_i \ln(c_i)$
  - ▶  $c_i = P_{NDE}(i)P_{PERT}(i)$

## 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)

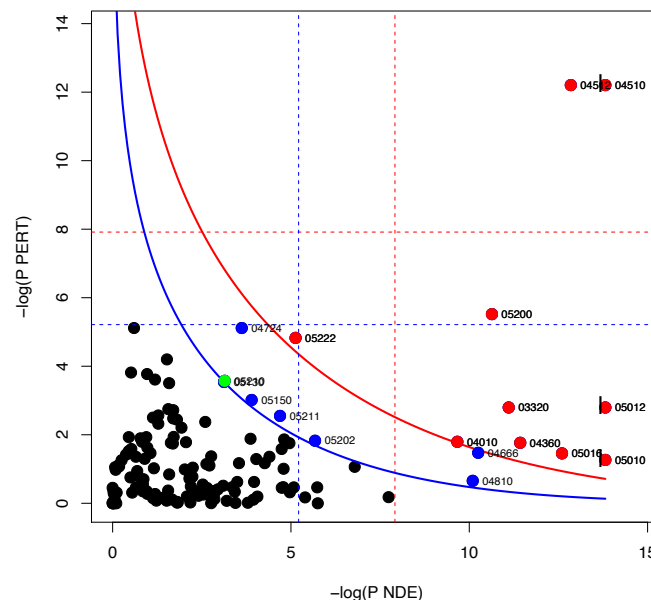
#pathway analysis using SPIA
#use nB=2000 or higher for more accurate results
#uses older version of KEGG signaling pathways graphs
res <- spia(de=DE_Colorectal, all=ALL_Colorectal, organism="hsa", beta=NULL,
  nB=2000, plots=FALSE, verbose=TRUE, combine="fisher")

#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

SPIA two-way evidence plot



## Network-Based Gene Set Analysis (NetGSA)<sup>4</sup>

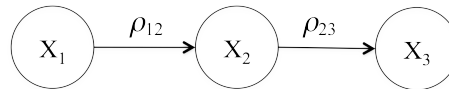
- ▶ Generalizes SPIA, to allow for more complex experiments & incorporate interactions among pathways
- ▶ Assesses the overall behavior of arbitrary subnetworks (pathways): **changes in gene expression & network structure**
- ▶ Uses **latent variables** to model the interaction between genes defined by the network
- ▶ Uses **mixed linear models** for inference in complex data
- ▶ Computationally challenging for large networks, unless pathways separately analyzed (similar to SPIA)

<sup>4</sup>S & Michailidis (2009, 2010); Ma, S & Michailidis (2016)

## Problem Setup

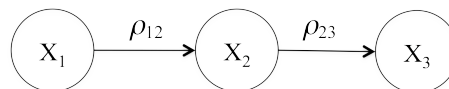
- ▶ Gene (protein/metabolite) expression data for  **$K$  experimental conditions** and  **$J_k$  time points**
- ▶ Network information (partially) available in the form of a **directed weighted graph  $G = (V, E)$** , with vertex set  $V$  corresponding to the genes/proteins/metabolites and edge set  $E$  capturing their associations
- ▶ Network edges can be **directed  $j \rightarrow k$**  or **undirected  $j \leftrightarrow k$**
- ▶ Edges 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  $k \rightarrow j$  & for undirected edges,  $A_{jk} = A_{kj}$**

## The Latent Variable Model: Main Idea



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

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

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

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  $Y$ , we can write

$$Y = \Psi\beta + \Pi\gamma + \varepsilon$$

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

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

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

## Mixed Linear Model Representation

Rearranging the expression matrix into  $np$ -vector  $Y$ , we can write

$$Y = \Psi\beta + \Pi\gamma + \varepsilon$$

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

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

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

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

## Estimation of MLM Parameters

MLE for  $\beta$ :

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

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

$\hat{\beta}$  depends on estimates of  $\sigma_\gamma^2$  and  $\theta_\varepsilon^2$  (estimated using **restricted maximum likelihood (REML)**).

## Inference using MLM

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

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

## Inference using MLM

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

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

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

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

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

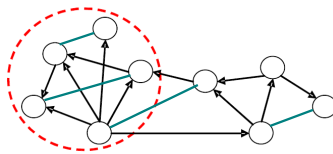
- ▶ Under the null hypothesis,  $T$  is approximately  $t$ -distributed with degrees of freedom that needs to be estimated

## “Optimal” Choice of Contrast Vector

- ▶ An intuitive choice is the **indicator (membership) vector** for the pathway,  $\mathbf{b}$ , but this only captures changes in mean
- ▶ Need to *de-couple the effect of subnetwork* from other nodes

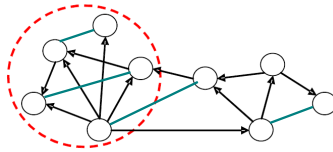
## “Optimal” Choice of Contrast Vector

- ▶ An intuitive choice is the **indicator (membership) vector** for the pathway,  $\mathbf{b}$ , but this only captures changes in mean
- ▶ Need to *de-couple the effect of subnetwork* from other nodes

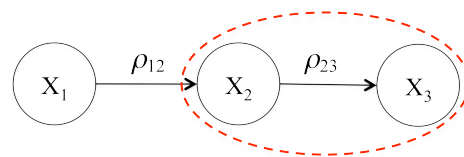


## “Optimal” Choice of Contrast Vector

- ▶ An intuitive choice is the **indicator (membership) vector** for the pathway,  $\mathbf{b}$ , but this only captures changes in mean
- ▶ Need to **de-couple the effect of subnetwork** from other nodes



## “Optimal” Choice of Contrast Vector



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

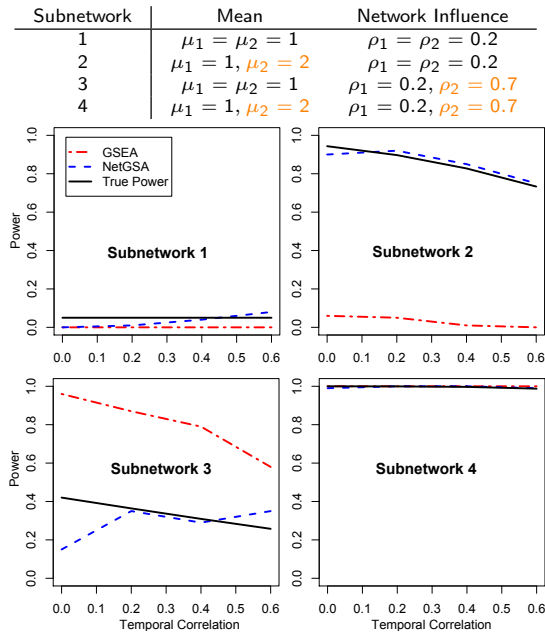
Consider the set,  $\mathbf{b} = (0, 1, 1)$ ; then

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

On the other hand,

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

## Comparison in Simulated Data



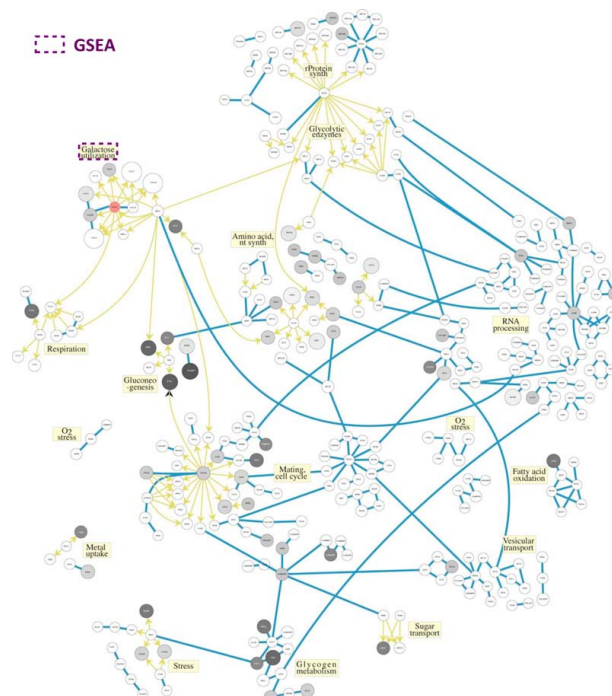
## Yeast Galactose Utilization Pathway

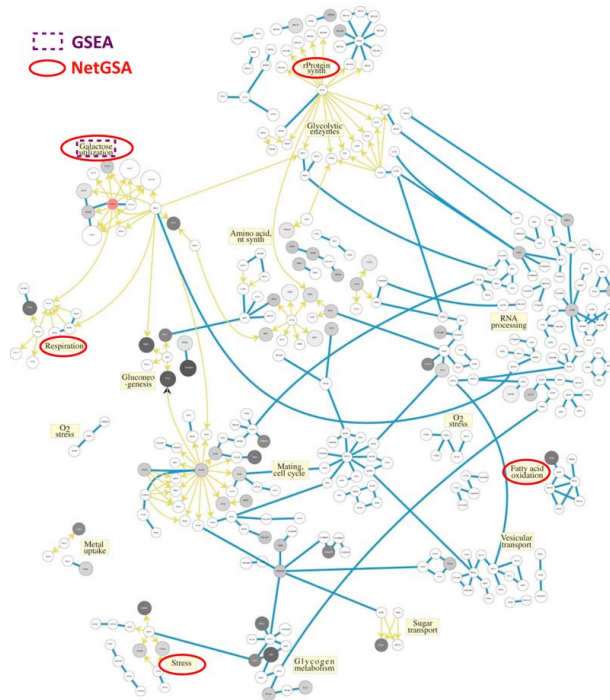
Ideker et al (2001) data on yeast Galactose Utilization Pathway

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

## Analysis of Yeast GAL Data

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





## Environmental Stress Response in Yeast

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

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



## Yeast ESR Data

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

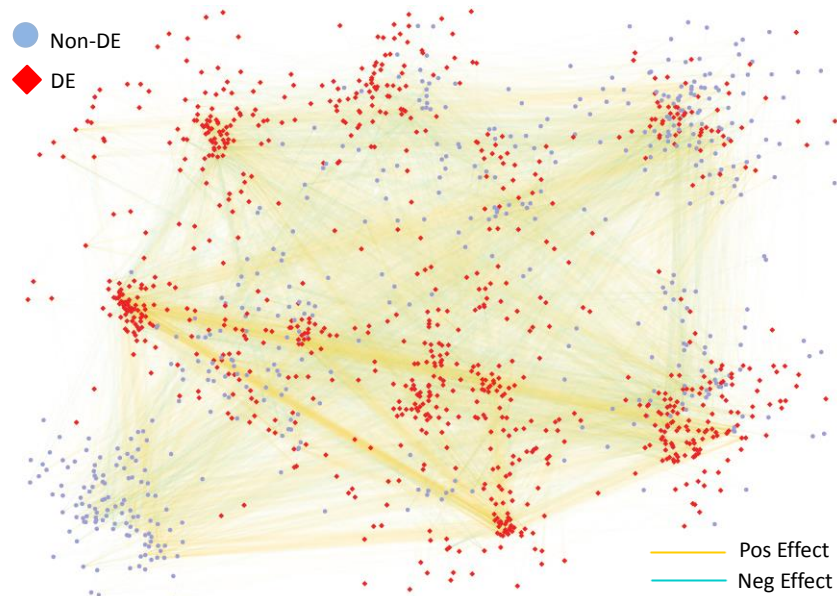
## Model and Results

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

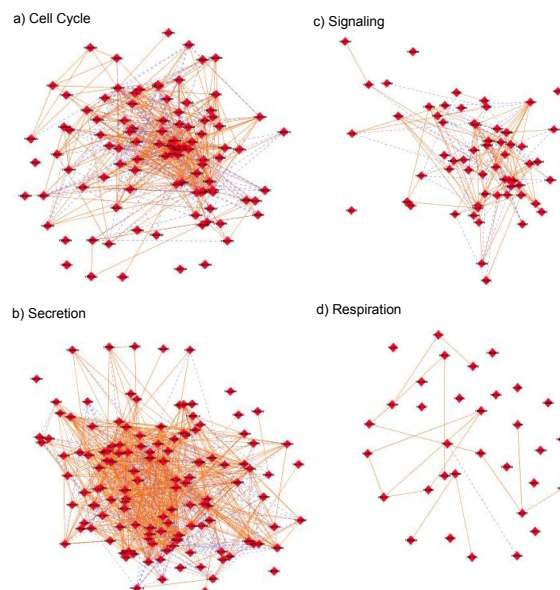
$$\mathbb{E}Y_{11} = \Lambda\mu, \quad \mathbb{E}Y_{jk} = \Lambda(\mu + \alpha_j + \delta_k) \quad j, k = 2, 3$$

- **Temporal correlation** is modeled directly via  $R$  (as  $AR(1)$  process)
- Results:
  - $\sim 3000$  genes,
  - 47 pathways showed significant changes of expression
  - 24 pathways showed changes over **time**
  - 29 pathways showed changes in response to different **sorbitol** levels
  - 12 pathways showed **both** types of changes
  - Significant pathways overlap with the gene functions recognized by Gasch et al.

## Yeast ESR Network

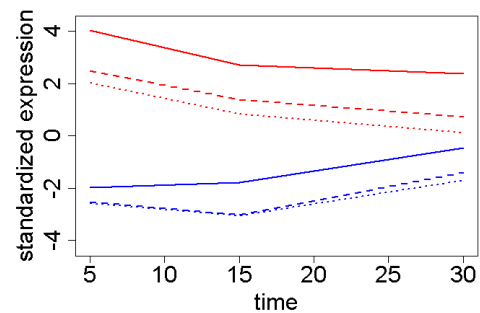


## Significant subnetworks



## Expression Profiles

Average Standardized Expression Levels of Pathways



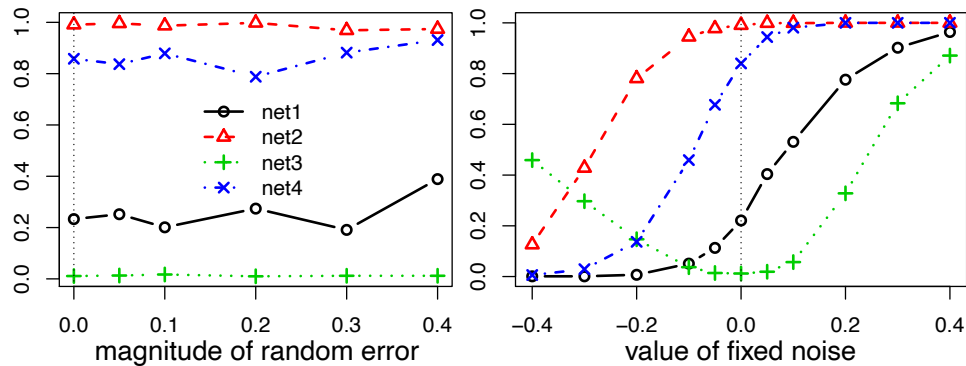
- ▶ **Induced** and **Suppressed** Pathways
- ▶ Can observe the **transient patterns of expressions** as predicted by *Gasch et al.*

## 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)

## Effect of Noise In Network Information

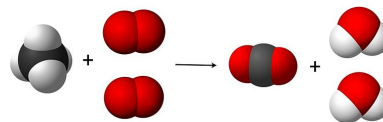
- ▶ Let  $\tilde{A}$  be observed network information, and  $A$  be the truth.
- ▶ It can be shown that, if  $\|\tilde{A} - A\|$  is small then, NetGSA **still works** (is asymptotically most powerful unbiased test)



## Metabolic Profiling in Bladder Cancer

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

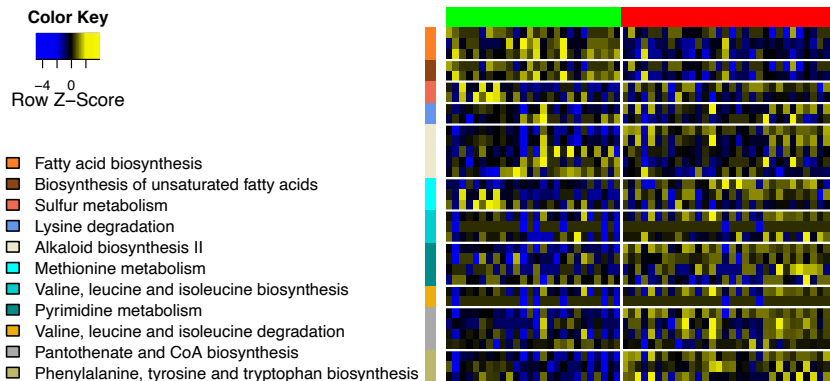
- ▶ 58 bladder cancer and adjacent benign samples
- ▶ Pathways information obtained from **KEGG**



