Pathway & Network Analysis of Omics Data: Directed Graphical Models (Bayesian Networks)

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

Summer Institute for Statistical Genetics – Australia, 2017

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?

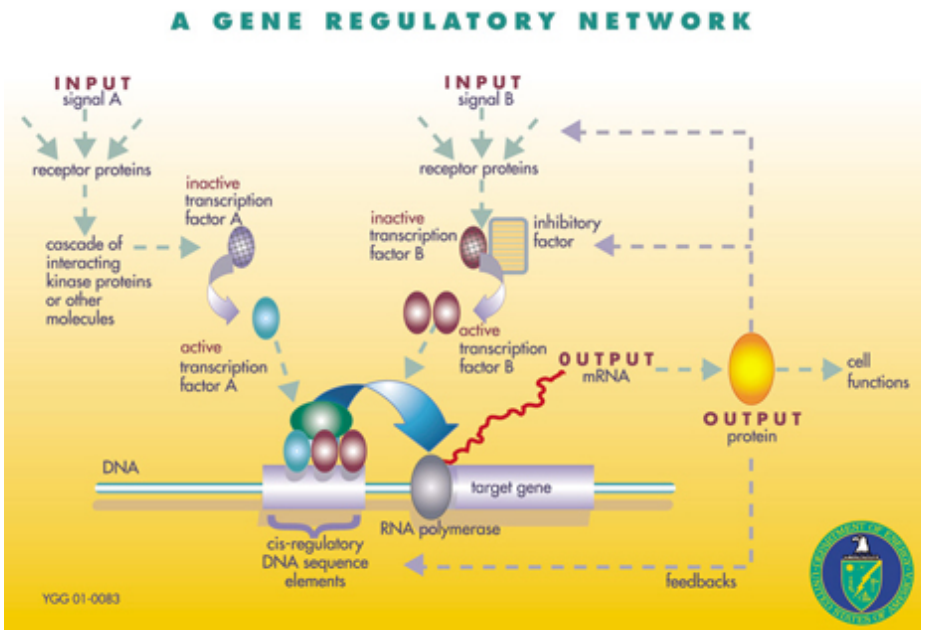
Why Bayesian Networks?

Many biological networks include directed edges:

Why Bayesian Networks?

Many biological networks include directed edges:

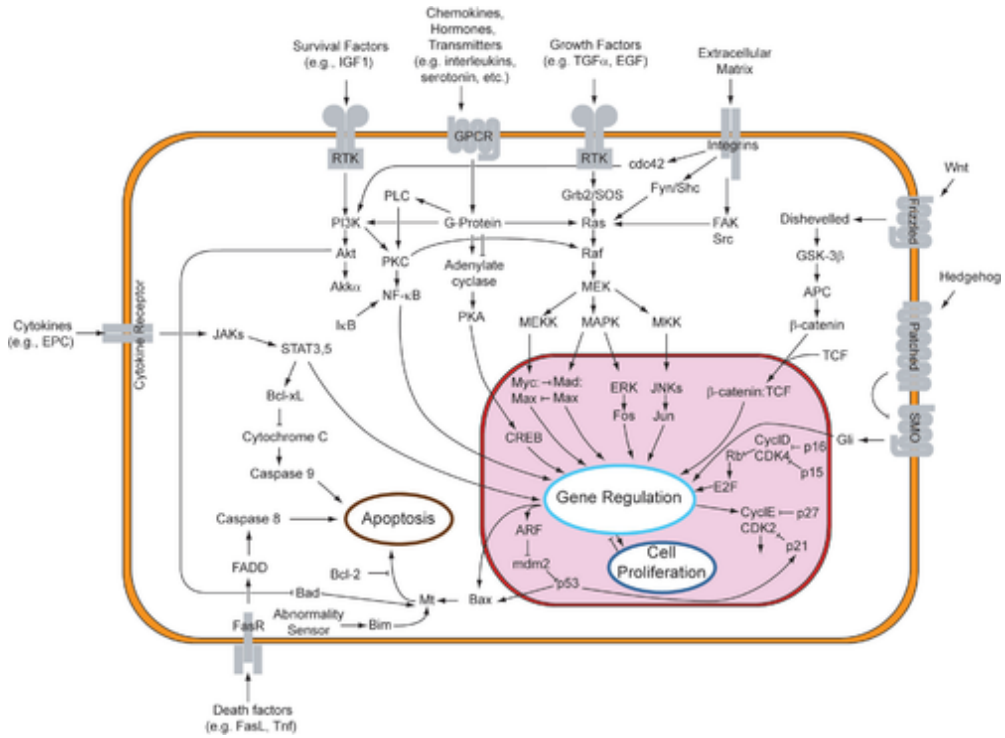
- In gene regulatory networks, protein products of transcription factors can alter the expression of target genes, but the target genes (usually) don't have a direct effect on the expression of transcription factors



Why Bayesian Networks?

Many biological networks include directed edges:

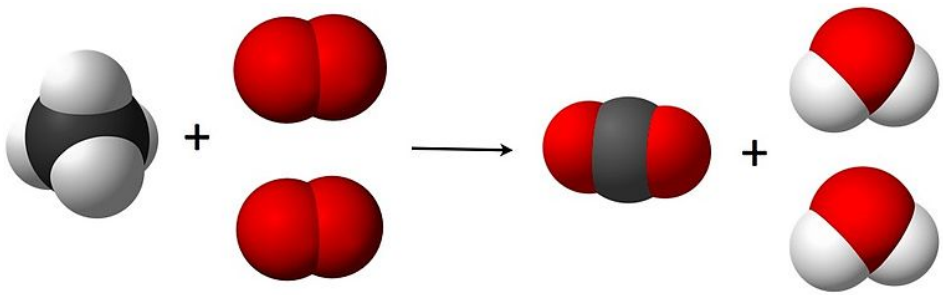
- In cell signaling networks, the signal from the cell's environment is transducted into the cell, and results in (global) changes in gene expression, but gene expression may not affect the environmental factors



Why Bayesian Networks?

Many biological networks include directed edges:

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



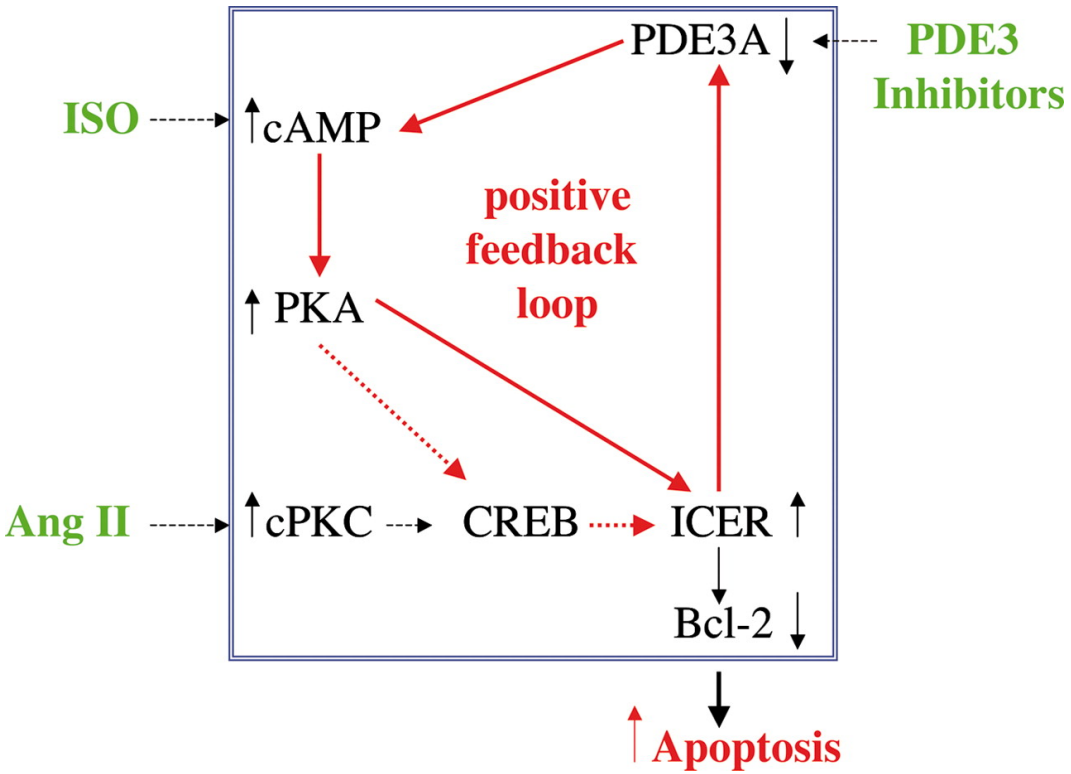
Why Bayesian Networks?

However, biological networks may not be DAGs:

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, of course, the edge directions; 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, of course, the edge directions; 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 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 $X_1 \to X_2$ and $X_2 \to X_1$, so basically what we see is:

Outline

Outline

- Basics of Bayesian networks, including

Outline

- Basics of Bayesian networks, including
  - directed acyclic graphs (DAGs)
  - conditional independence in DAGs, d-separation, etc
  - probability distributions over DAGs
  - structural equation models (SEM)

Outline

- Basics of Bayesian networks, including
  - directed acyclic graphs (DAGs)
  - conditional independence in DAGs, d-separation, etc
  - probability distributions over DAGs
  - structural equation models (SEM)
- Estimation of Bayesian networks from observational data

Directed Graphs: Some Terminology

- nodes in directed networks represent random variables; we denote the set of nodes by V
- edges are directed, and represent causal relationships among variables; we denote the set of edges by E