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

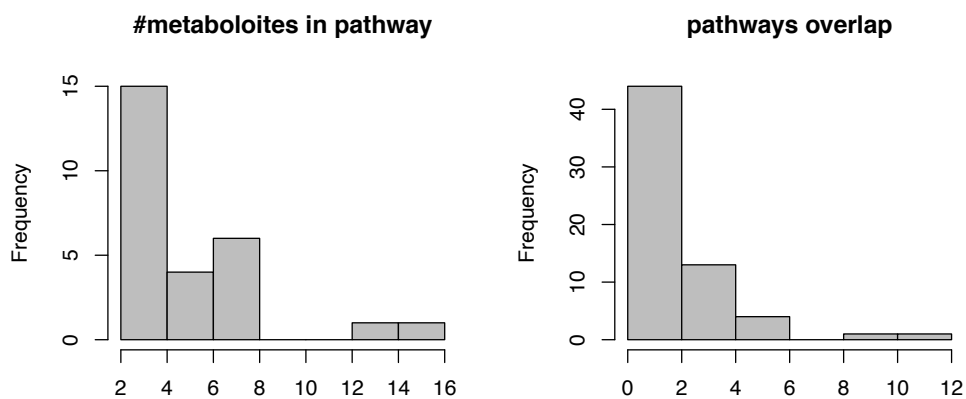
## Metabolic Profiling in BCa

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



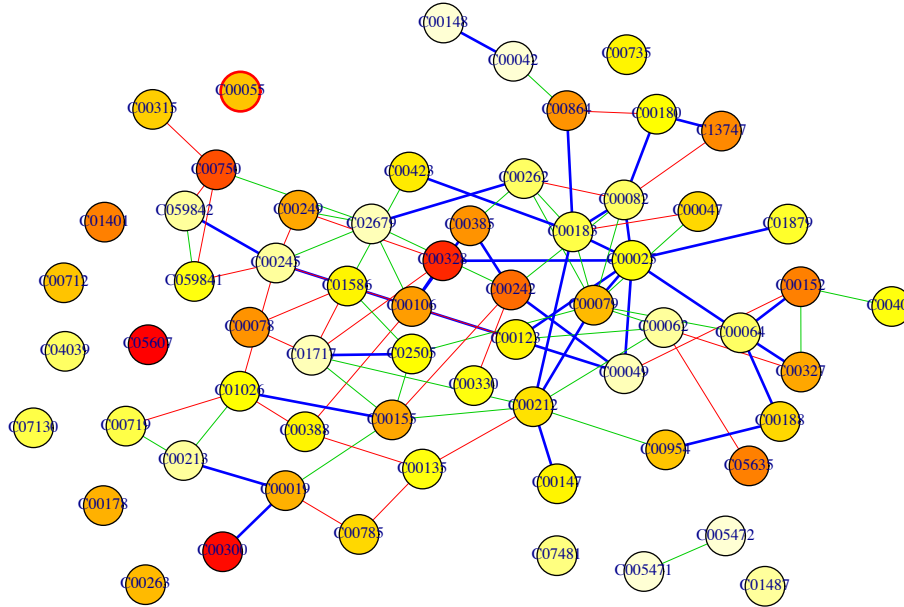
## Metabolic Profiling in BCa

- ▶ 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, similar to *Putluri et al* (2012)

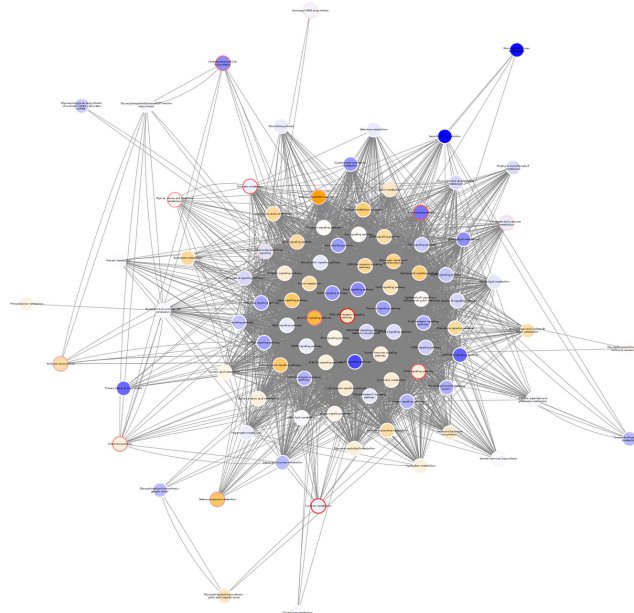
## R-Package netgsa

► Basic usage:

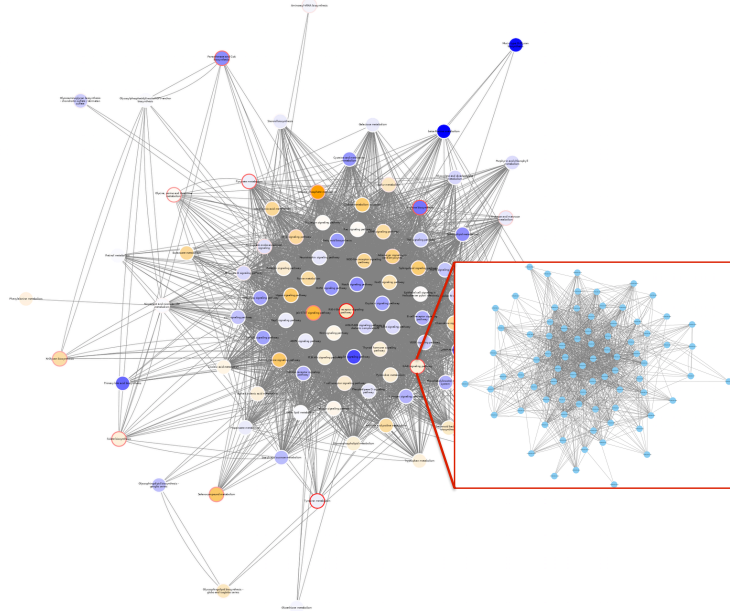
`NetGSA(A, x, group, pathways)`

- **A**: List of  $p \times p$  **weighted adjacency matrices** for each condition (e.g. normal vs cancer), to **capture changes in the network**
- **pathways**: a  $K \times p$  0-1 matrix of pathway membership:  $\text{pathways}_{kj} = 1$  if gene/.../metabolite  $j$  in pathway  $k$
- **Output**: test statistics and p-values for each pathway
- The NetGSA function takes a weighted **A** as input. The package includes functions to **learn A for undirected networks** from a (partial) list of network edges

## R-Package netgsa



## R-Package netgsa

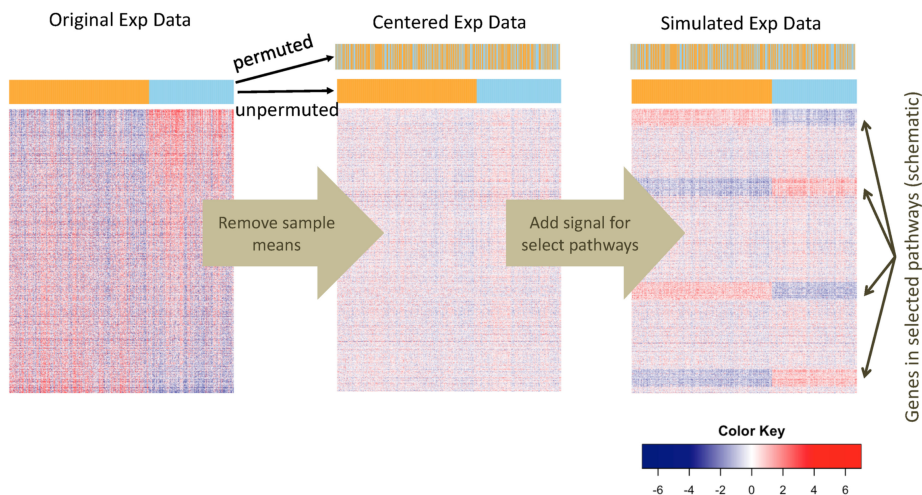


## Comparison Using Synthetic Data (Ma, S., Michailidis, 2019)

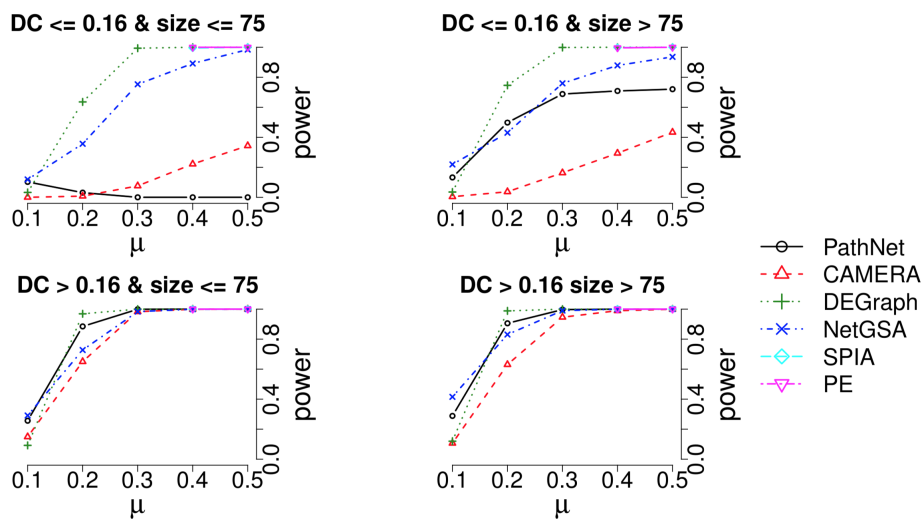
- ▶ Comparison of topology-based pathway enrichment methods using two synthetic data sets
  - ▶ Gene expression data  $p \approx 3000$
  - ▶ Metabolomics data  $p \approx 100$
- ▶ *In silico* data sets with known signal:
  1. Remove the original signal, but **keep the correlation structure**
  2. **Perturb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



## Comparison Using Synthetic Data



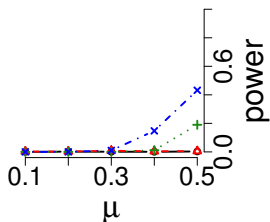
## Results for Gene Expression Data — Equal Covariance



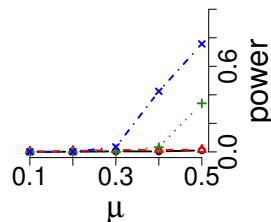


## Results for Metabolomics Data — Equal Covariance

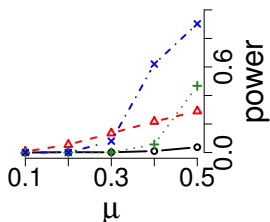
DC  $\leq 0.5$  & size  $\leq 11$



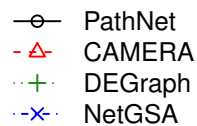
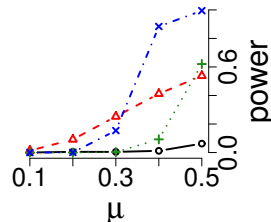
DC  $\leq 0.5$  & size  $> 11$



DC  $> 0.5$  & size  $\leq 11$

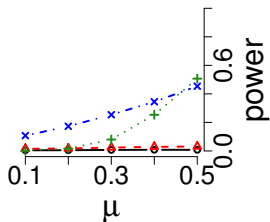


DC  $> 0.5$  size  $> 11$

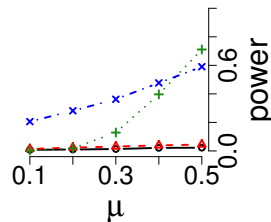


## Results for Metabolomics Data — Diff Covariance

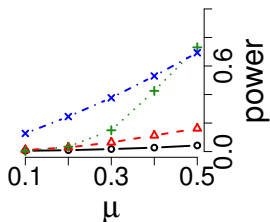
DC  $\leq 0.5$  & size  $\leq 11$



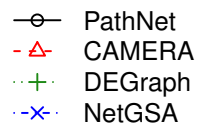
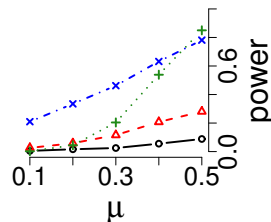
DC  $\leq 0.5$  & size  $> 11$



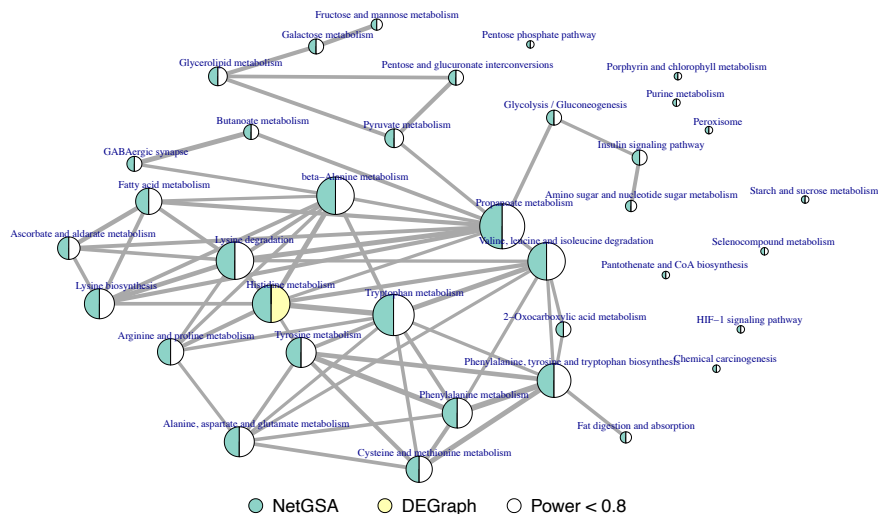
DC  $> 0.5$  & size  $\leq 11$



DC  $> 0.5$  size  $> 11$



## Results for Metabolomics Data



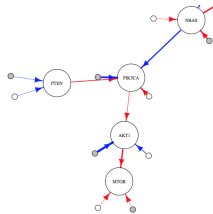
## Multi-Omics NetGSA

Pan-cancer integration of **expression**, **methylation** and **CNV** in **BRAF** (TCGA data)<sup>5</sup>

<sup>5</sup>Zhang et al (2018)

## Multi-Omics NetGSA

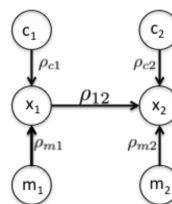
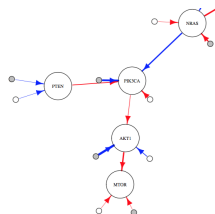
Pan-cancer integration of **expression**, **methylation** and **CNV** in **BRAF** (TCGA data)<sup>5</sup>



<sup>5</sup>Zhang et al (2018)

## Multi-Omics NetGSA

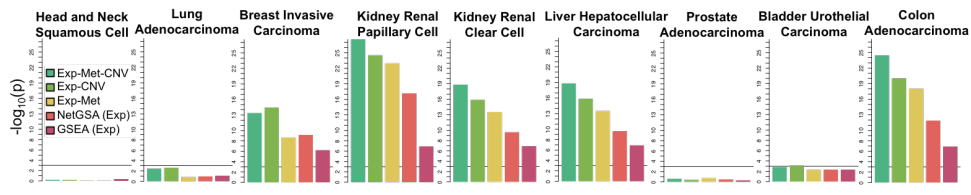
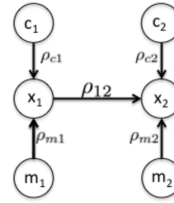
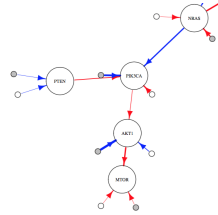
Pan-cancer integration of **expression**, **methylation** and **CNV** in **BRAF** (TCGA data)<sup>5</sup>



<sup>5</sup>Zhang et al (2018)

## Multi-Omics NetGSA

Pan-cancer integration of expression, methylation and CNV in BRAF (TCGA data)<sup>5</sup>



<sup>5</sup>Zhang et al (2018)

## Identifying Enriched Modules in Networks

## Identifying Enriched Modules in Networks

Two general strategies:

- ▶ Assess the significance of **data-driven modules** (WGCNA):
  1. Identify modules (network clustering, etc)
  2. Assess the significance of modules
- ▶ **Search** for enriched (connected) subnetworks (often using greedy search methods)

## Identifying Enriched Modules in Networks

Two general strategies:

- ▶ Assess the significance of **data-driven modules** (WGCNA):
  1. Identify modules (network clustering, etc)
  2. Assess the significance of modules
- ▶ **Search** for enriched (connected) subnetworks (often using greedy search methods)
- ▶ Advantage: No need to rely on known pathways — especially useful when known pathways are not complete, etc
- ▶ Disadvantage: Interpretation may become challenging...