Directed Graphs: Some Terminology

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

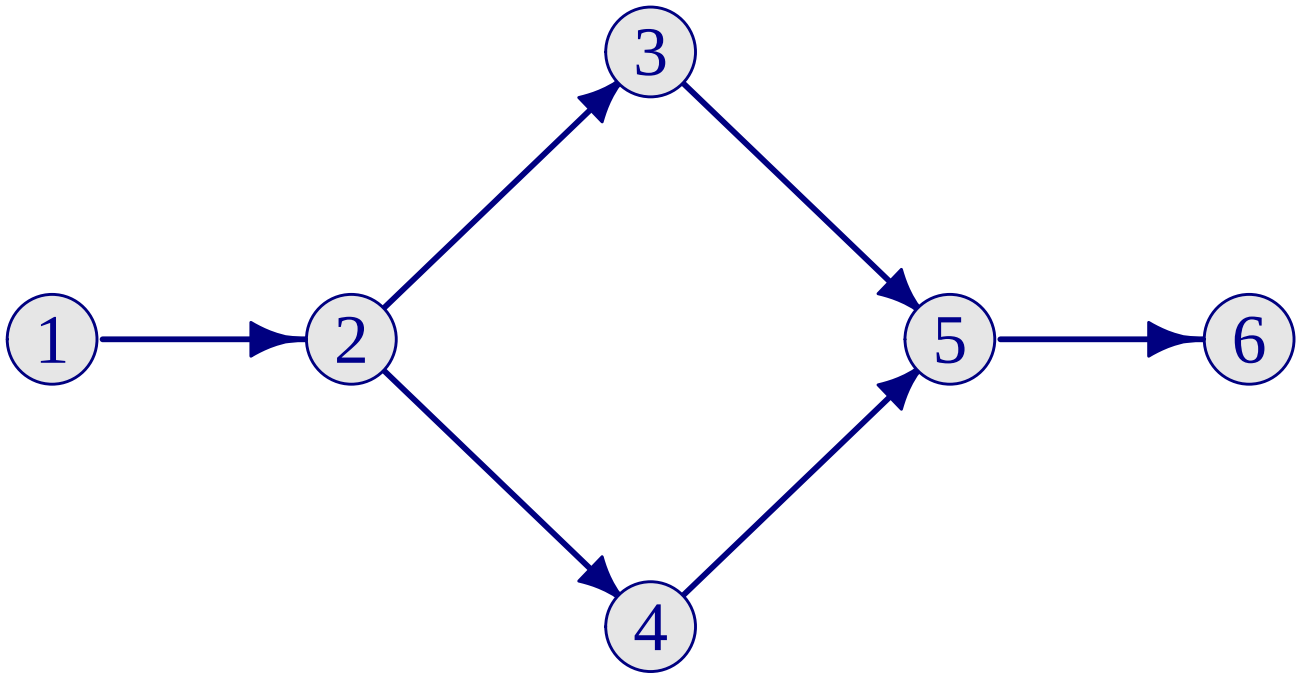
Directed Graphs: Some Terminology

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

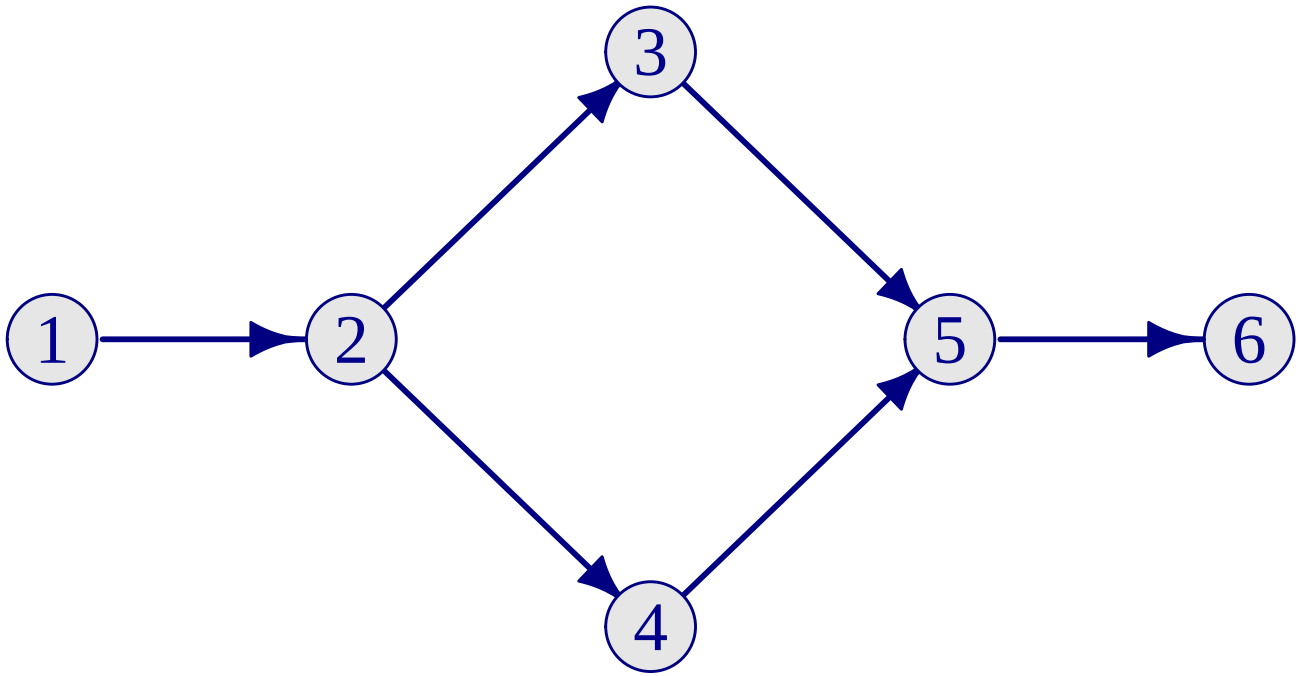
Directed Graphs: Some Terminology

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

Directed Graphs: Some Terminology

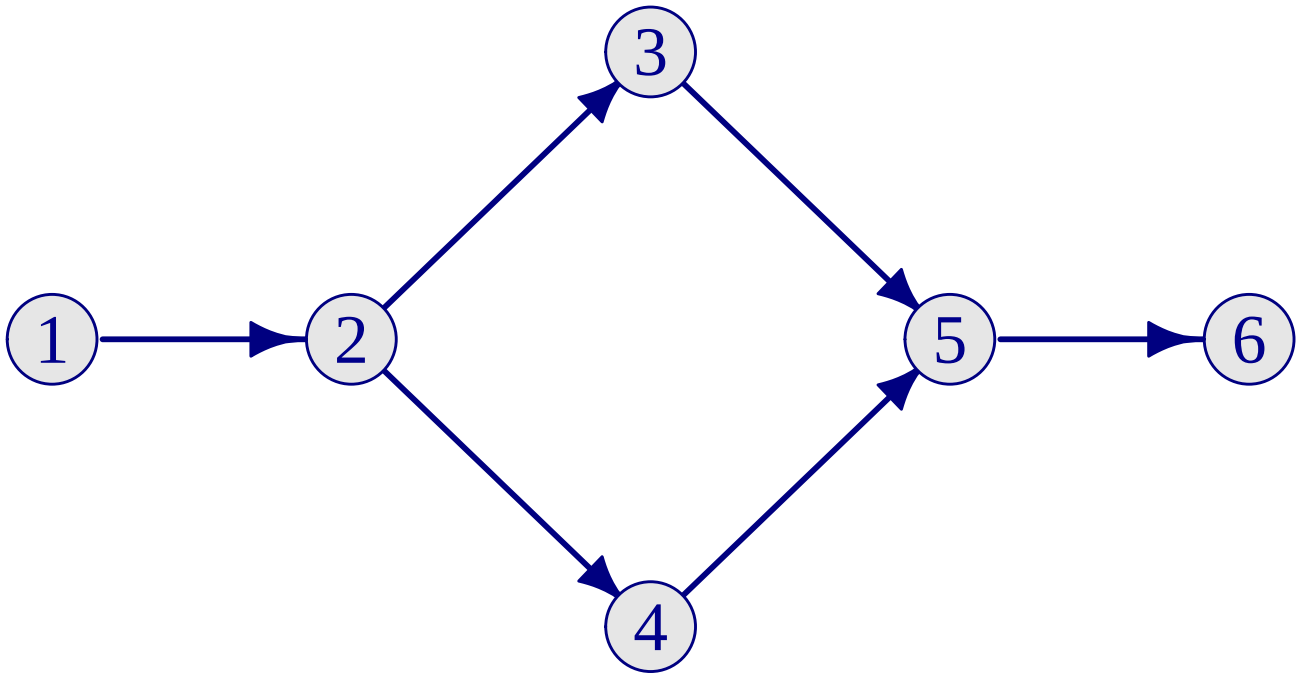


Directed Graphs: Some Terminology



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

Directed Graphs: Some Terminology



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

Directed Graphs: Some Terminology

Directed Graphs: Some Terminology

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

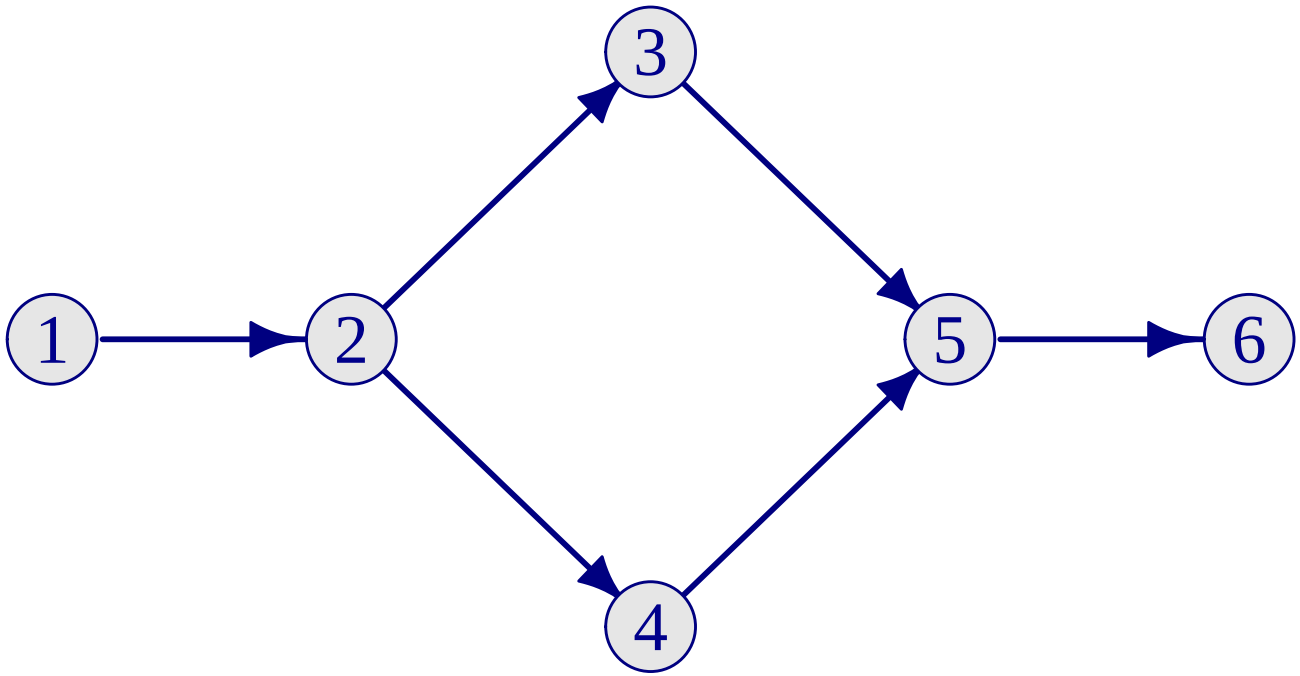
Directed Graphs: Some Terminology

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