## WGCNA<sup>6</sup>

- ▶ WGCNA is a method for constructing weighted gene co-expression networks (discussed in the next lecture), which also facilitates topology-based enrichment analysis, in a different way than many other topology-based methods

---

<sup>6</sup>Horvath & Zhang (2005); Langfelder et al (2008)

## WGCNA<sup>6</sup>

- ▶ WGCNA is a method for constructing weighted gene co-expression networks (discussed in the next lecture), which also facilitates topology-based enrichment analysis, in a different way than many other topology-based methods
- ▶ Here's how it works:
  1. Estimate the **co-expression network** (more in the next lecture)
  2. Find **modules** by **clustering** the nodes in the estimated network
  3. Summarize the expressions of genes in each module using PCA (**eigen-genes**)
  4. Test if the eigen-genes are associated with the outcome

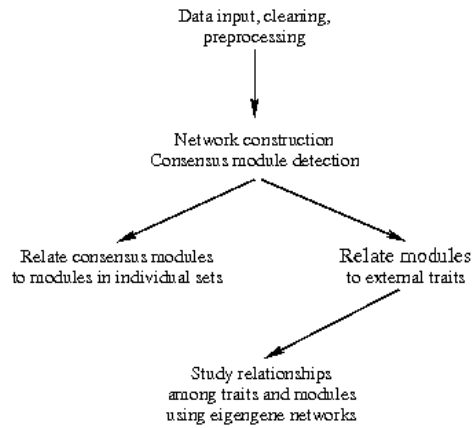
---

<sup>6</sup>Horvath & Zhang (2005); Langfelder et al (2008)



## WGCNA

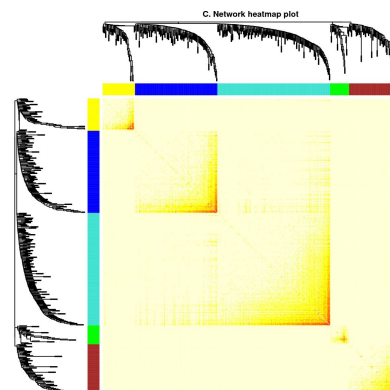
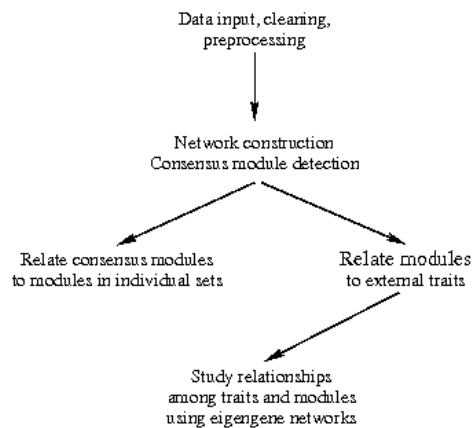
- Here's how it works:



Let's look at an example in R...

## WGCNA

- Here's how it works:



Let's look at an example in R...

## Walktrap<sup>7</sup>

- ▶ Searches for connected modules containing significant genes
  - ▶ Weights each edges based on the **significance of its corresponding nodes**

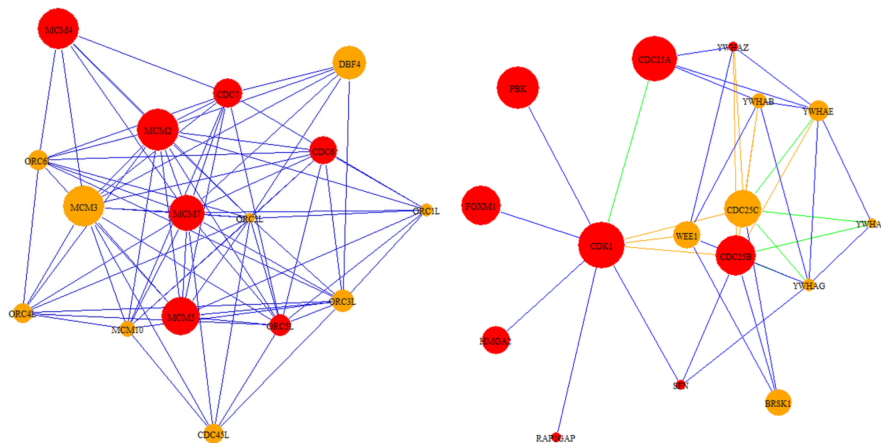
$$w_{ij} = (|FC_i| + |FC_j|)/2$$

- ▶ Connected significant modules are found through **community detection** using a **random walk** with transition probability

$$P_{ij} = \frac{w_{ij}}{\sum_j w_{ij}}$$

<sup>7</sup>Petrochilos et al (2013)

## Identifying Cancer-Related Modules

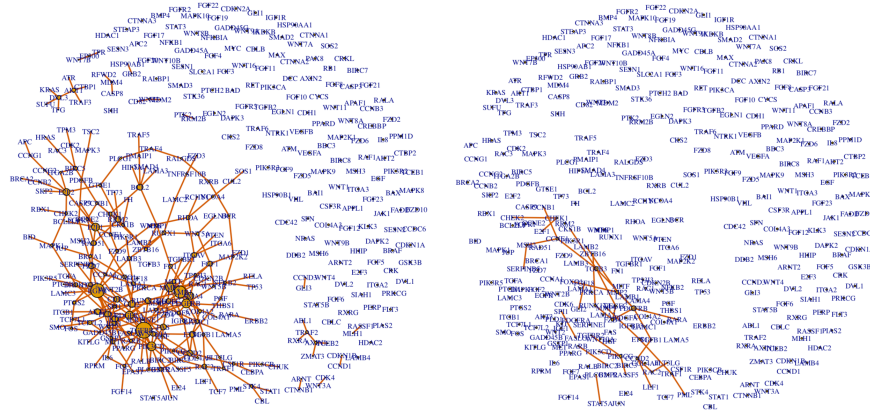


## Summary

- ▶ Network-based methods (centrality-based, pathway topology, etc) rely on network information — helpful if correct network information avail
- ▶ What if network information is not available?

## Summary

- ▶ Focus is shifting towards estimating changes in the structure of networks: **differential network biology**<sup>8</sup>



<sup>8</sup>Ideker & Krogan (2012); S (2020)

## Pathway & Network Analysis of Omics Data: Learning Undirected Networks

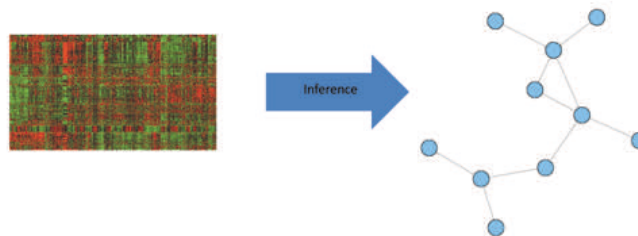
Ali Shojaie  
Department of Biostatistics  
University of Washington  
faculty.washington.edu/ashojaie

Summer Institute for Statistical Genetics – 2021

## Learning Undirected Networks

Learn network from data (**structure learning**):

- ▶ Data matrix:  $X_{n \times p}$ .
- ▶ Features correspond to the  $p$  nodes in the network.
- ▶ Goal: Learn edges between nodes  $\equiv$  learn the **statistical relationships** between features.



## Why Do We Need Network Inference?

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

## Network Inference — An Overview

## Network Inference — An Overview

Two general classes of network inference methods:

- ▶ Methods based on **marginal measures of association**:
  - ▶ Co-expression Networks (based on 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 (g1asso, etc)
  - ▶ Generalizations to allow for nonlinear dependencies (nonparanormal, etc)

## Graphical Models

### Probabilistic Graphical Models <sup>1</sup>

Joint multivariate probability distribution where dependencies can be represented as a network.

Advantages:

- ▶ Graphical models offer efficient factorized forms for joint distributions with easily interpretable dependencies.
  - ▶ **Conditional dependencies** denoted via an edge in network.
- ▶ Convenient visual representation.

---

<sup>1</sup>A detailed technical introduction is provided in *Graphical Models, Exponential Families, and Variational Inference*; Wainwright & Jordan (2008)

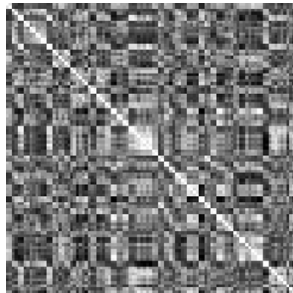
# Marginal Association Networks

## Correlation Networks (Association Networks)

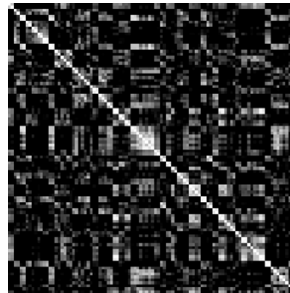
- ▶ Simplest (and most-widely used!) method for estimating networks — key assumption:  
large correlation  $\equiv$  presence of an edge
- ▶ Let  $r(i, j)$  be correlation between  $X_i$  and  $X_j$ ; we claim an **edge between  $i$  and  $j$  if  $|r(i, j)| > \tau$** .
  - ▶  $\tau$ : a user-specified threshold (**tuning parameter**).

## Correlation Networks (Association Networks)

- ▶ Simplest (and most-widely used!) method for estimating networks — key assumption:  
large correlation  $\equiv$  presence of an edge
- ▶ Let  $r(i, j)$  be correlation between  $X_i$  and  $X_j$ ; we claim an **edge between  $i$  and  $j$  if  $|r(i, j)| > \tau$** .
  - ▶  $\tau$ : a user-specified threshold (**tuning parameter**).



Correlation matrix



Thresholded correlation matrix

## Limitations of Correlation Networks

1. The estimation is highly dependent on the **choice of  $\tau$** .
2. Correlations capture **linear associations**, but **many real-world relationships are nonlinear**.
3. Large correlations can occur due to **confounding**.



## Limitations of Correlation Networks

The estimation is highly dependent on the choice of  $\tau$ .

## Limitations of Correlation Networks

The estimation is highly dependent on the choice of  $\tau$ .

- ▶ We can work with **weighted co-expression networks** (WGCNA)
- ▶ We can instead **test  $H_0 : r_{xy} = 0$** 
  - ▶ A commonly used test is based on the **Fisher transformation**

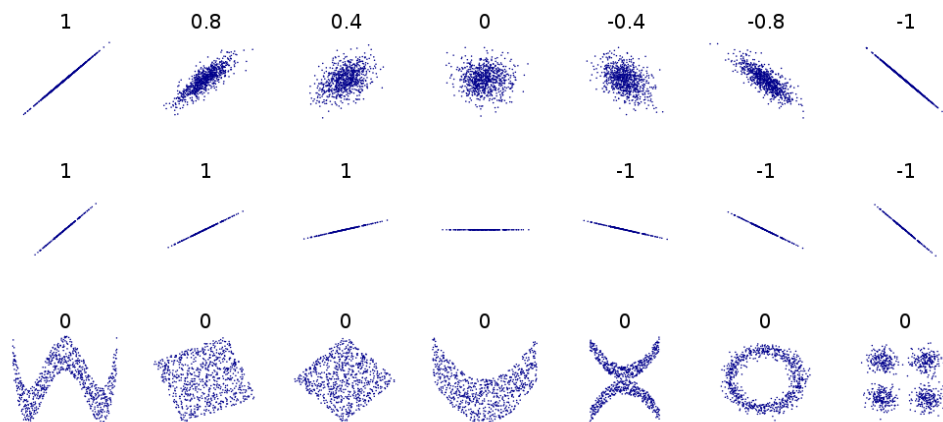
$$Z = \frac{1}{2} \ln \left( \frac{1+r}{1-r} \right) = \text{artanh}(r) \sim_{H_0} N \left( 0, \frac{1}{\sqrt{n-3}} \right)$$

## Limitations of Correlation Networks

Correlations capture **linear** associations, but many real-world relationships are nonlinear.

## Limitations of Correlation Networks

Correlations capture **linear** associations, but many real-world relationships are nonlinear.



## Limitations of Correlation Networks

Correlations capture **linear** associations, but many real-world relationships are nonlinear.

## Limitations of Correlation Networks

Correlations capture **linear** associations, but many real-world relationships are nonlinear.

- ▶ We can use other measures of association, for instance, **Spearman correlation** or **Kendal's  $\tau$** .
  - ▶ These methods define the 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 algorithms, e.g. ARACNE.

## ARACNE: Algorithm for the Reconstruction of Accurate Cellular NEtworks<sup>2</sup>

---

<sup>2</sup>Margolin et al (2006)

## ARACNE: Algorithm for the Reconstruction of Accurate Cellular NEtworks<sup>2</sup>

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

---

<sup>2</sup>Margolin et al (2006)

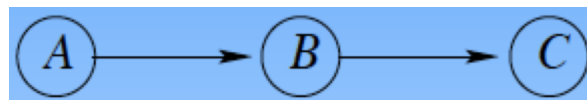
## ARACNE: Algorithm for the Reconstruction of Accurate Cellular NEtworks<sup>2</sup>

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

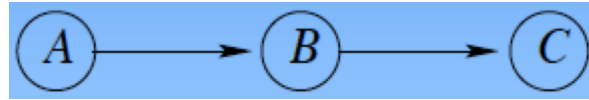
---

<sup>2</sup>Margolin et al (2006)

## Key Idea: Data Processing Inequality (DPI)



## Key Idea: Data Processing Inequality (DPI)

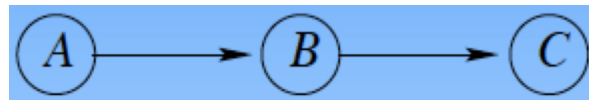


$$I(A, C) \leq \min[I(A, B), I(B, C)]$$

where

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

## Key Idea: 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

- ▶ The algorithm 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

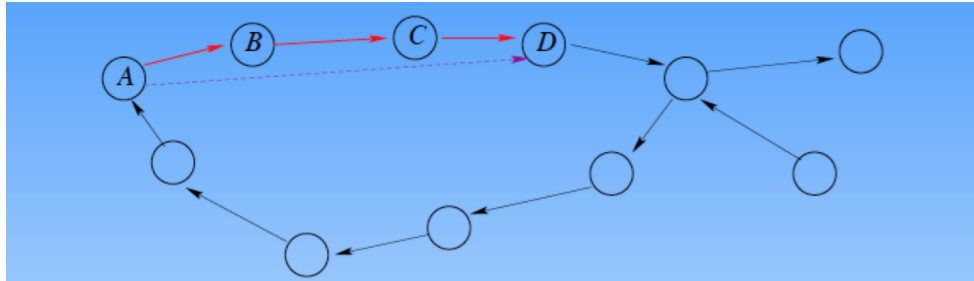
- ▶ The algorithm examines each gene triplet for which all pairwise MIs are greater than a cut-off and removes the edge with the smallest value based on DPI.
  - ▶ Each triplet is analyzed even if its edges have been selected for removal by prior DPI applications to other triplets.
  - ▶ The least of 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 are estimated with no errors**, then ARACNE reconstructs the underlying interaction network exactly, **if the network is a tree** and has only pairwise interactions.
- ▶ The maximum MI spanning tree is a subnetwork of the network built by ARACNE.



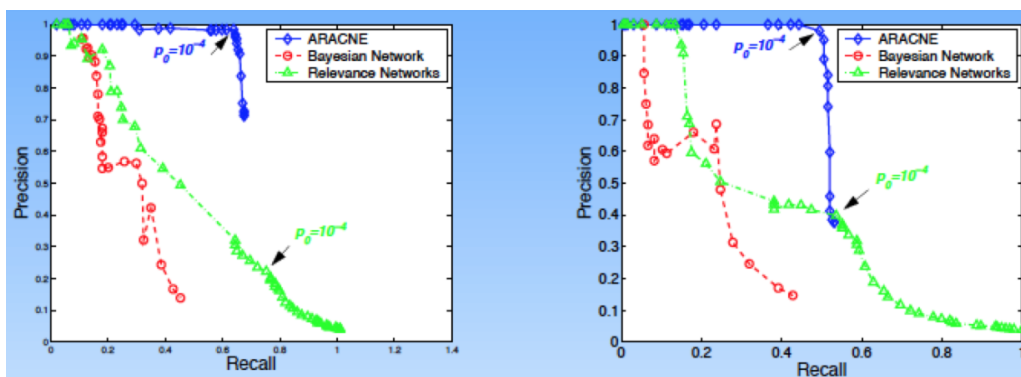
## Rationale and Guarantees



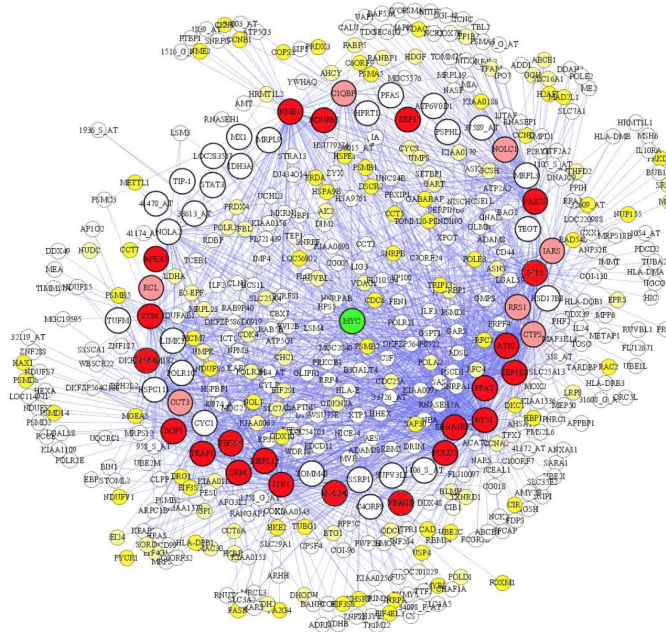
**Theorem.** Let  $\pi_{ik}$  be the set of nodes forming the shortest path in the network between nodes  $i$  and  $k$ . Then, if MIs can be estimated without errors, ARACNE reconstructs an interaction network without false positives edges, provided: (a) the network consists only of pairwise interactions,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

Further, ARACNE does not produce any false negatives, and the network reconstruction is exact iff (c) for each directly connected pair  $ij$  and for any other node  $k$ , we have  $I_{ij} > \min[I_{ik}, I_{jk}]$ .

## Performance on Synthetic Data



## Application: B-lymphocytes Expression Data



## Application: B-lymphocytes Expression Data

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

## Software

- ▶ Implemented in the R-package `minet`:

```
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")
```

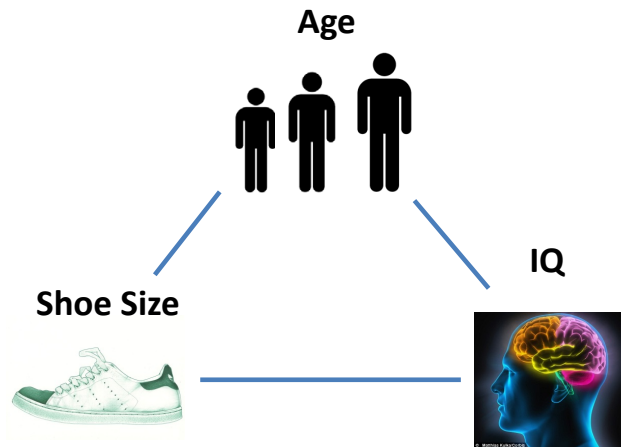
- ▶ `eps`: threshold for setting an edge to zero, prior to searching over triplets

## Limitations of Correlation Networks

Large correlations can occur due to **confounding**.

## Limitations of Correlation Networks

Large correlations can occur due to **confounding**.

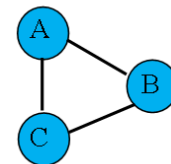


## Markov Networks

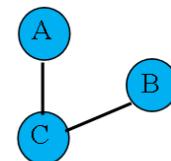
Markov Network

An *undirected graphical model* that characterizes **conditional dependence** ( $\equiv$  direct relationships).

- ▶ *Edge*: Two nodes are **conditionally dependent**.
- ▶ *No edge*: Two nodes are **conditionally independent**.
- ▶ Conditions on all other nodes.



$$A \perp B \mid C$$

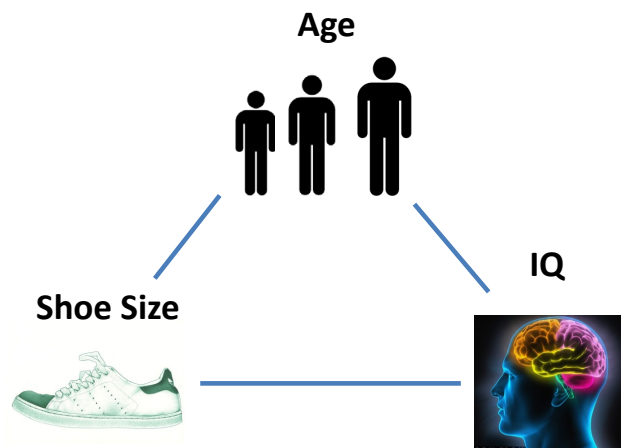


## Markov Networks — Conditional Dependence

Regression Interpretation:

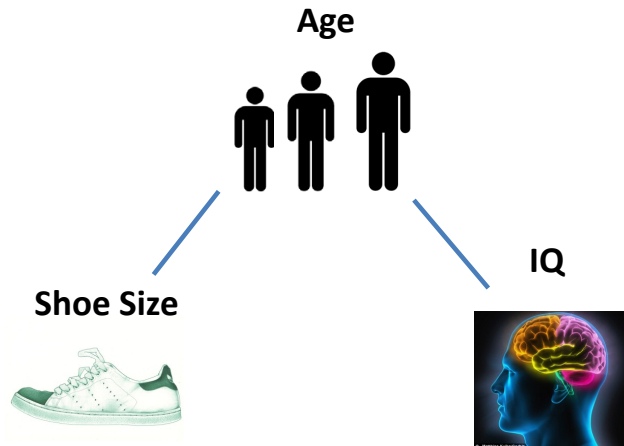
- ▶ Imagine trying to predict the observations in **Node A** (response) by the observations of all other nodes (predictors).
- ▶ **Node B** predictive of **Node A** (with all other nodes in model).
  - ▶ **A** is conditionally dependent on **B**.
  - ▶ Edge.
- ▶ Because of other nodes in model, **Node B** does not add any predictive value for **Node A**.
  - ▶ **A** is conditionally independent of **B**.
  - ▶ No Edge.

## Markov Networks — Conditional Dependence



Correlation.

## Markov Networks — Conditional Dependence



Conditional Dependence.

## Markov Networks — Conditional Dependence

How can we learn conditional dependencies?

- ▶  $A$  and  $B$  are conditionally independent given  $C$  if

$$P(A, B | C) = P(A | C)P(B | C)$$

- ▶ Generally difficult (need to estimate multivariate densities).
- ▶ Alternatively, can use nonparametric approaches, e.g. **conditional mutual information**, but not easy in high dimensions.
- ▶ Often resort to models, or simple measures, such as **partial correlations**...

## Partial Correlation

- ▶ Partial correlation measures the **correlation between  $A$  and  $B$  after the effect of the other variables are removed.**
  - ▶ In our example, this means correlation between shoe size and IQ, **after adjusting for age.**

## Partial Correlation

- ▶ Partial correlation measures the **correlation between  $A$  and  $B$  after the effect of the other variables are removed.**
  - ▶ In our example, this means correlation between shoe size and IQ, **after adjusting for age.**
- ▶ The partial correlation between  $A$  and  $B$  **given  $C$**  is given by:

$$\rho_{AB \cdot C} \equiv \rho(A, B|C) = \frac{\rho_{AB} - \rho_{AC}\rho_{BC}}{\sqrt{1 - \rho_{AC}^2}\sqrt{1 - \rho_{BC}^2}}.$$

## Partial Correlation

- ▶ Partial correlation measures the **correlation between  $A$  and  $B$  after the effect of the other variables are removed**.
  - ▶ In our example, this means correlation between shoe size and IQ, **after adjusting for age**.
- ▶ The partial correlation between  $A$  and  $B$  **given  $C$**  is given by:

$$\rho_{AB \cdot C} \equiv \rho(A, B|C) = \frac{\rho_{AB} - \rho_{AC}\rho_{BC}}{\sqrt{1 - \rho_{AC}^2}\sqrt{1 - \rho_{BC}^2}}.$$

- ▶ Alternatively, **regress  $A$  on  $C$**  and get the residual,  $r_A$ ; do the same for  $B$  to get  $r_B$ . The partial correlation between  $A$  and  $B$  **give  $C$**  is  $\text{Cor}(r_A, r_B)$ .

## Partial Correlation

- ▶ Partial correlation is **symmetric**  $\Rightarrow$  **undirected network**
- ▶ Partial correlation takes values **between -1 and 1**
- ▶ In partial correlation networks, we **draw an edge** between  $A$  and  $B$ , **if the partial correlation between them is large**
- ▶ Calculation of partial correlation is more involved

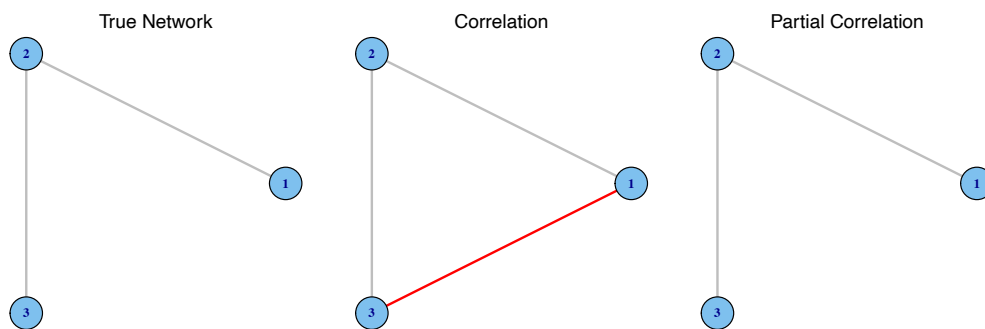


## A Simple Example

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

## A Simple Example

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



## A Larger Example

## A Larger Example

- ▶ A network with 10 nodes and 20 edges

## A Larger Example

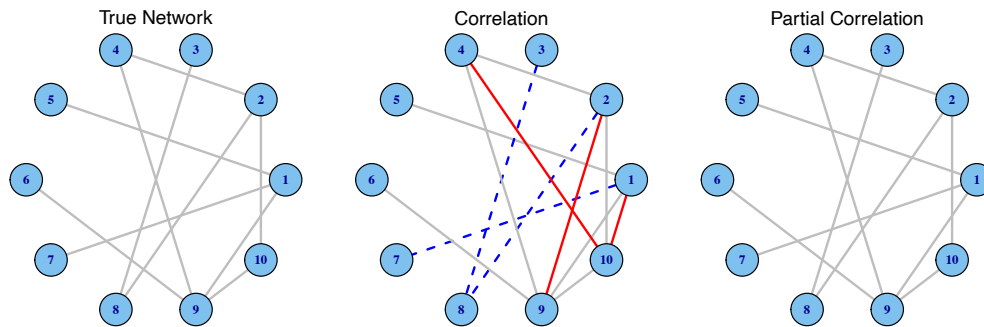
- ▶ A network with 10 nodes and 20 edges
- ▶  $n = 100$  observations

## 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)



## Gaussian Graphical Models (GGMs)

## Partial Correlation for Gaussian Random Variables

## Partial Correlation for Gaussian Random Variables

- ▶ For Gaussian (multivariate normal) random variables, partial correlation between  $X_i$  and  $X_j$  given all other variables is given by the inverse of the (standardized) covariance matrix  $\Sigma$ .

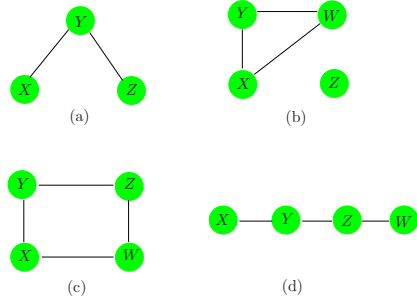
## Partial Correlation for Gaussian Random Variables

- ▶ For Gaussian (multivariate normal) random variables, partial correlation between  $X_i$  and  $X_j$  given all other variables is given by the inverse of the (standardized) covariance matrix  $\Sigma$ .
  - ▶ The  $(i, j)$  entry in  $\Sigma^{-1}$  gives the partial correlation between  $X_i$  and  $X_j$  given all other variables  $X_{\setminus i, j}$ .

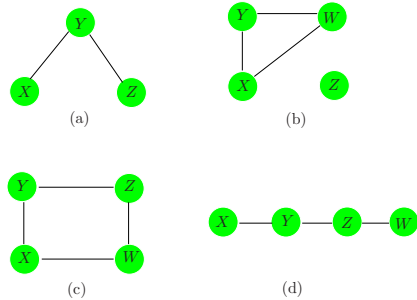
## Partial Correlation for Gaussian Random Variables

- ▶ For Gaussian (multivariate normal) random variables, partial correlation between  $X_i$  and  $X_j$  given all other variables is given by the inverse of the (standardized) covariance matrix  $\Sigma$ .
  - ▶ The  $(i, j)$  entry in  $\Sigma^{-1}$  gives the partial correlation between  $X_i$  and  $X_j$  given all other variables  $X_{\setminus i, j}$ .
  - ▶ Multivariate normal:  $X \sim N(0, \Sigma)$
  - ▶  $\Theta \equiv \Sigma^{-1}$  = inverse covariance/precision/concentration matrix.
  - ▶ Zeros in  $\Theta \implies$  conditional independence!
  - ▶ Edges correspond to non-zeros in  $\Theta$ .

## Partial Correlation for Gaussian Random Variables



## Partial Correlation for Gaussian Random Variables



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

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

## Estimating GGMs

From the discussion so far, to estimate the network, we can

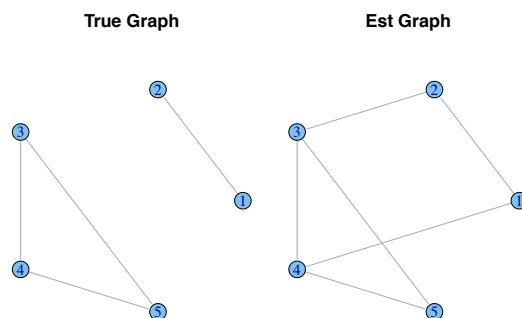
1. Calculate the **empirical covariance matrix**: for (centered)  $n \times p$  data matrix  $X$ ,  $S = (n - 1)^{-1}X^T X$ .
2. **Get the inverse of  $S$** . Non-zero values of  $S^{-1}$  give the edges.

## Estimating GGMs

From the discussion so far, to estimate the network, we can

1. Calculate the **empirical covariance matrix**: for (centered)  $n \times p$  data matrix  $X$ ,  $S = (n - 1)^{-1}X^T X$ .
2. **Get the inverse of  $S$** . Non-zero values of  $S^{-1}$  give the edges.

While simple, this may not work well in practice, even with large samples!





## Estimating GGMs in High Dimensions

Many problems arise in high-dimensional settings, when  $p \gg n$ .

## Estimating GGMs in High Dimensions

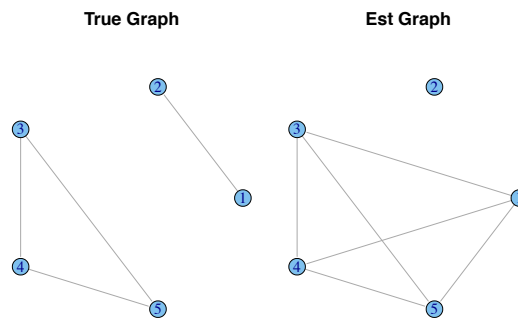
Many problems arise in high-dimensional settings, when  $p \gg n$ .

- ▶ 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 many false positives/negatives.

## Estimating GGMs in High Dimensions

Many problems arise in high-dimensional settings, 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 many false positives/negatives.



## Estimating GGMs in High Dimensions

- ▶ A number of methods have been recently proposed for estimating GGMs in high dimensions.
- ▶ The main idea in most of these methods is to use a regularization penalty, like the lasso.
- ▶ We discuss two approaches:
  - ▶ neighborhood selection
  - ▶ graphical lasso

## The Lasso

- ▶ The lasso involves finding  $\beta$  that minimizes

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

## The Lasso

- ▶ The lasso involves finding  $\beta$  that minimizes

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

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

## The Lasso

- ▶ The lasso involves finding  $\beta$  that minimizes

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

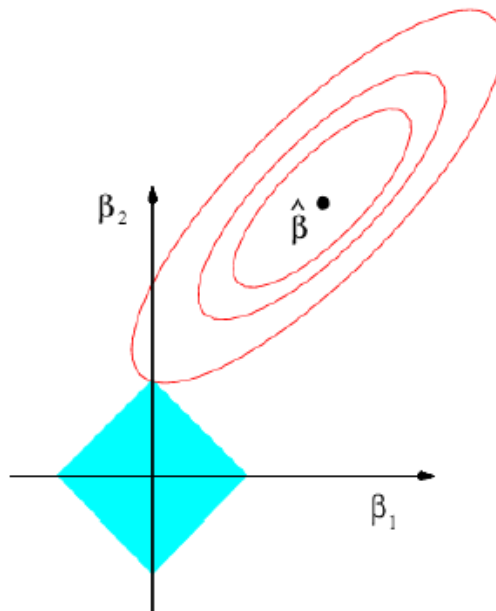
- ▶ Here  $\lambda$  is a **tuning parameter**
  - ▶ When  $\lambda = 0$ , we get least squares!
  - ▶ When  $\lambda$  is very large, we get  $\hat{\beta} = 0$ .
- ▶ Equivalently, find  $\beta$  that minimizes

$$\left\| y - \sum_{k=1}^p X_k \beta_k \right\|^2$$

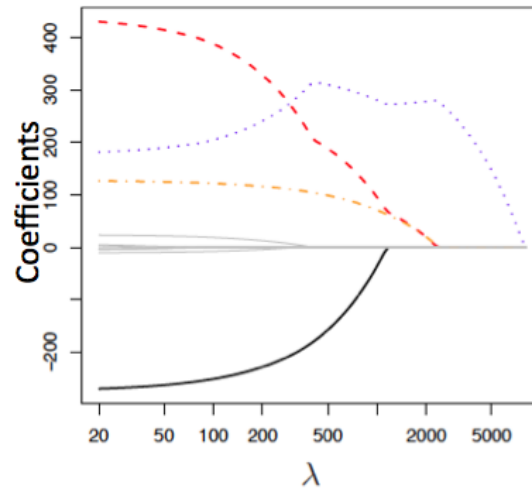
subject to the constraint that

$$\sum_{k=1}^p |\beta_k| \leq s.$$

## A Geometric Interpretation



## Lasso As $\lambda$ Varies



## Estimating GGMs in High Dimensions – Method 1

The idea behind **neighborhood selection**, is to estimate the graph by fitting a **penalized regression of each variable on all other variables**.

- Find **neighbors** of each node  $X_j$  by  $l_1$ -penalized regression or lasso:

$$\underset{\beta^j}{\text{minimize}} \quad \|X_j - X_{\neq j}\beta^j\|_2^2 + \lambda \sum_{k \neq j} |\beta_k^j|$$

## Estimating GGMs in High Dimensions – Method 1

The idea behind **neighborhood selection**, is to estimate the graph by fitting a **penalized regression of each variable on all other variables**.