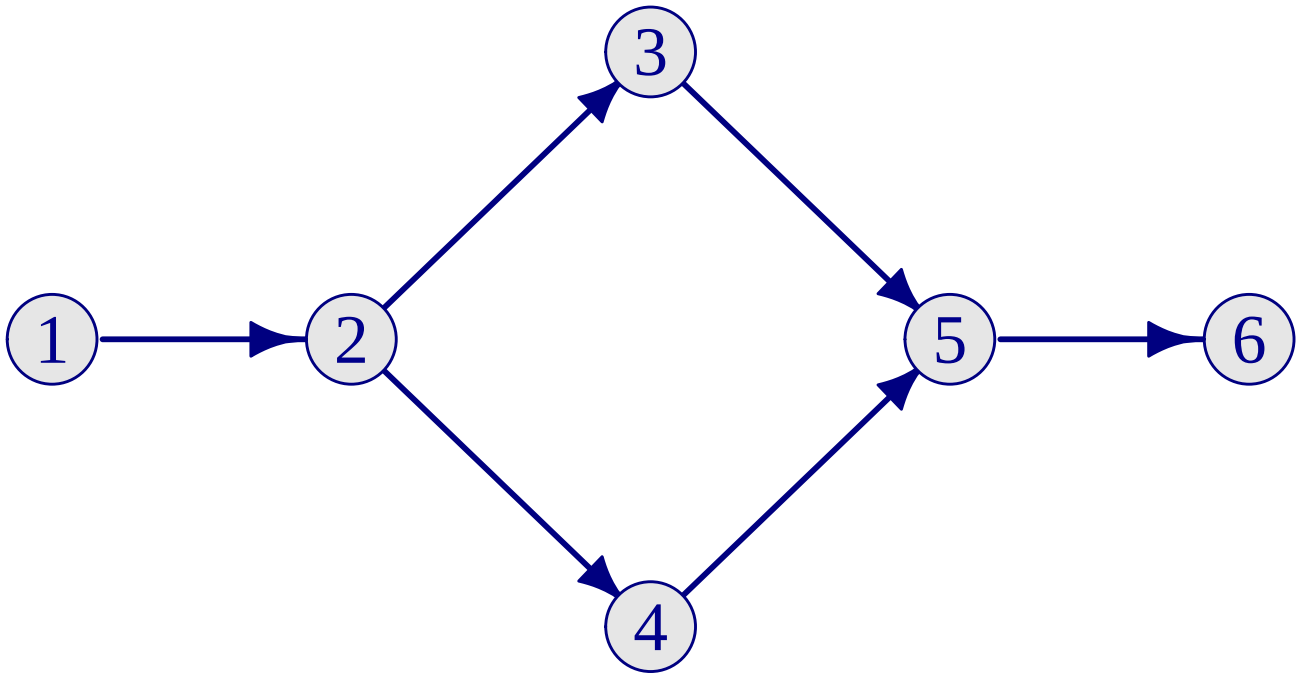
Directed Graphs: Some Terminology

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

Directed Graphs: Some Terminology

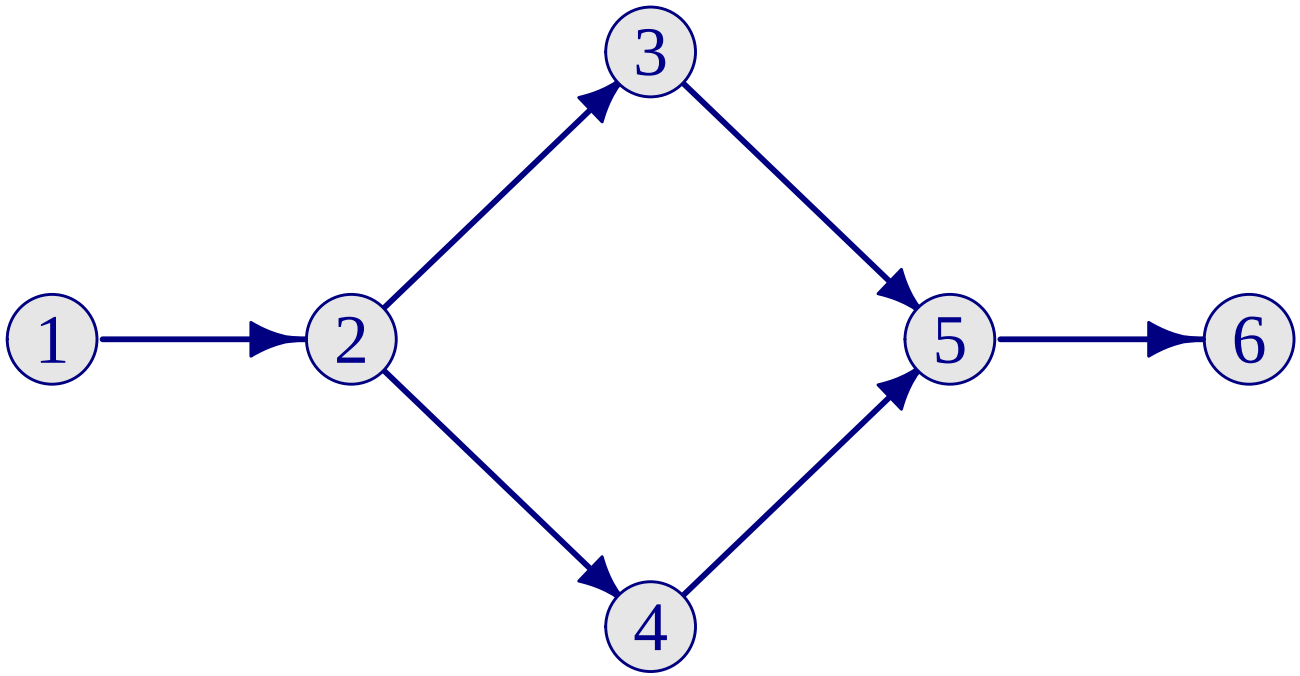


Directed Graphs: Some Terminology



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

Directed Graphs: Some Terminology



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

Directed Graphs: Some Terminology

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

Directed Graphs: Some Terminology

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

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

$$i \ldots \rightarrow k \leftarrow \ldots j$$

Directed Graphs: Some Terminology

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

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

$$i \ldots \rightarrow k \leftarrow \ldots j$$