- ▶ Find **neighbors** of each node  $X_j$  by  $l_1$ -penalized regression or lasso:

$$\underset{\beta^j}{\text{minimize}} \quad \|X_j - X_{\neq j}\beta^j\|_2^2 + \lambda \sum_{k \neq j} |\beta_k^j|$$

- ▶ The final estimate is found by combining all of the edges from these individual regression problems.
  - ▶ Symmetry —  $\beta_k^j$  not always same as  $\beta_j^k$ .
  - ▶ Use min or max rule.

## Estimating GGMs in High Dimensions – Method 2

Estimate a sparse  $\Theta$  via penalized maximum likelihood estimation (MLE).

Graphical Lasso (glasso)

$$\underset{\Theta}{\text{maximize}} \quad \log\det(\Theta) - \text{tr}(S\Theta) - \lambda \|\Theta\|_1$$

- ▶ **Blue**: Log-likelihood;  $\log\det$  denotes the logarithm of the determinant of  $\Theta$  and  $\text{tr}$  the trace (sum of diagonal elements)  $S\Theta$ .
- ▶ **Red**: Penalty term encourages zeros on the off-diagonal elements of  $\Theta$ .

## Comparing the Two Approaches

- ▶ Neighborhood selection is an **approximation for graphical lasso**:
  - ▶ Consider regression of  $X_j$  on  $X_k, j \neq k$
  - ▶ Then, the regression coefficient for neighborhood selection is related to the  $j, k$  element of  $\Theta$ :

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

## Comparing the Two Approaches

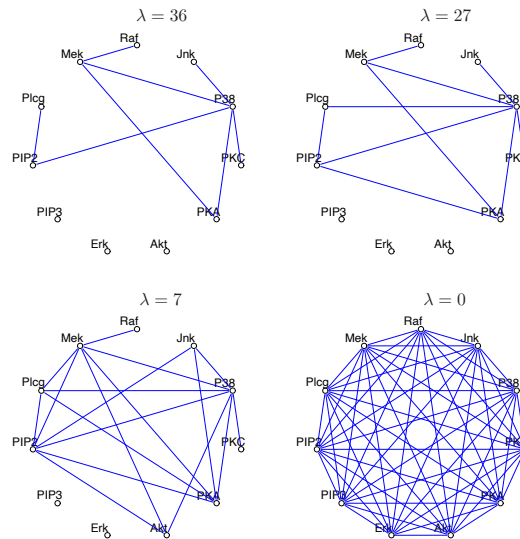
- ▶ Neighborhood selection is an **approximation for graphical lasso**:
  - ▶ Consider regression of  $X_j$  on  $X_k, j \neq k$
  - ▶ Then, the regression coefficient for neighborhood selection is related to the  $j, k$  element of  $\Theta$ :

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

- ▶ Neighborhood selection is computationally more efficient, and may give better estimates, but doesn't give an estimate of  $\Theta$ !

## A Real Example

- ▶ **Flow cytometry** proteomics in single cells (Sachs et al, 2003).
- ▶  $p = 11$  proteins measured in  $n = 7466$  cells



## How to Choose $\lambda$ ?

- ▶  $\lambda$  modulates trade-off between **model fit** and **network sparsity**:
  - ▶  $\lambda = 0$  gives a dense network (no sparsity).
  - ▶ As  $\lambda$  increases, network becomes more sparse.
- ▶ A number of approaches proposed in the literature and used in practice
  1. **Cross-Validation** — tends to yield overly dense networks.
  2. **Extended BIC** — adjusted BIC for high dimensions.
  3. **Controlling the probability of falsely connecting disconnected components** at level  $\alpha$  (Banerjee et al, 2008):

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

( $t_{n-2}(\alpha)$  is the  $(100 - \alpha)\%$  quantile of  $t$ -dist with  $n - 2$  d.f.)

4. **Stability selection** — Choose  $\lambda$  that gives the most **stable network** (R: huge package)



## Other Types of Graphical Models

## Nonparanormal (Gaussian Copula) Models

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

## Nonparanormal (Gaussian Copula) Models

- ▶ Suppose  $X \approx N(0, \Sigma)$ , but there exist monotone functions  $f_j, j = 1, \dots, p$  such that  $[f_1(X_1), \dots, f_p(X_p)] \sim N(0, \Sigma)$ 
  - ▶  $X$  has a nonparanormal distribution  $X \sim NPN_p(f, \Sigma)$ .
  - ▶  $f$  and  $\Sigma$  are parameters of the distribution, and estimated from data.
  - ▶ For continuous distributions, the nonparanormal family is the same as the Gaussian copula family

## Nonparanormal (Gaussian Copula) Models

- ▶ Suppose  $X \approx N(0, \Sigma)$ , but there exist monotone functions  $f_j, j = 1, \dots, p$  such that  $[f_1(X_1), \dots, f_p(X_p)] \sim N(0, \Sigma)$ 
  - ▶  $X$  has a nonparanormal distribution  $X \sim NPN_p(f, \Sigma)$ .
  - ▶  $f$  and  $\Sigma$  are parameters of the distribution, and estimated from data.
  - ▶ For continuous distributions, the nonparanormal family is the same as the Gaussian copula family
- ▶ To estimate the nonparanormal network:
  - transform the data:  $[f_1(X_1), \dots, f_p(X_p)]$
  - 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  $\rho$**  or **Kendall's  $\tau$**

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

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

## 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  $\rho$**  or **Kendall's  $\tau$**

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

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

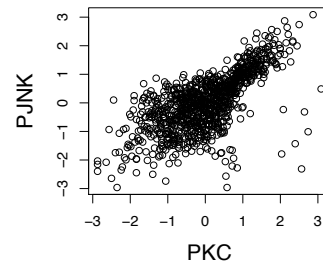
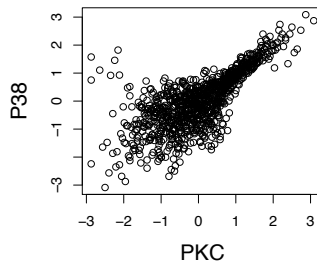
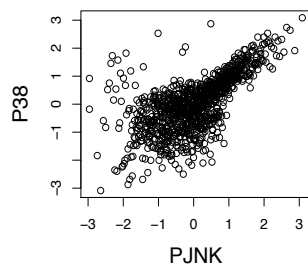
- ▶ If  $X \sim NPN_p(f, \Sigma)$ , then  $\Sigma_{jk} = 2 \sin(\rho_{jk}\pi/6) = \sin(\tau_{jk}\pi/2)$
- ▶ Therefore, we can estimate  $\Sigma^{-1}$  by **plugging in rank-based correlations into graphical lasso** (R-package huge)

## A Real Data Example

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

## A Real Data Example

- ▶ Protein cytometry data for cell signaling (Sachs et al, 2005)
- ▶ Transform the data using a **Gaussian copula** (Liu et al, 2009), giving marginal normality
- ▶ Pairwise relationships still 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
- ▶ In this case, the estimators do not have a closed-form!
- ▶ A special case, which is computationally more tractable, is the class of **pairwise MRFs**

## Pairwise Markov Random Fields

---

<sup>3</sup>Wainwright & Jordan (2008)

## 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 graph  $G$  over the random vector  $X$  is the family of probability distributions  $P(X)$  that can be written as

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

- ▶ For each edge  $(j, k) \in E$ ,  $\phi_{jk}$  is called the **edge potential function**
  - ▶ For discrete random variables, any MRF can be transformed to an MRF with pairwise interactions by introducing additional variables<sup>3</sup>

---

<sup>3</sup>Wainwright & Jordan (2008)

## Graphical Models for Binary Random Variables

## Graphical Models for Binary Random Variables

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

## Graphical Models for Binary Random Variables

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

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



## Graphical Models for Binary Random Variables

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

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

- ▶ 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

---

<sup>4</sup>Ravikumar et al (2010)

## Graphical Models for Binary Random Variables

- ▶ We can consider a **neighborhood selection**<sup>4</sup> approach with an  $\ell_1$  (lasso) penalty to find the neighborhood of each node  
 $N(j) = \{k \in V : (j, k) \in E\}$

---

<sup>4</sup>Ravikumar et al (2010)

## Graphical Models for Binary Random Variables

- ▶ We can consider a **neighborhood selection**<sup>4</sup> approach with an  $\ell_1$  (lasso) penalty to find the neighborhood of each node  
 $N(j) = \{k \in V : (j, k) \in E\}$
- ▶ For  $j = 1, \dots, p$ , need to solve (after some algebra)

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

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

---

<sup>4</sup>Ravikumar et al (2010)

## Graphical Models for Binary Random Variables

- ▶ We can consider a **neighborhood selection**<sup>4</sup> approach with an  $\ell_1$  (lasso) penalty to find the neighborhood of each node  $N(j) = \{k \in V : (j, k) \in E\}$
- ▶ For  $j = 1, \dots, p$ , need to solve (after some algebra)

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

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

<sup>4</sup>Ravikumar et al (2010)

## Other Non-Gaussian Distributions

- ▶ Assume a **pairwise graphical model**

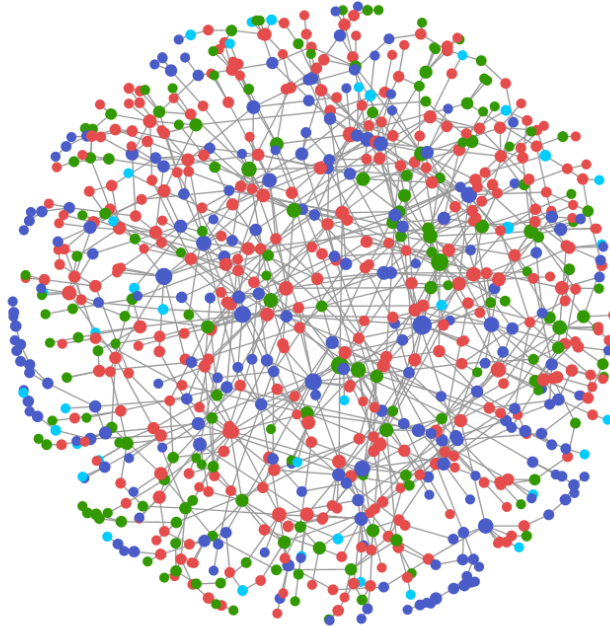
$$P(X) \propto \exp \left\{ \sum_{j \in V} \theta_j \phi_j(X_j) + \sum_{(j,k) \in E} \theta_{jk} \phi_{jk}(X_j, X_k) \right\}$$

- ▶ 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<sup>5</sup>
  - ▶ We can even learn networks with multiple types of nodes (gene expression, SNPs, and CNVs)<sup>6</sup>

<sup>5</sup>Yang et al (2012)

<sup>6</sup>Yang et al (2014), Chen et al (2015)

## Mixed Graphical Models



## A General Approach for Estimation of Graphical Models

- ▶ Consider  $n$  iid observations from a  $p$ -dimensional random vector  $\mathbf{x} = (X_1, \dots, X_p) \sim \mathcal{P}$
- ▶ Consider the (undirected) graph  $G = (V, E)$  with vertices  $V = \{1, \dots, p\}$

- ▶ Want to estimate edges  $E \subset V \times V$  that satisfy  $\forall j \in V, \exists N(j)$  such that:

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

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

## Estimating Conditional Independencies

Question: how to condition?

## Estimating Conditional Independencies

Question: how to condition?

- ▶ **Approach 1:** Estimate the joint density  $f(X_1, \dots, X_p)$ ; then get the conditionals  $f_j(X_j | X_{-j})$

## Estimating Conditional Independencies

Question: how to condition?

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

## Estimating Conditional Independencies

Question: how to condition?

- ▶ **Approach 1:** Estimate the joint density  $f(X_1, \dots, X_p)$ ; then get the conditionals  $f_j(X_j | X_{-j})$ 
  - ▶ Efficient, coherent
  - ▶ Computationally challenging
  - ▶ Restrictive: how many joint distributions do you know?
  - ▶ Hard to check if assumptions hold!
- ▶ **Approach 2:** Estimate the conditionals directly  $f_j(X_j | X_{-j})$

## Estimating Conditional Independencies

Question: how to condition?

- ▶ **Approach 1:** Estimate the joint density  $f(X_1, \dots, X_p)$ ; then get the conditionals  $f_j(X_j | X_{-j})$ 
  - ▶ Efficient, coherent
  - ▶ Computationally challenging
  - ▶ Restrictive: how many joint distributions do you know?
  - ▶ Hard to check if assumptions hold!
- ▶ **Approach 2:** Estimate the conditionals directly  $f_j(X_j | X_{-j})$ 
  - ▶ Computationally easy
  - ▶ Leads to easy & flexible models (regression)!
  - ▶ May not be efficient or coherent

## A Semi-parametric Approach

## A Semi-parametric Approach

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

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

## A Semi-parametric Approach

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

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

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



## A Semi-parametric Approach

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

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

- ▶ Then if  $f_{jk}(X_k) = f_{kj}(X_j) = 0$ , we conclude that  $X_j$  and  $X_k$  are **conditionally independent**, given the other variables
- ▶ In other words, we **assume that conditional distributions and conditional means depend on the same set of variables**

## A Semi-parametric Approach

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

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

- ▶ Then if  $f_{jk}(X_k) = f_{kj}(X_j) = 0$ , we conclude that  $X_j$  and  $X_k$  are **conditionally independent**, given the other variables
- ▶ In other words, we **assume that conditional distributions and conditional means depend on the same set of variables**
- ▶ We then use a semi-parametric approach for estimating the conditional dependencies

## SpaCE JAM<sup>7</sup>

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

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

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

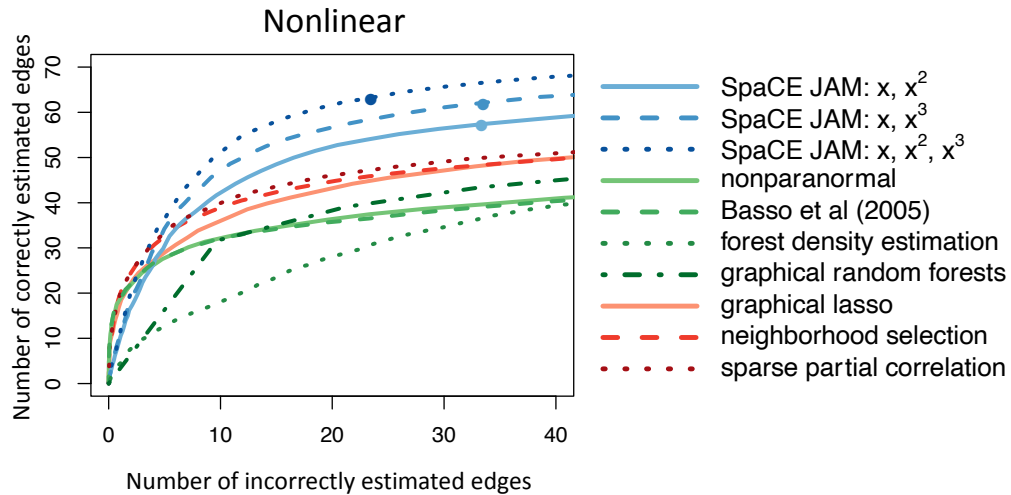
<sup>7</sup>Voorman et al (2014), R-package spacejam

## 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  $E[X_j | X_{\setminus j}]$  non-parametrically
  - ▶ Theoretical properties have not yet been justified

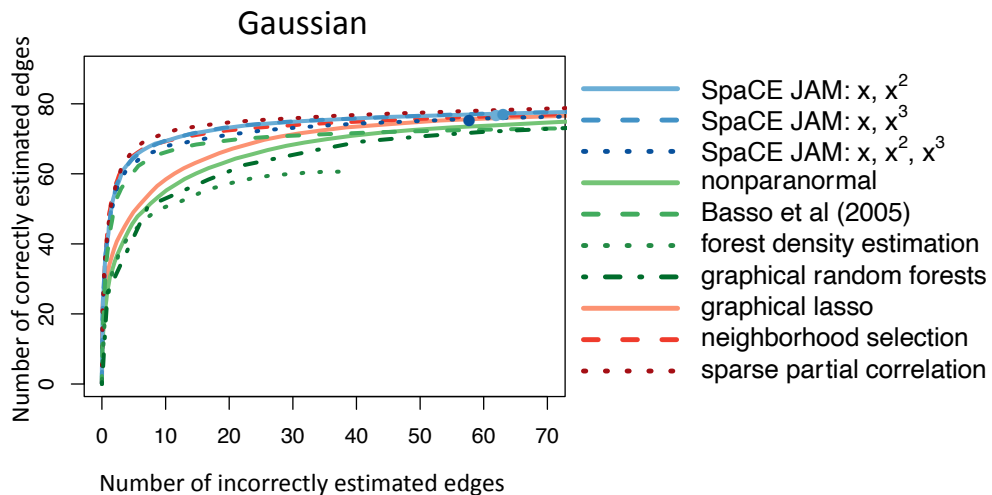
## Comparison on Simulated Data

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

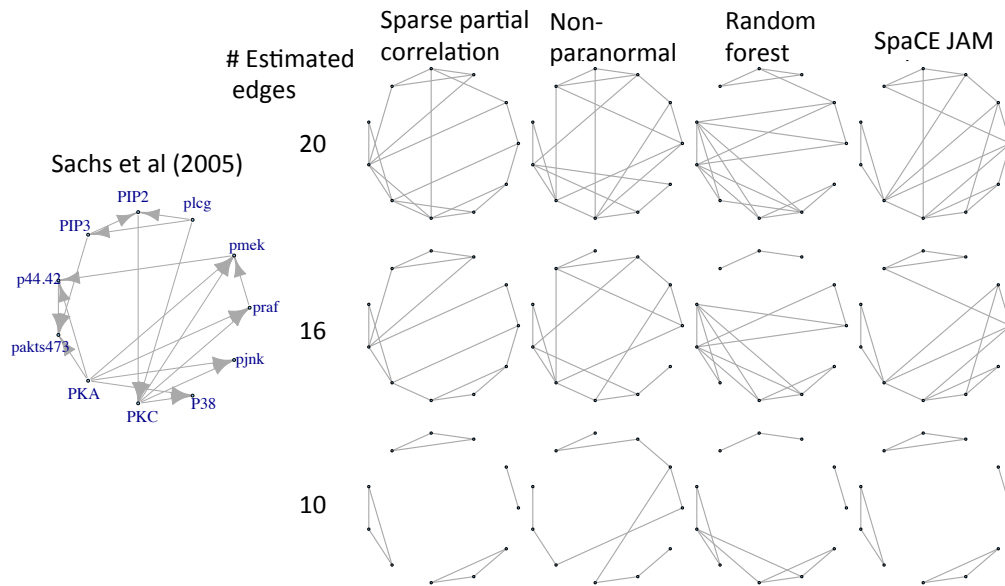


## Comparison on Simulated Data

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



## Estimation of Cell Signaling Network



## Other Extensions of GGMs

- ▶ Multiple Graphical Models
  - ▶ For groups of observations, estimate graphical models with shared structure across groups and individual structure within groups.
- ▶ Time Varying Graphical Models
  - ▶ Smoothly varying graph over time estimated via local kernel smoothers.
  - ▶ Change points in graph structure over time estimated via fusion penalties.
- ▶ Latent Variable Graphical Models
  - ▶ Assume observed features are dependent on latent variables which exhibit a low-rank effect. Estimate a sparse (graph structure) plus low-rank inverse covariance matrix.