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

Directed Graphs: Some Terminology

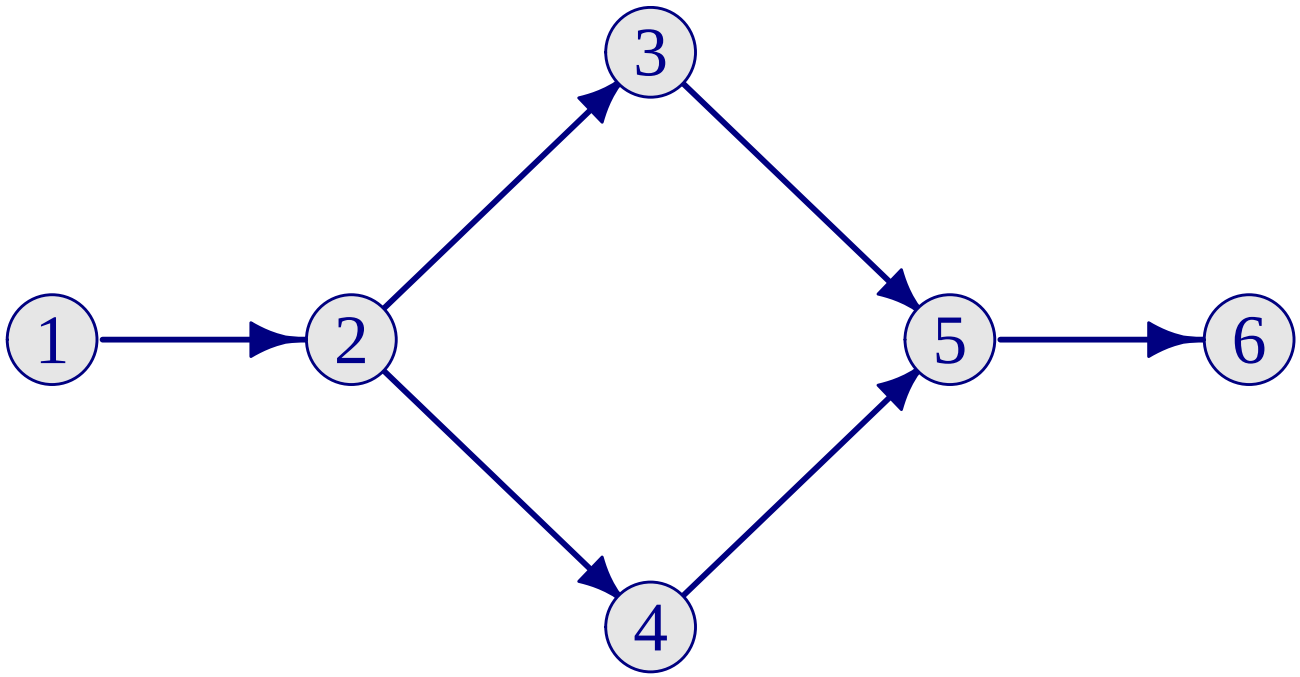
An important concept in DAGs is that of colliders (aka "inverted forks"):

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

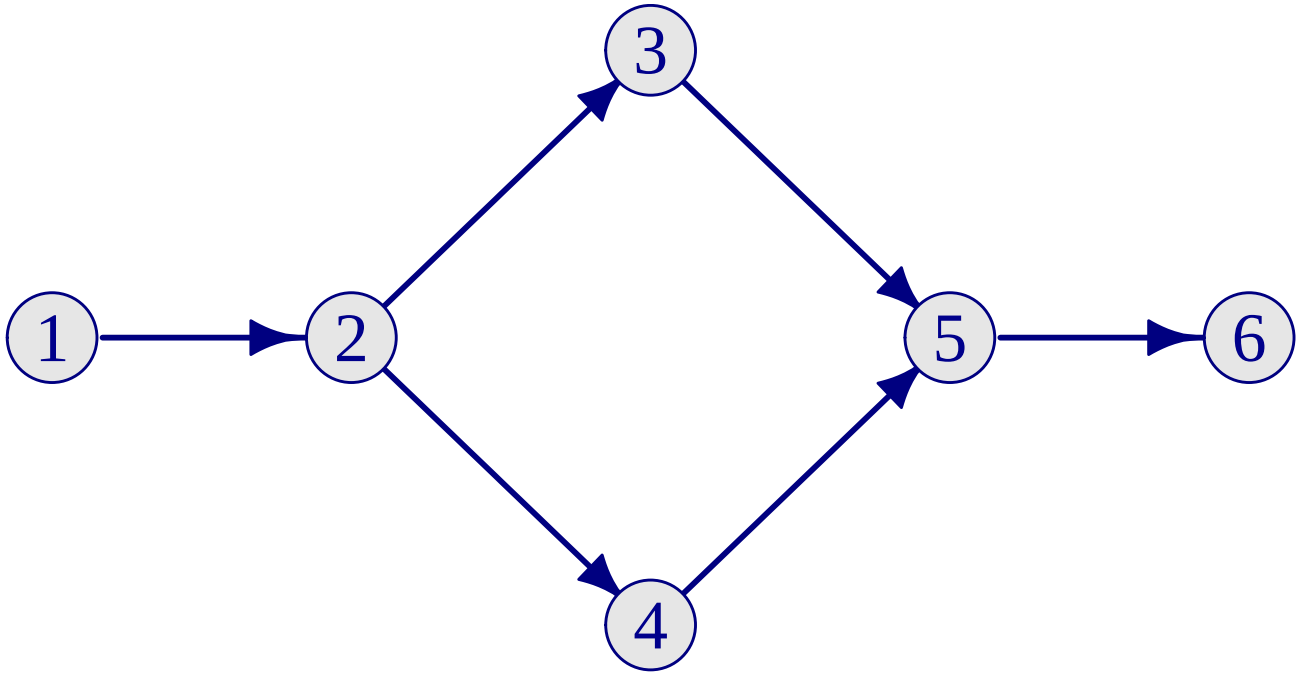
$$i \ldots \rightarrow k \leftarrow \ldots j$$

- k is an non-collider if it is not an end-point, and is not a collider on a path:
  - $i \ldots \leftarrow k \leftarrow \ldots j$
  - $i \ldots \rightarrow k \rightarrow \ldots j$
  - $i \ldots \leftarrow k \rightarrow \ldots j$
- Note: colliders and non-colliders are defined w.r.t. paths; a collider in one path can be a non-collider in another!

Directed Graphs: Some Terminology

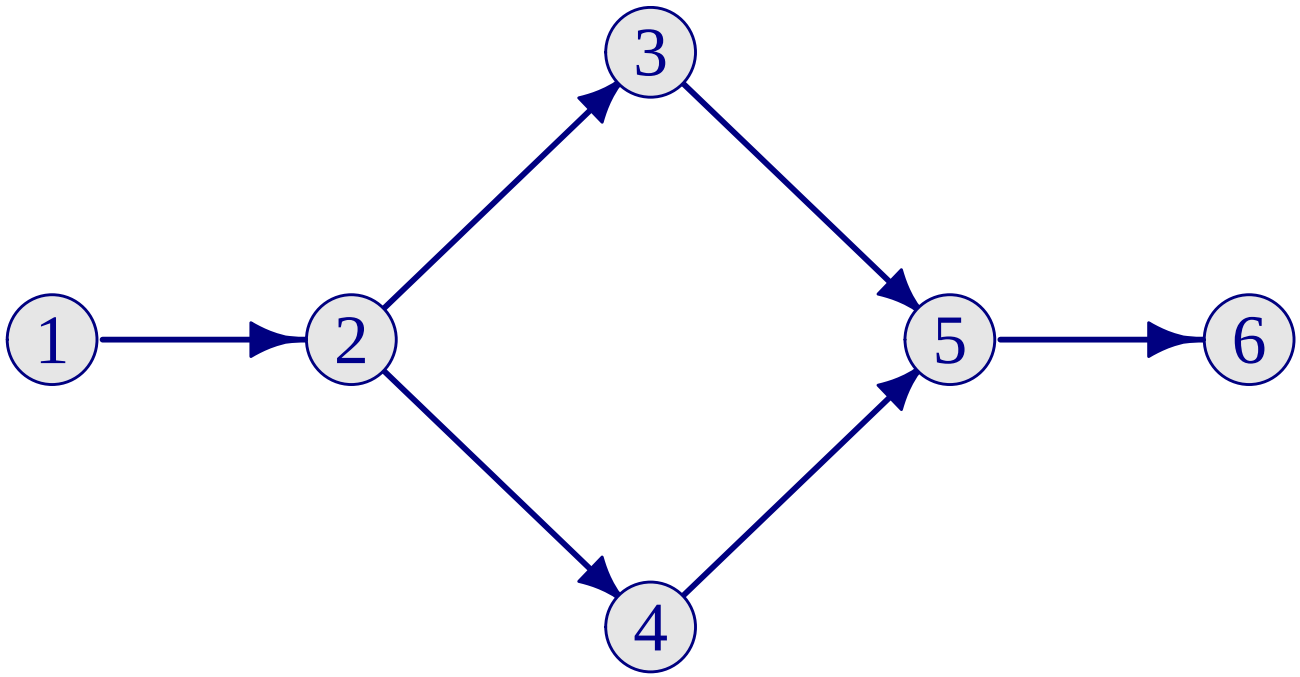


Directed Graphs: Some Terminology



- 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?