## Pathway & Network Analysis of Omics Data: Learning Directed Networks

Ali Shojaie  
Department of Biostatistics  
University of Washington  
[faculty.washington.edu/ashojaie](http://faculty.washington.edu/ashojaie)

Summer Institute for Statistical Genetics – 2021

## Bayesian Networks

## Bayesian Networks

- ▶ Bayesian networks are a special class of graphical models defined on **directed acyclic graphs**.

## Bayesian Networks

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

## Bayesian Networks

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

## Bayesian Networks

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

## Bayesian Networks

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

## Why Bayesian Networks?



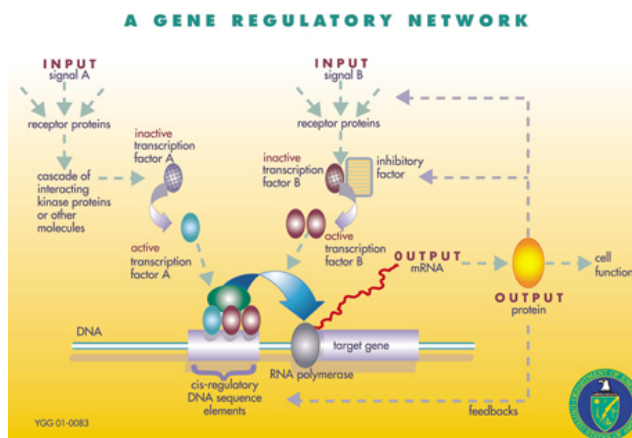
## Why Bayesian Networks?

Many biological networks include directed edges:

## Why Bayesian Networks?

Many biological networks include directed edges:

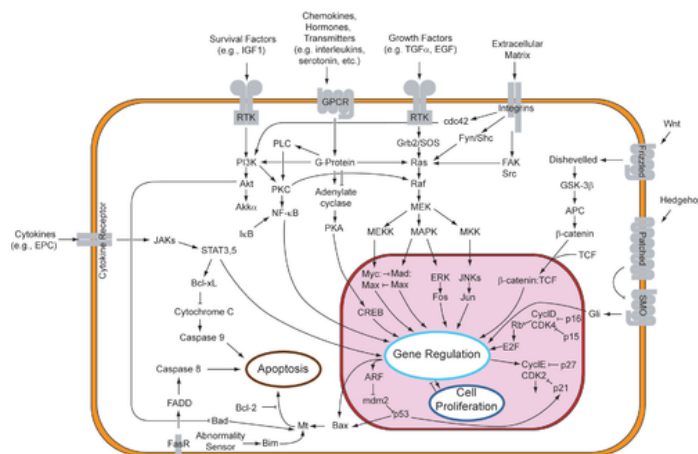
- ▶ In **gene regulatory networks**, protein products of **transcription factors** can alter the expression of **target genes**, but the target genes (usually) don't have a direct effect on the expression of transcription factors



## Why Bayesian Networks?

Many biological networks include directed edges:

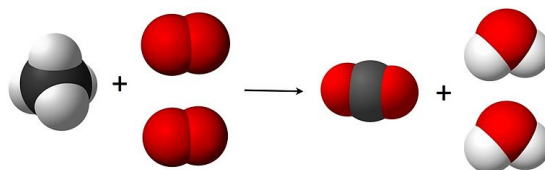
- ▶ In **cell signaling networks**, the signal from the cell's environment is transducted into the cell, and results e.g. in (global) changes in gene expression, but gene expression may not affect the environmental factors



## Why Bayesian Networks?

Many biological networks include directed edges:

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



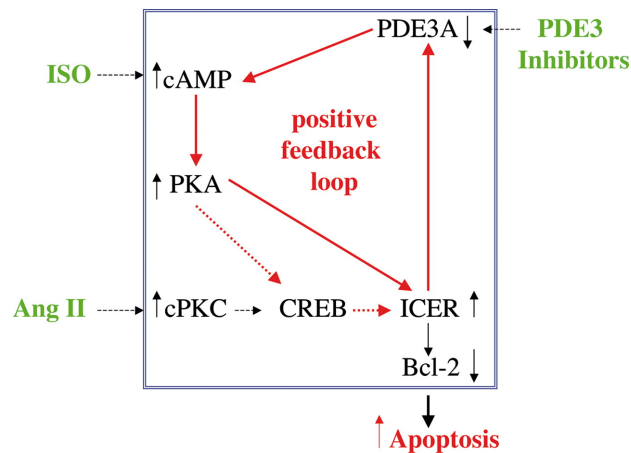
## Why Bayesian Networks?

However, biological networks **may not be DAGs**:

## 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?

## What's the Difference?

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

## What's the Difference?

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

## What's the Difference?

- ▶ Bayesian networks are widely used to model **causal relationships** between variables.
- ▶ Undirected networks (e.g. GGM) provide information about **associations** among variables; while this greatly helps in the study of biological systems, in some cases, they are not enough (e.g. drug development).
- ▶ The main difference is the **direction of the edges**; however, it turns out that there are also some differences in terms of **structure/skeleton** of the network (more on this later).

## What's the Difference?

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

## Why is estimation more difficult?

- ▶ Estimation of Bayesian networks requires estimating both the **skeleton** of the network (i.e. whether there is an edge between  $i$  and  $j$ ) and also the **direction** of the edges.

## Why is estimation more difficult?

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



## 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  $X_1 \rightarrow X_2$  and  $X_2 \rightarrow X_1$ , so basically what we see is:



## Directed Graphs: Some Terminology

- ▶ The **parents** of node  $j$  are  $\{k : k \rightarrow j\}$ , we denote this by  $\text{pa}_j$  or  $\text{pa}(j)$
- ▶ The **children** of node  $j$  are  $\{k : j \rightarrow k\}$
- ▶ Two vertices connected by an edge are called **adjacent**

## Directed Graphs: Some Terminology

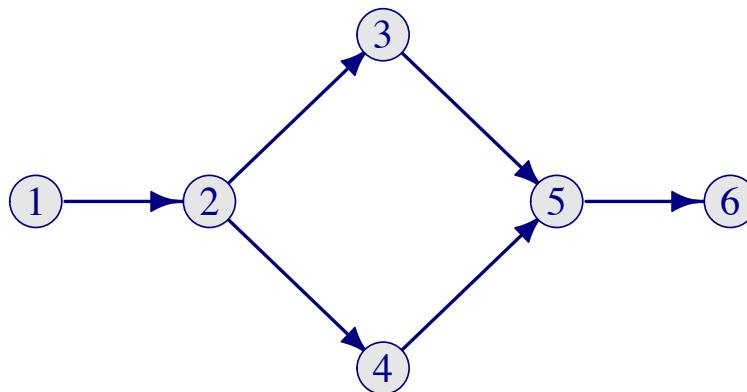
- ▶ The **parents** of node  $j$  are  $\{k : k \rightarrow j\}$ , we denote this by  $\text{pa}_j$  or  $\text{pa}(j)$
- ▶ The **children** of node  $j$  are  $\{k : j \rightarrow k\}$
- ▶ Two vertices connected by an edge are called **adjacent**
- ▶ A **path** between two nodes  $i$  and  $j$  is a **sequence of distinct adjacent nodes**:
  - ▶ e.g.  $i \leftarrow k_1 \rightarrow k_2 \rightarrow k_3 \leftarrow j$
  - ▶ In a DAG with  $p$  nodes, there cannot be a path longer than  $p - 1$  (why?)
  - ▶ There can be multiple paths between two nodes



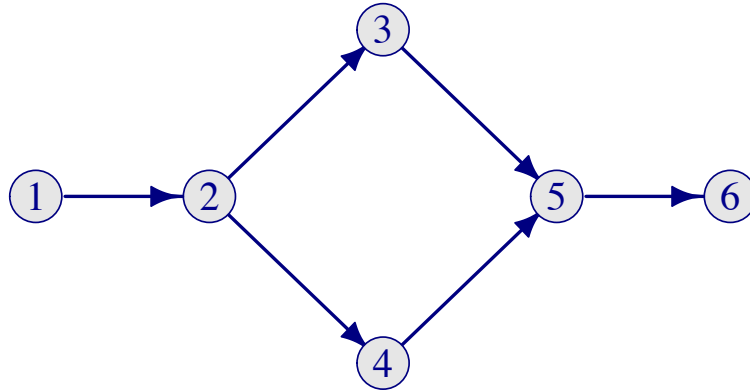
## Directed Graphs: Some Terminology

- ▶ The **parents** of node  $j$  are  $\{k : k \rightarrow j\}$ , we denote this by  $\text{pa}_j$  or  $\text{pa}(j)$
- ▶ The **children** of node  $j$  are  $\{k : j \rightarrow k\}$
- ▶ Two vertices connected by an edge are called **adjacent**
- ▶ A **path** between two nodes  $i$  and  $j$  is a **sequence of distinct adjacent nodes**:
  - ▶ e.g.  $i \leftarrow k_1 \rightarrow k_2 \rightarrow k_3 \leftarrow j$
  - ▶ In a DAG with  $p$  nodes, there cannot be a path longer than  $p - 1$  (why?)
  - ▶ There can be multiple paths between two nodes
- ▶  $i$  is an **ancestor** of  $j$  if there is a **directed path** of length  $\geq 1$  from  $i$  to  $j$ :  $i \rightarrow \dots \rightarrow j$  (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



## Directed Graphs: Some Terminology



- ▶ What are **parents/children** of  $\{1, \dots, 5\}$ ?
- ▶ 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 **colliders** (aka “inverted forks”):

## Directed Graphs: Some Terminology

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

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

$$i \dots \rightarrow k \leftarrow \dots j$$

## Directed Graphs: Some Terminology

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

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

$$i \dots \rightarrow k \leftarrow \dots j$$

- ▶  $k$  is a **non-collider** if it is not an end-point, and is not a collider on a path:

- ▶  $i \dots \leftarrow k \leftarrow \dots j$

- ▶  $i \dots \rightarrow k \rightarrow \dots j$

- ▶  $i \dots \leftarrow k \rightarrow \dots j$

## Directed Graphs: Some Terminology

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

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

$$i \dots \rightarrow k \leftarrow \dots j$$

- ▶  $k$  is a **non-collider** if it is not an end-point, and is not a collider on a path:

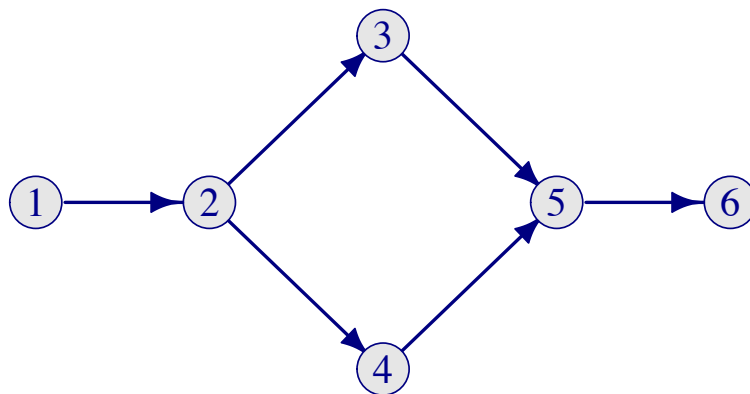
- ▶  $i \dots \leftarrow k \leftarrow \dots j$

- ▶  $i \dots \rightarrow k \rightarrow \dots j$

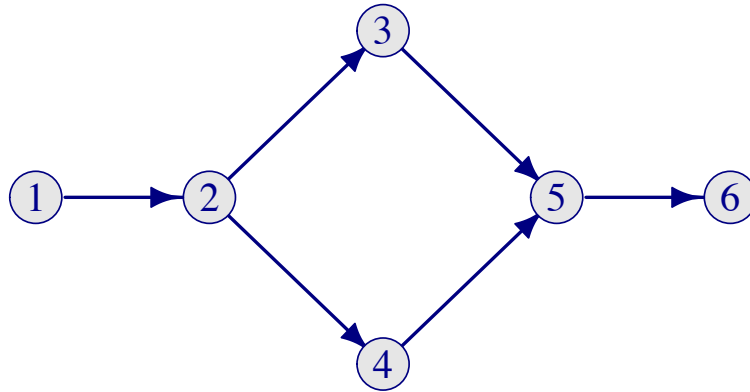
- ▶  $i \dots \leftarrow k \rightarrow \dots j$

- ▶ 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

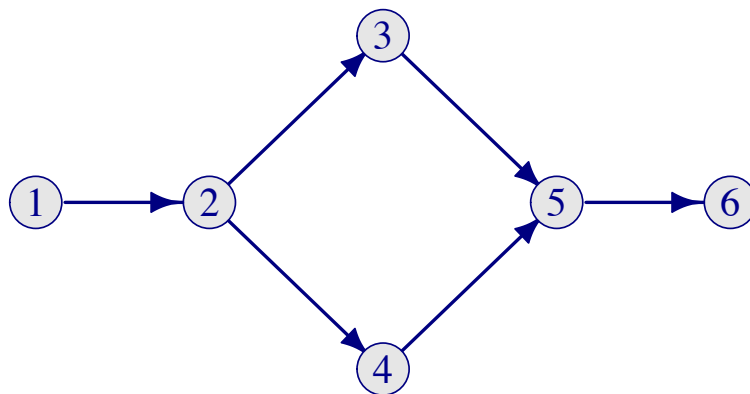


## Directed Graphs: Some Terminology



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

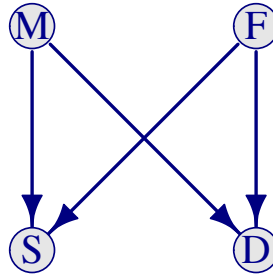
## Directed Graphs: Some Terminology



- ▶ 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?

## Estimating Directed Graphs

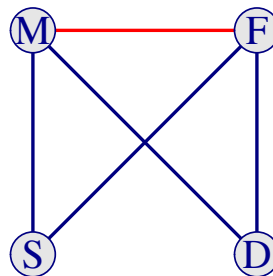
- ▶ The presence of **colliders** makes the estimation of directed graphs very challenging...