Factorization of Probability Distributions over DAGs

Factorization of Probability Distributions over DAGs

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

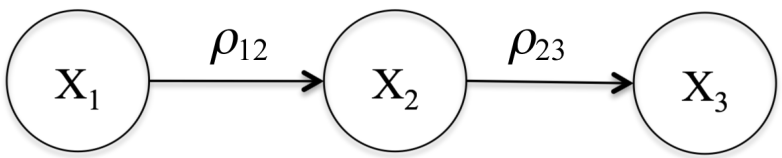
$$\begin{aligned} P(X_1, X_2, X_3) &= P(X_1 \mid X_2, X_3) P(X_2 | X_3) P(X_3) \\ &= P(X_3 \mid X_1, X_2) P(X_2 | X_1) P(X_1) \\ &= \dots \end{aligned}$$

Factorization of Probability Distributions over DAGs

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

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

- Now, consider this simple DAG

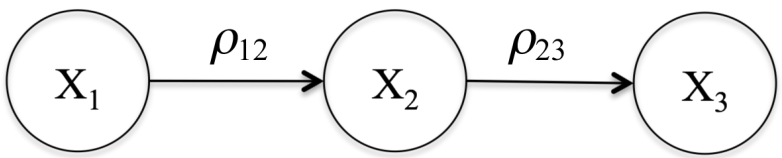


Factorization of Probability Distributions over DAGs

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

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

- Now, consider this simple DAG



- Then, the probability distribution can be factorized as

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

Factorization of Probability Distributions over DAGs

- In general, for random variables on a DAG $G = (V, E)$, and a compatible probability distribution P (ie Markov relative to G)

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

Factorization of Probability Distributions over DAGs

- In general, for random variables on a DAG $G = (V, E)$, and a compatible probability distribution P (ie Markov relative to G)

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

- Compare this with the general probability decomposition

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

Factorization of Probability Distributions over DAGs

- In general, for random variables on a DAG $G = (V, E)$, and a compatible probability distribution P (ie Markov relative to G)

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

- Compare this with the general probability decomposition

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

- This means that for DAGs we have

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

Factorization of Probability Distributions over DAGs

- In general, for random variables on a DAG $G = (V, E)$, and a compatible probability distribution P (ie Markov relative to G)

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

- Compare this with the general probability decomposition

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

- This means that for DAGs we have

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

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

Independence (unconditional)

Independence (unconditional)

- Two random variables are independent, $X \perp\!\!\!\perp Y$ if knowledge of X provides no information about Y.

Independence (unconditional)

- Two random variables are independent, $X \perp\!\!\!\perp Y$ if knowledge of X provides no information about Y.
- The following are equivalent characterizations of independence, $X \perp\!\!\!\perp Y$:

Independence (unconditional)

- Two random variables are independent, $X \perp\!\!\!\perp Y$ if knowledge of X provides no information about Y.
- The following are equivalent characterizations of independence, $X \perp\!\!\!\perp Y$:
  - $P(X = x, Y = y) = P(X = x)P(Y = y)$
  - $P(X = x|Y = y) = P(X = x)$ (is symmetric)

Independence (unconditional)

- Two random variables are independent, $X \perp\!\!\!\perp Y$ if knowledge of X provides no information about Y.
- The following are equivalent characterizations of independence, $X \perp\!\!\!\perp Y$:
  - $P(X = x, Y = y) = P(X = x)P(Y = y)$
  - $P(X = x|Y = y) = P(X = x)$ (is symmetric)
- These can be generalized for vectors.

Independence (unconditional)

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

Independence (unconditional)

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

Conditional Independence

Conditional Independence

- Two variables X and Y are conditionally independent given a third variable Z (written $X \perp\!\!\!\perp Y | Z$) if given Z the knowledge of X provides no information about Y.

Conditional Independence

- Two variables X and Y are conditionally independent given a third variable Z (written $X \perp\!\!\!\perp Y | Z$) if given Z the knowledge of X provides no information about Y.
- Conditional independence $X \perp\!\!\!\perp Y \mid Z$ means:

Conditional Independence

- Two variables X and Y are conditionally independent given a third variable Z (written $X \perp\!\!\!\perp Y|Z$) if given Z the knowledge of X provides no information about Y.
- Conditional independence $X \perp\!\!\!\perp Y \mid Z$ means:
  i) $P(X = x, Y = y|Z = z) = P(X = x|Z = z)P(Y = y|Z = z)$
  ii) $P(X = x|Y = y, Z = z) = P(X = x|Z = z)$ (is symmetric)

Conditional Independence

- Two variables X and Y are conditionally independent given a third variable Z (written $X \perp\!\!\!\perp Y|Z$) if given Z the knowledge of X provides no information about Y.
- Conditional independence $X \perp\!\!\!\perp Y \mid Z$ means:
  i) $P(X = x, Y = y|Z = z) = P(X = x|Z = z)P(Y = y|Z = z)$
  ii) $P(X = x|Y = y, Z = z) = P(X = x|Z = z)$ (is symmetric)
- We also have,

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

Conditional Independence

- Two variables X and Y are conditionally independent given a third variable Z (written $X \perp\!\!\!\perp Y|Z$) if given Z the knowledge of X provides no information about Y.
- Conditional independence $X \perp\!\!\!\perp Y \mid Z$ means:
  i) $P(X = x, Y = y|Z = z) = P(X = x|Z = z)P(Y = y|Z = z)$
  ii) $P(X = x|Y = y, Z = z) = P(X = x|Z = z)$ (is symmetric)
- We also have,

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

- These can also be generalized for vectors.