- ▶ 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**

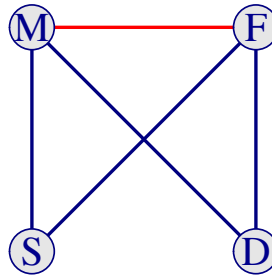
## Estimating Directed Graphs

- ▶ **Conditioning on all other nodes**, gives additional **moral** (!!)  
edges ( $\Rightarrow$  **moral graph**)



## Estimating Directed Graphs

- ▶ Conditioning on all other nodes, gives additional moral (!!)  
edges ( $\Rightarrow$  moral graph)



- ▶ Learning the skeleton of DAGs from observational data requires finding right conditioning set
  - ▶ Naively, this is done by searching over all possible subset of other  $p - 2$  nodes — NP-hard with complexity  $O(2^{p^2})!!$

## Estimation of DAGs from Observational Data

Two general classes of algorithms for estimating DAGs:

- ▶ **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)
  - ▶ Greedy search to find the best scoring graph (Hill Climbing)
- ▶ **“hybrid” methods**
  - ▶ Usually first find the Markov blanket (e.g. the moral graph)
  - ▶ Then search in a restricted space (Max-Min Hill Climbing)

## 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!!
- ▶ These methods find the **DAG skeleton** (*conditional independence is symmetric*) — will talk about direction later

## PC Algorithm (Spirtes et al, 1993)

- ▶ One of the first algorithms for learning structure of DAGs
- ▶ Efficient implementations that allow for learning DAG structures with  $p$  up to  $\sim 1000$ 
  - ▶ R-package `pcaIlg` (Kalisch & Buhlmann, 2007)
- ▶ The algorithm **starts with a complete graph** (i.e. fully connected)
- ▶ 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)

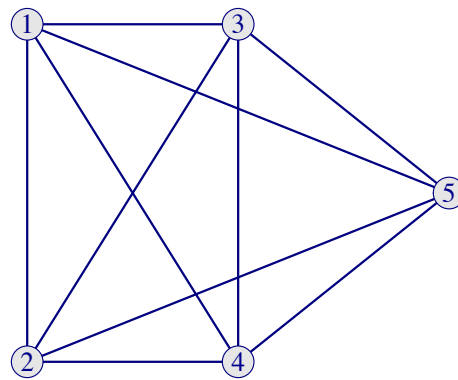
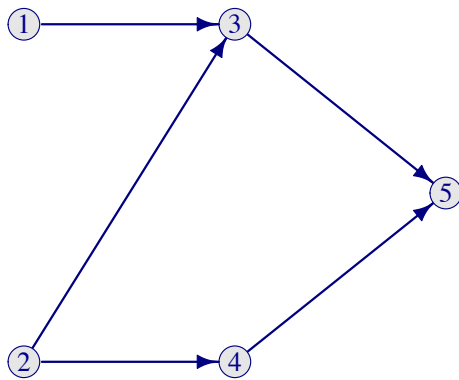
Start with a complete undirected graph, and set  $i = 0$

Repeat

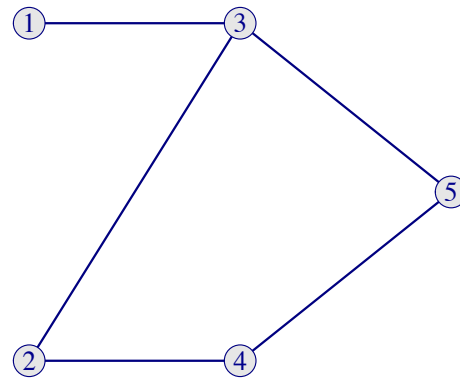
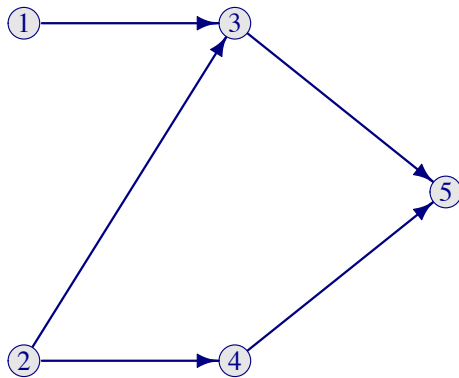
- ▶ For each  $j \in V$
- ▶ For each  $j' \in \text{ne}(j)$
- ▶ Determine if  $\exists S \subset \text{ne}(j) \setminus \{j'\}$  with  $|S| = i$ 
  - ▶ Test for CI: is  $X_j \perp\!\!\!\perp X_{j'} \mid S$ ?
  - ▶ If such an  $S$  exists, then set  $S_{jj'} = S$ , remove  $j - j'$  edge
- ▶  $i = i +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

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

## 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** ( $|ne_j| < 3 \forall j$ )

## Analysis of Protein Flow Cytometry using pcalg

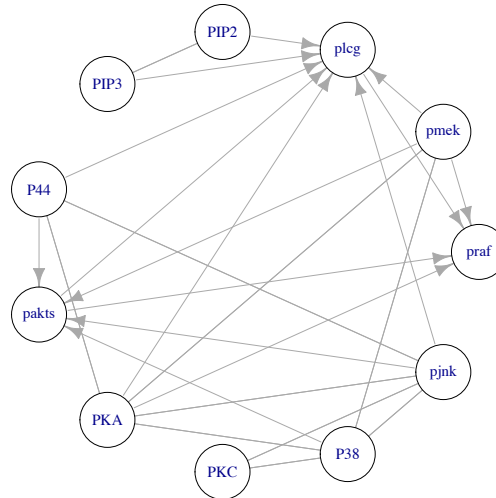
```

> 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  $\alpha$  (alpha), the **probability of false positive** for selecting edges.
  - ▶ Larger values of  $\alpha$  allow more edges (not adjusted for multiple comparisons)
  - ▶ The algorithm works faster when  $\alpha$  is small

## Analysis of Protein Flow Cytometry using pcalg

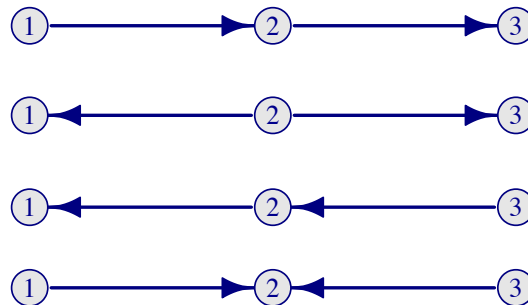
PC Algorithm



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

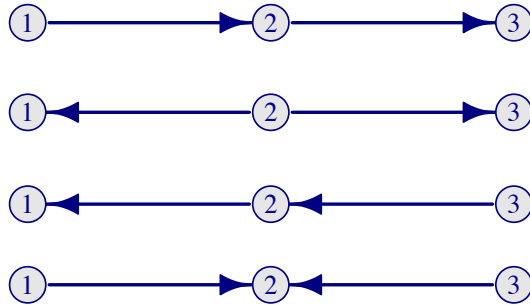
## Markov Equivalence

Consider the following 4 graphs



## Markov Equivalence

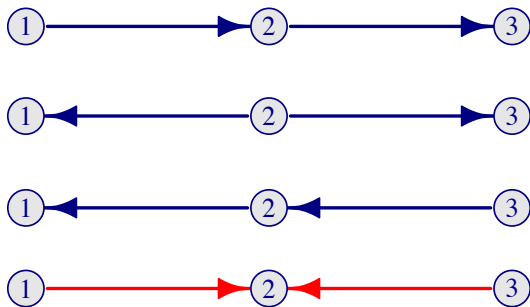
Consider the following 4 graphs



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

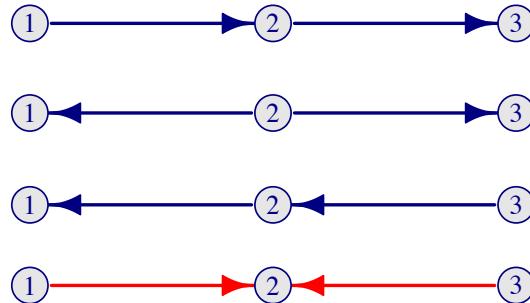
## Markov Equivalence

Consider the following 4 graphs



## Markov Equivalence

Consider the following 4 graphs



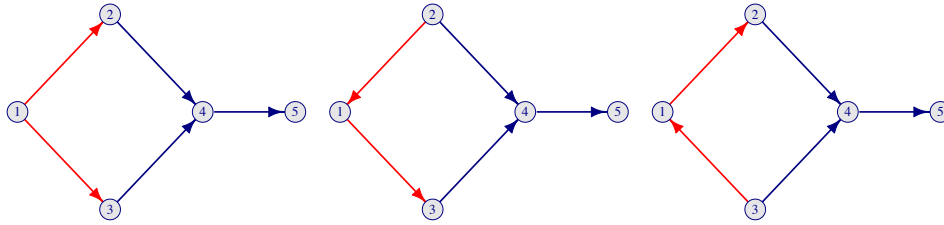
In the first 3 graphs,  $X_1 \perp\!\!\!\perp X_3 \mid X_2$ ?

Two graphs that imply the same CI relationships via d-separation are called **Markov equivalent**

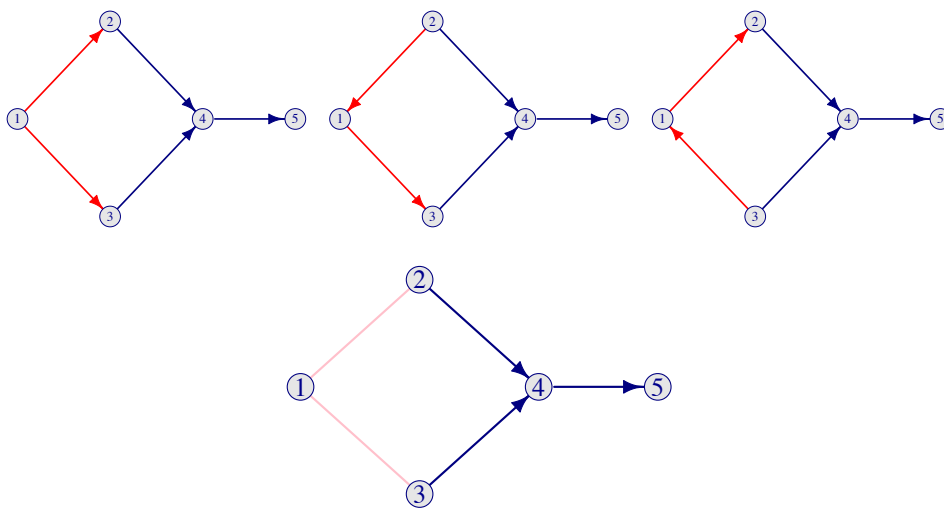
## Representation of Markov Equivalence

- ▶ 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



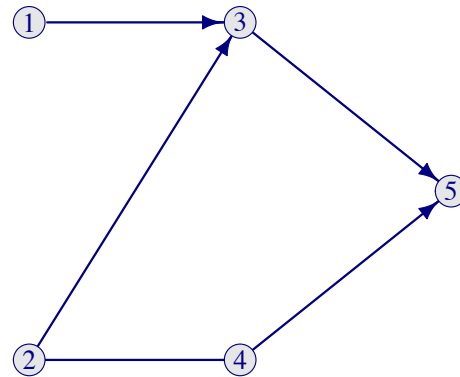
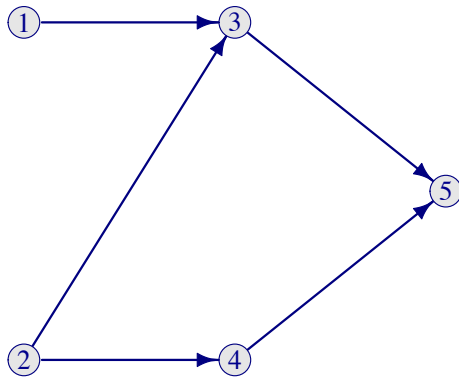
## CPDAGs



## Finding Partial Directions in DAGs

- ▶ Partial directions are 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



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

## The bnlearn package

- ▶ There are a couple of R-packages for learning (CP)DAGs, including `pclag`, `bnlearn`, `deal`
- ▶ `bnlearn` implements a number of estimation methods, both constraint-based and search-based:
  - ▶ constraint-based algorithms:
    - ▶ Grow-Shrink (GS)
    - ▶ Incremental Association Markov Blanket (IAMB)
    - ▶ Fast Incremental Association (Fast-IAMB)
    - ▶ Interleaved Incremental Association (Inter-IAMB)
  - ▶ score-based algorithms:
    - ▶ Hill Climbing (HC)
    - ▶ Tabu Search (Tabu)
  - ▶ hybrid 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  $\alpha$  (alpha), the **false positive probability** for selecting edges
- ▶ `gs` (and other structure-based methods) find a PC DAG
- ▶ `hc` gives a directed graph (with highest score)
  - ▶ Multiple criteria for choosing the “best” graph
  - ▶ 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

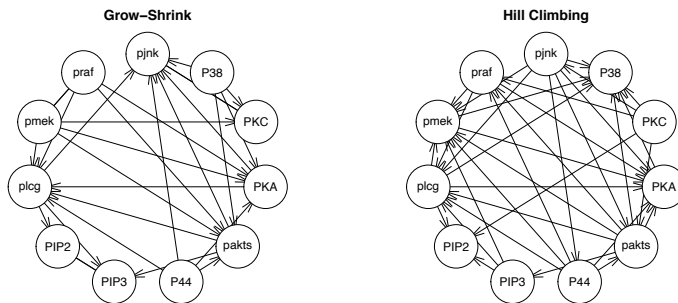
```
> dag2
Bayesian network learned via Score-based methods

model:
  [PKC|pjdk|PKC] [P44|pjdk] [pakts|P44:PKC:pjdk] [praf|P44:pakts:PKC] [PIP3|pakts
  [plcg|praf:PIP3:P44:pakts:pjdk] [pmek|praf:plcg:PIP3:P44:pakts:pjdk]
  [PIP2|plcg:PIP3:PKC] [PKA|praf:pmek:plcg:P44:pakts:pjdk]
  [P38|pmek:plcg:pakts:PKA:PKC:pjdk]
nodes:                               11
arcs:                                 35
  undirected arcs:                    0
  directed arcs:                      35
average markov blanket size:          8.00
average neighbourhood size:           6.36
average branching factor:             3.18

learning algorithm:                   Hill-Climbing
score:                                Bayesian Information Criterion (Gaussian)