Conditional Independence

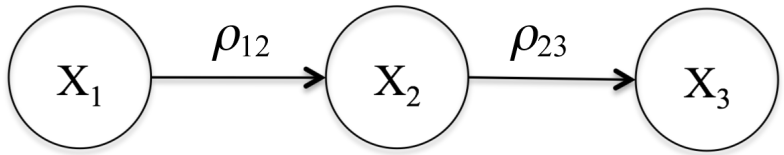
Conditional Independence

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

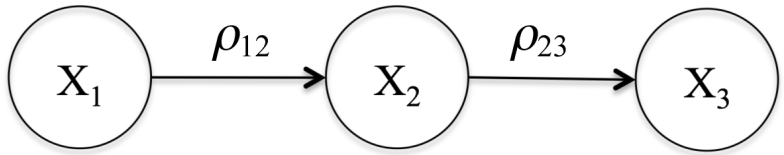
Conditional Independence

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

The Toy Example, Revisited

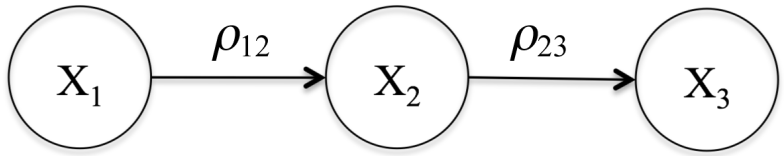


The Toy Example, Revisited



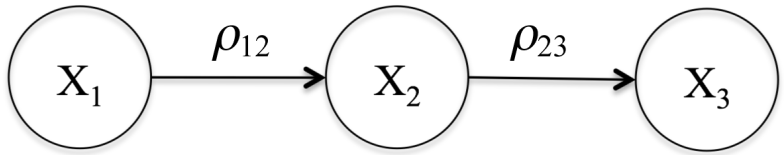
- Recall that $P(X_1, X_2, X_3) = P(X_3|X_2)P(X_2|X_1)P(X_1)$

The Toy Example, Revisited



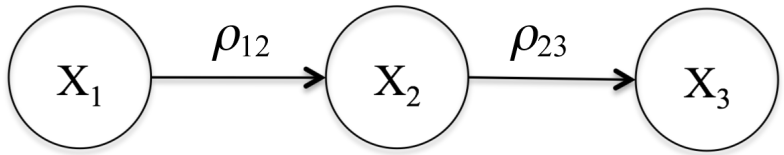
- Recall that $P(X_1, X_2, X_3) = P(X_3|X_2)P(X_2|X_1)P(X_1)$
- This implies that $X_3 \perp\!\!\!\perp X_1|X_2$ (by (i))

The Toy Example, Revisited



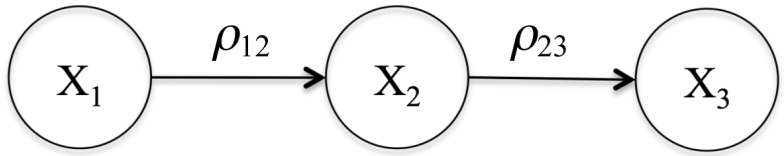
- Recall that $P(X_1, X_2, X_3) = P(X_3|X_2)P(X_2|X_1)P(X_1)$
- This implies that $X_3 \perp\!\!\!\perp X_1|X_2$ (by (i))
- However, this is not always the case on DAGs!

The Toy Example, Revisited



- Recall that $P(X_1, X_2, X_3) = P(X_3|X_2)P(X_2|X_1)P(X_1)$
- This implies that $X_3 \perp\!\!\!\perp X_1|X_2$ (by (i))
- However, this is not always the case on DAGs!
- How to read conditional independences from the DAG?

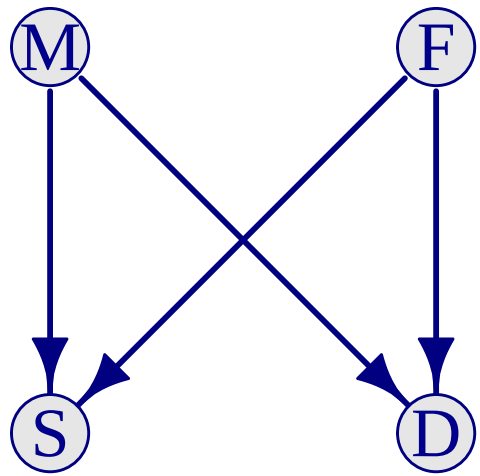
The Toy Example, Revisited



- Recall that $P(X_1, X_2, X_3) = P(X_3|X_2)P(X_2|X_1)P(X_1)$
- This implies that $X_3 \perp\!\!\!\perp X_1|X_2$ (by (i))
- However, this is not always the case on DAGs!
- How to read conditional independences from the DAG?
- We can do this using a concept called d-separation?

An example from genetics

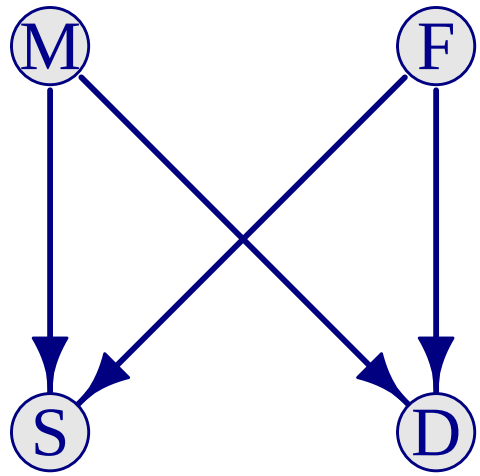
Consider an example from population genetics:



- Genetic information for *M*other, *F*ather, *D*aughter and *S*on in form of dominant/recessive genotype (A/a) for a single gene
- Then each individual can have one of three states: AA, aa, Aa

An example from genetics

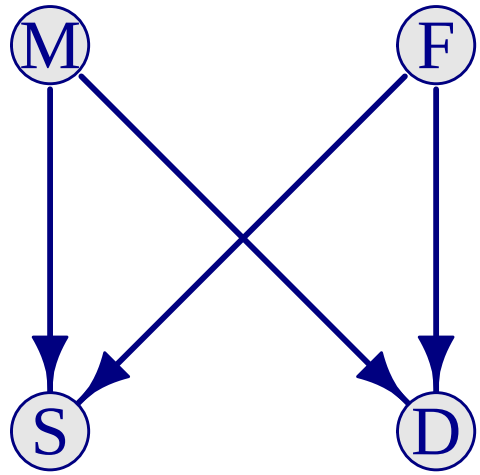
Consider an example from population genetics:



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

An example from genetics

Consider an example from population genetics:



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

d-separation

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

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

d-separation

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

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

How is this used?

d-separation

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

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

How is this used?

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

d-separation

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

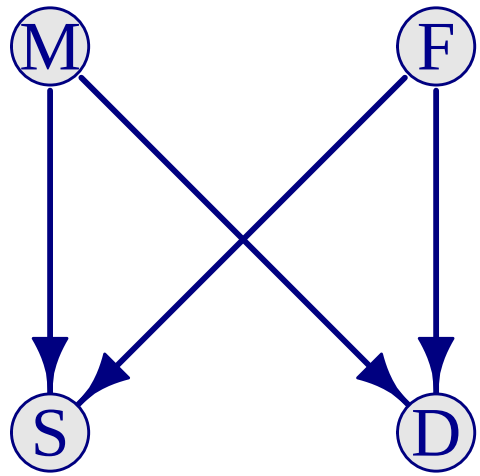
1. π includes a chain $i \rightarrow m \rightarrow j$ or a fork $i \leftarrow m \rightarrow j$ such that the middle note is in Z, or
2. π contains a collider (or inverted fork) $i \rightarrow m \leftarrow j$ such that neither the middle node m nor its descendants are NOT in Z.

How is this used?