penalization coefficient:             4.459057
tests used in the learning procedure: 505
optimized:                            TRUE
```

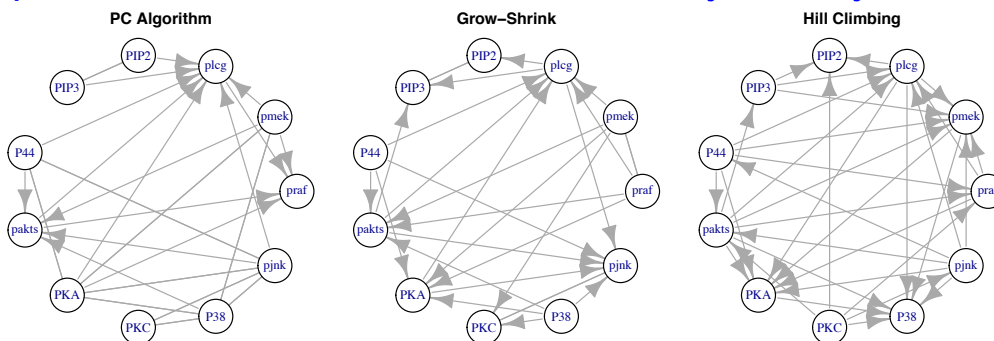
## Analysis of Protein Flow Cytometry using bnlearn



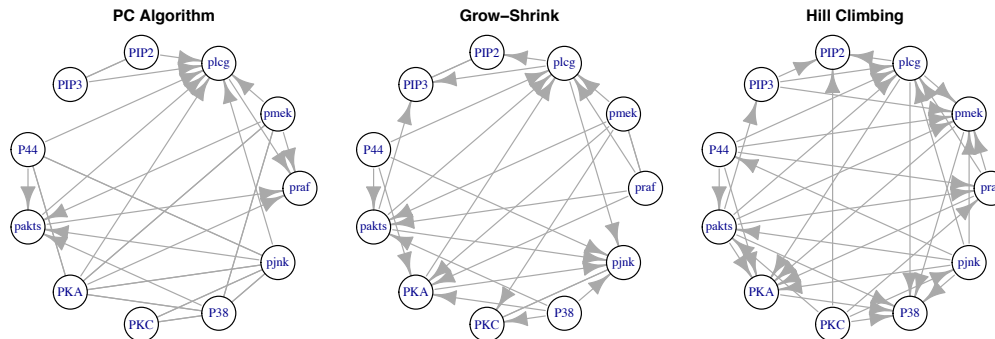
The two graphs are quite different

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

## Comparison of Results for Protein Flow Cytometry Data



## 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

- ▶ Causal relationships (and probability distributions) on DAGs can be represented using **structural equation models**

$$X_i = f_i(\text{pa}_i, \gamma_i), \quad i = 1, \dots, p$$

- ▶ And, for Gaussian random variables, we can write

$$X_i = \sum_{j \in \text{pa}_i} \rho_{ji} X_j + \gamma_i, \quad i = 1, \dots, p$$

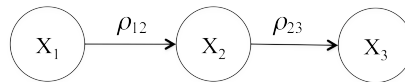
## Penalized Likelihood Estimation of DAGs

- Causal relationships (and probability distributions) on DAGs can be represented using **structural equation models**

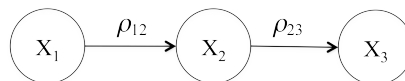
$$X_i = f_i(\text{pa}_i, \gamma_i), \quad i = 1, \dots, p$$

- And, for Gaussian random variables, we can write

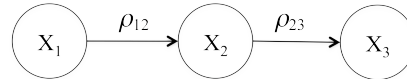
$$X_i = \sum_{j \in \text{pa}_i} \rho_{ji} X_j + \gamma_i, \quad i = 1, \dots, p$$



## Penalized Likelihood Estimation of DAGs

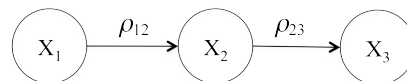


## Penalized Likelihood Estimation of DAGs



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

## 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

---

<sup>1</sup>Shojaie & Michailidis (2010)

## Penalized Likelihood Estimation of DAGs

- ▶ It turns out that  $\Lambda = (I - A)^{-1}$ , where  $A$  is the weighted adjacency matrix of the DAG<sup>1</sup>
- ▶ Thus, for Gaussian random variables, **if we know the ordering of the variables** (which is a BIG assumption!)

after some math...

we can estimate the adjacency matrix of DAGs, by minimizing the log-likelihood as a function of  $A$ :

$$\hat{A} = \arg \min_{A \in \mathcal{A}} \{ \text{tr}[(I - A)^T(I - A)S] \}$$

---

<sup>1</sup>Shojaie & Michailidis (2010)

## Penalized Likelihood Estimation of DAGs

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

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

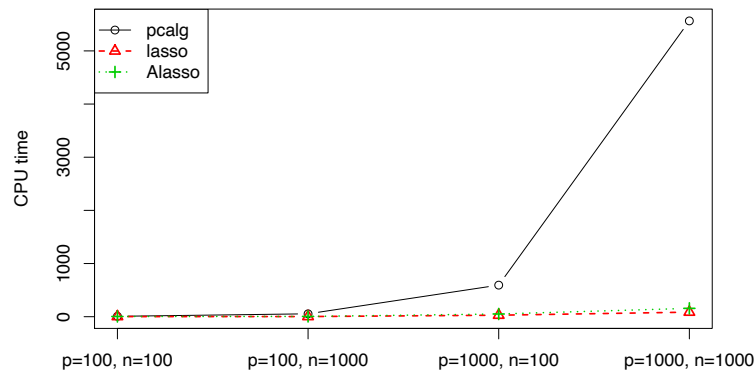
- ▶ As in **glasso**,  $\lambda$  controls the sparsity;  $\lambda = \frac{2}{\sqrt{n}} Z_{\alpha/(2p^2)}$  controls a false positive probability at level  $\alpha$

## Computational Complexity

- ▶ Compared to `pcaIlg`, 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

## Computational Complexity

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

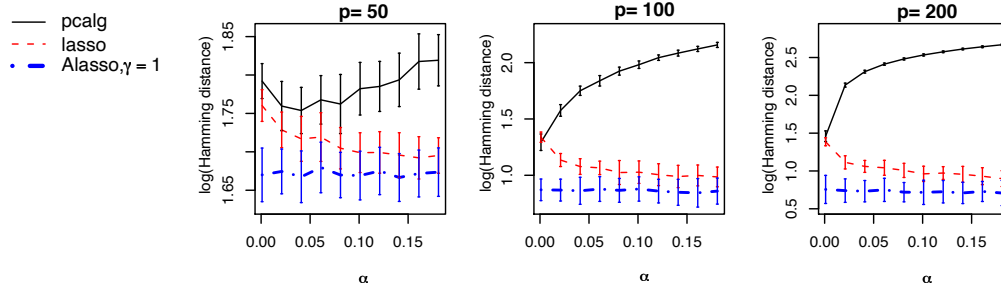


## Simulations

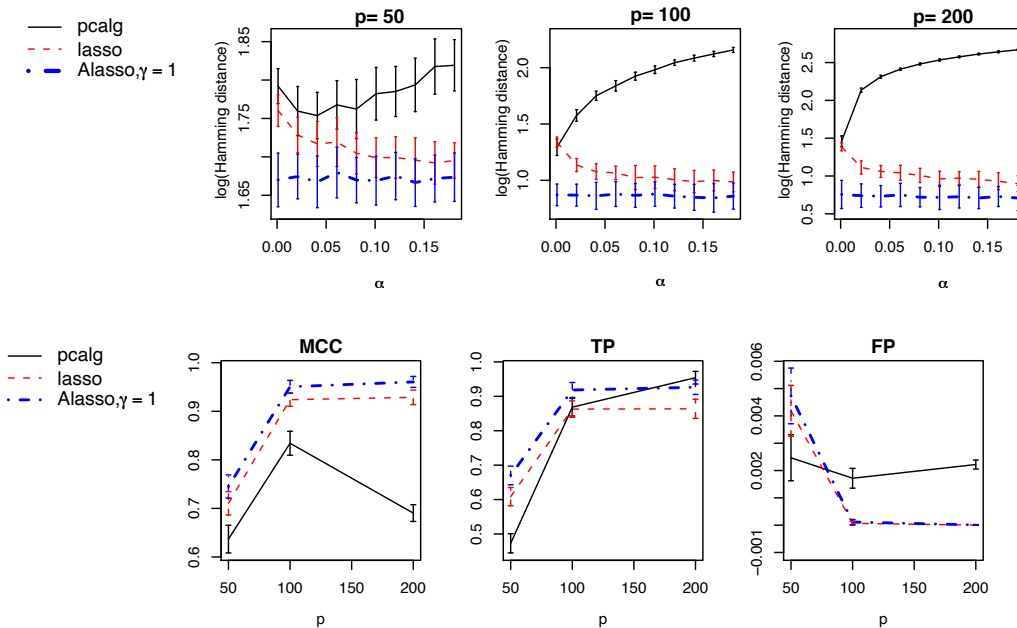
- Settings:
  - $p = 50, 100, 200$
  - $n = 100$
  - Total number of edges in the network =  $n$
  - 100 repetitions
- Performance Criteria
  1. Matthew's Correlation Coefficient (**MCC**): ranges between  $-1$  (worst fit) and  $1$  (best fit), similar to  $F_1$
  2. Structural Hamming Distance (**SHD**): sum of false positive and false negatives
  3. True positive and false positive rates
- Tuning parameter for both PC-Algorithm and penalized likelihood method based on false positive error  $\alpha$



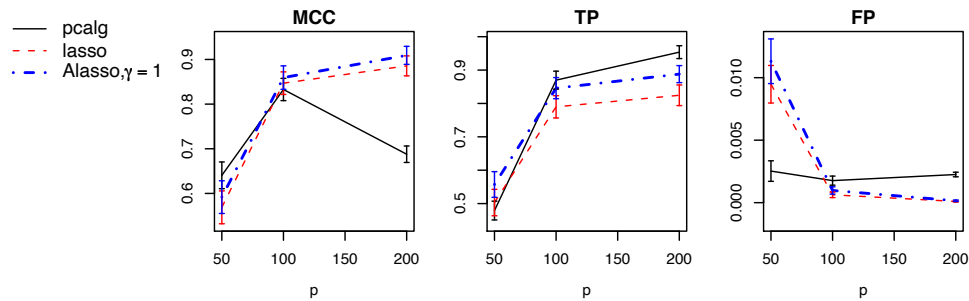
## Gaussian Observations



## Gaussian Observations



## Random Ordering of Variables

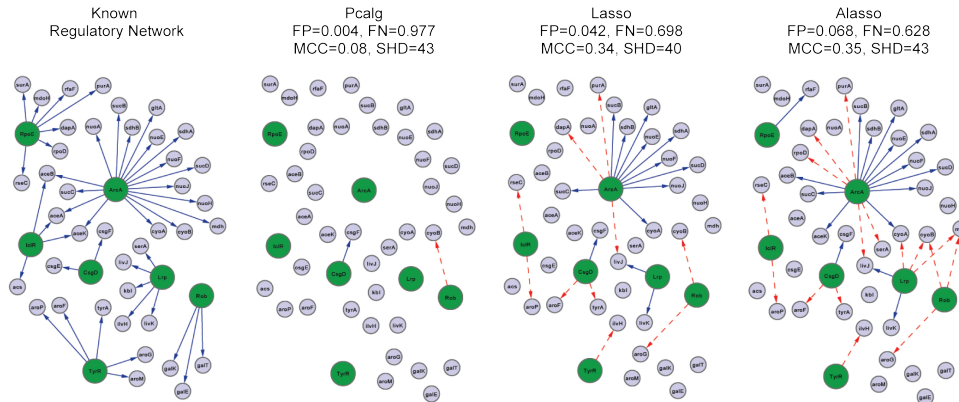


## Regulatory Network of E-Coli

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

## 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



## DAGs for Time Series Data

## Time Series Data: A setting where ordering is known

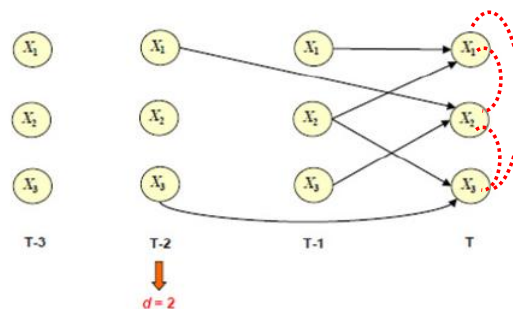
- ▶  $p$ -dimensional, discrete time, stationary process

$$X^t = \{X_1^t, \dots, X_p^t\}$$

$$X^t = A_1 X^{t-1} + \dots + A_d X^{t-d} + \epsilon^t, \quad \epsilon^t \stackrel{i.i.d.}{\sim} N(0, \Sigma_\epsilon) \quad (1)$$

- ▶  $A_1, \dots, A_d$ :  $p \times p$  transition matrices (solid, directed edges)
- ▶  $\Sigma_\epsilon^{-1}$ : contemporaneous dependence (dotted, undirected edges)

## DAGs for Time Series Data



Network Granger causality (NGC)

## Network Granger Causality with VARs

- ▶  $X_1, \dots, X_p$ : time series for  $p$  variables
- ▶  $X^t = (X_1^t, \dots, X_p^t)'$ : realizations at time  $t$

## Network Granger Causality with VARs

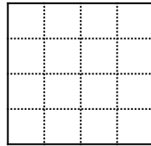
- ▶  $X_1, \dots, X_p$ : time series for  $p$  variables
- ▶  $X^t = (X_1^t, \dots, X_p^t)'$ : realizations at time  $t$
- ▶ **VAR** model for NGC:

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

## Network Granger Causality with VARs

- ▶  $X_1, \dots, X_p$ : time series for  $p$  variables
- ▶  $X^t = (X_1^t, \dots, X_p^t)'$ : realizations at time  $t$
- ▶ **VAR** model for NGC:


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

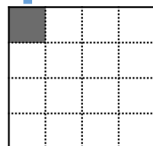


## Network Granger Causality with VARs

- ▶  $X_1, \dots, X_p$ : time series for  $p$  variables
- ▶  $X^t = (X_1^t, \dots, X_p^t)'$ : realizations at time  $t$
- ▶ **VAR** model for NGC:

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

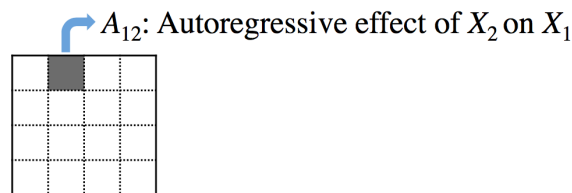
  $A_{11}$ : Autoregressive effect of  $X_1$  on itself



## Network Granger Causality with VARs

- ▶  $X_1, \dots, X_p$ : time series for  $p$  variables
- ▶  $X^t = (X_1^t, \dots, X_p^t)'$ : realizations at time  $t$
- ▶ **VAR** model for NGC:

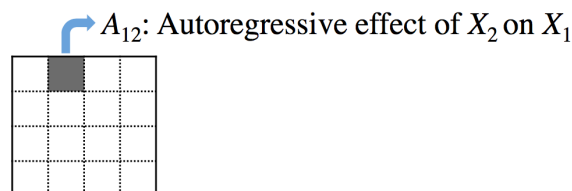
$$X^T = A^1 X^{T-1} + \dots + A^d X^{T-d} + \varepsilon^T$$



## Network Granger Causality with VARs

- ▶  $X_1, \dots, X_p$ : time series for  $p$  variables
- ▶  $X^t = (X_1^t, \dots, X_p^t)'$ : realizations at time  $t$
- ▶ **VAR** model for NGC:

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



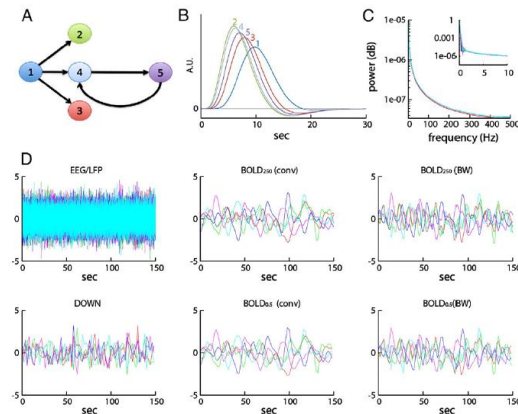
- ▶  $X_j$  **Granger-causal** for  $X_i$  if  $A_{i,j}^k \neq 0$  for some  $k$  ( $k = 1, \dots, d$ )





## Applications — Neuroscience

- ▶ Connectivity among brain regions from time-course fMRI data
- ▶ Connectivity of VAR generative model (Seth et al, 2013)



## Extensions

- ▶ Panel VAR Modeling (common in functional genomics and neuroscience)<sup>3</sup>
- ▶ Incorporating external information using group lasso penalties, etc<sup>4</sup>
- ▶ Dealing with non-stationarity (paucity of long stationary time series —  $T$  small)<sup>5</sup>
- ▶ Accounting for non-linearity
- ▶ ...

<sup>3</sup>S & Michailidis (2010); S, Basu & Michailidis (2012)

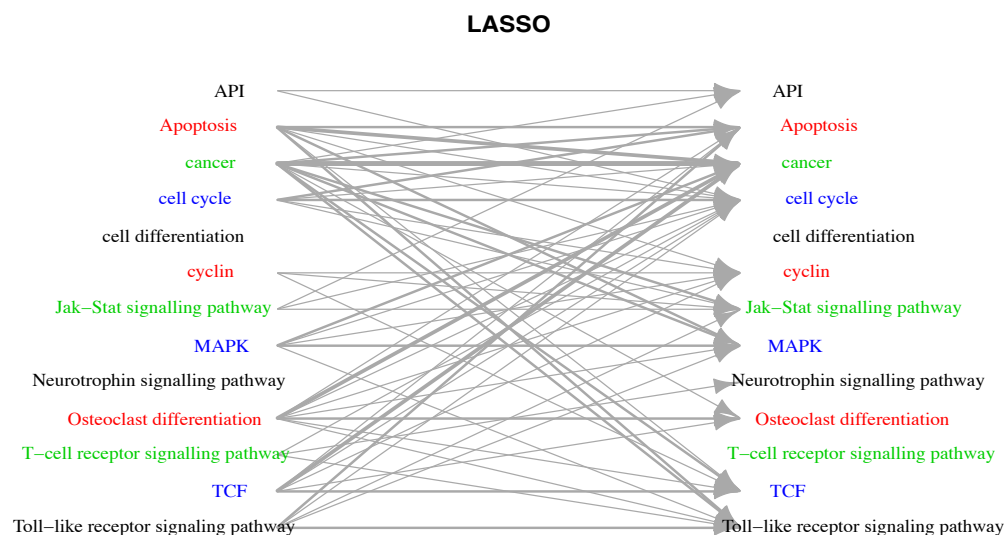
<sup>4</sup>Basu, S & Michailidis (2014)

<sup>5</sup>Safikhani & S (2020)

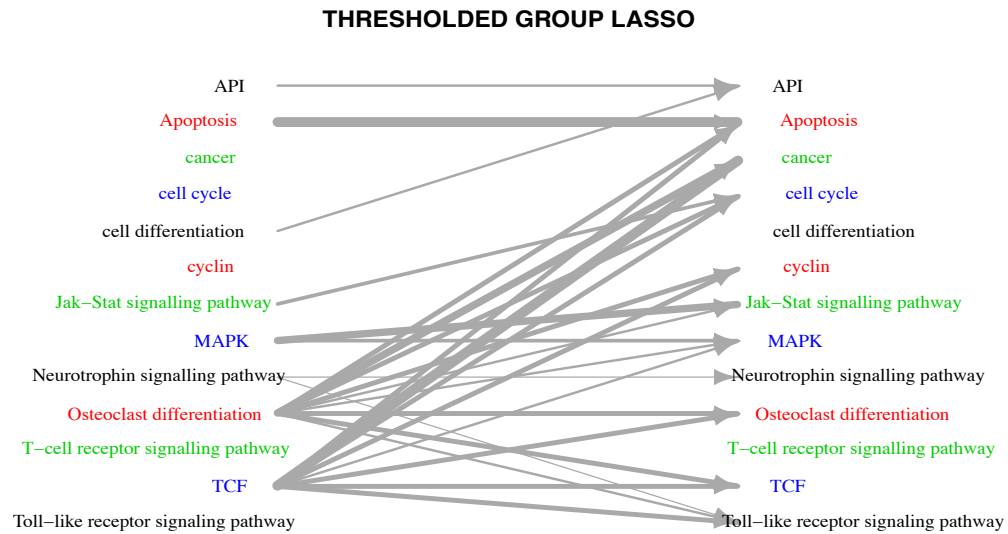
## Example: T-cell Activation Data

- ▶ Data from Rangel et al (2004) on T-cell activation — less insight and biological knowledge regarding pathways
- ▶  $p = 58$  genes,  $n = 44$  samples, and  $T = 10$  time points — the first 5 time points (0, 2, 4, 6 and 8 hours) were used on a subset of 38 genes for which pathway information avail
- ▶ Goal is to estimate regulatory interactions

## Estimated Network Structure



## Estimated Network Structure



## Summary

- Estimation of DAGs from observational data is both conceptually and computationally difficult
- Constraint-based & search-based algorithms — slow in high dim
- May not be able to distinguish DAGs from observational data (Markov equivalence)
- Efficient penalized likelihood methods can estimate DAGs **if the ordering is known**
- Important case is time series data, but **Granger causality  $\neq$  causality!**<sup>6</sup>
- Efficient implementations in R available for most methods

<sup>6</sup>S & Fox (2021)