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

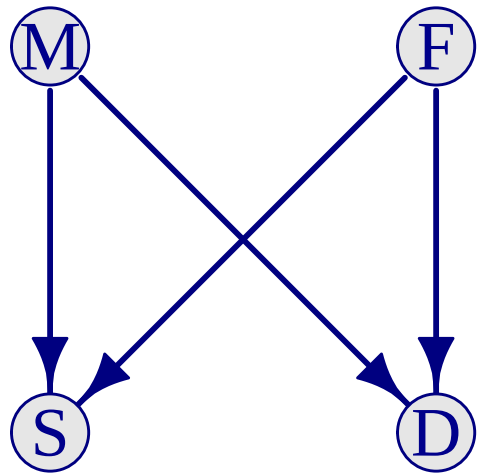
Genetics example, revisited

Consider an example from population genetics:



Genetics example, revisited

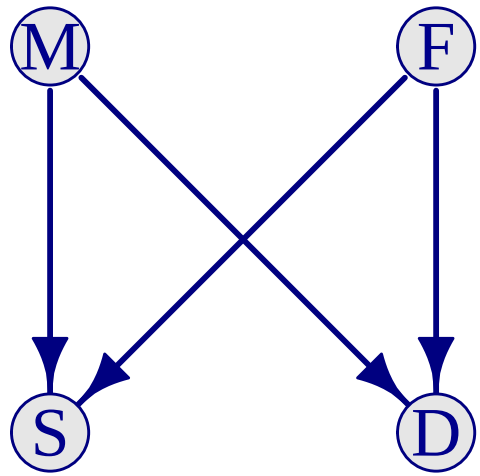
Consider an example from population genetics:



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

Genetics example, revisited

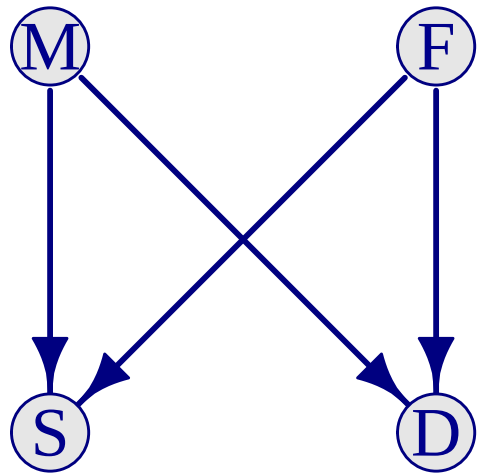
Consider an example from population genetics:



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

Genetics example, revisited

Consider an example from population genetics:

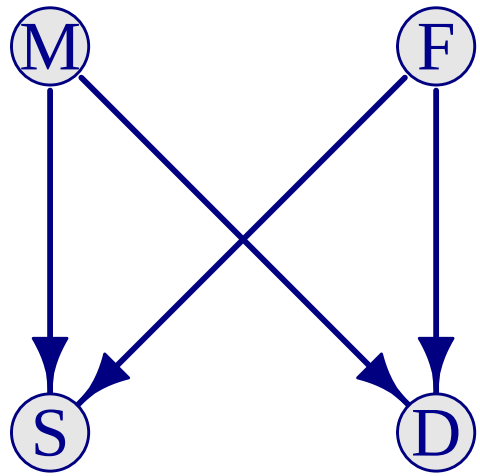


- $\{M, F\}$ block all paths from S to $D \Rightarrow D \perp\!\!\!\perp S \mid \{M, F\}$
- Is $M \perp\!\!\!\perp F$?
- Is $M \perp\!\!\!\perp F \mid \{S, D\}$, $\mid S$, $\mid D$?

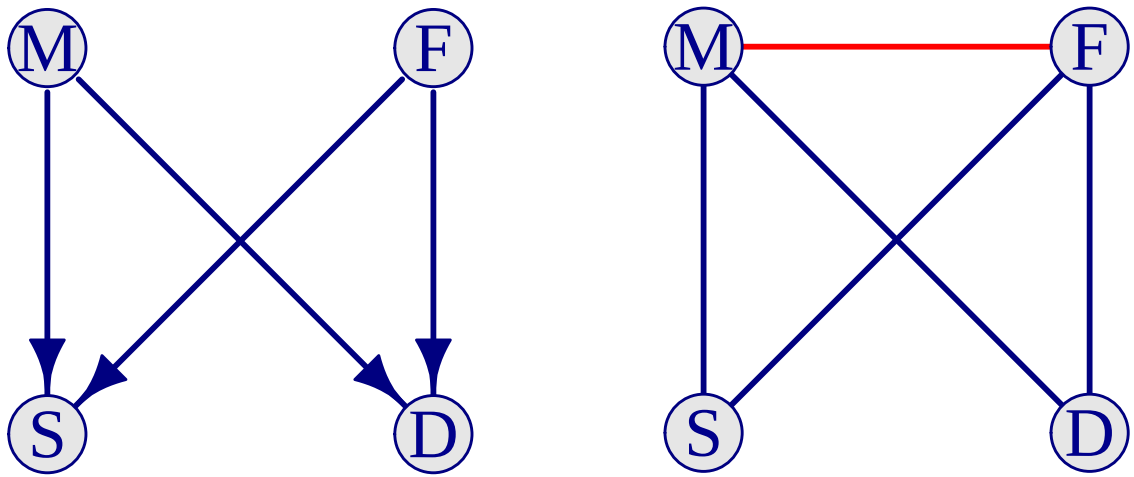
Moral Graphs

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

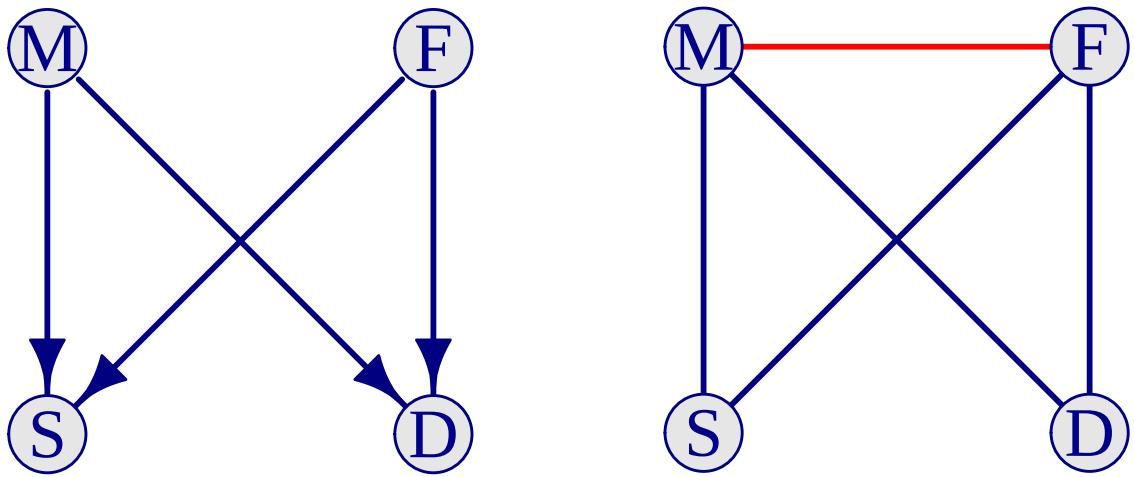
Genetics example, revisited (again)



Genetics example, revisited (again)



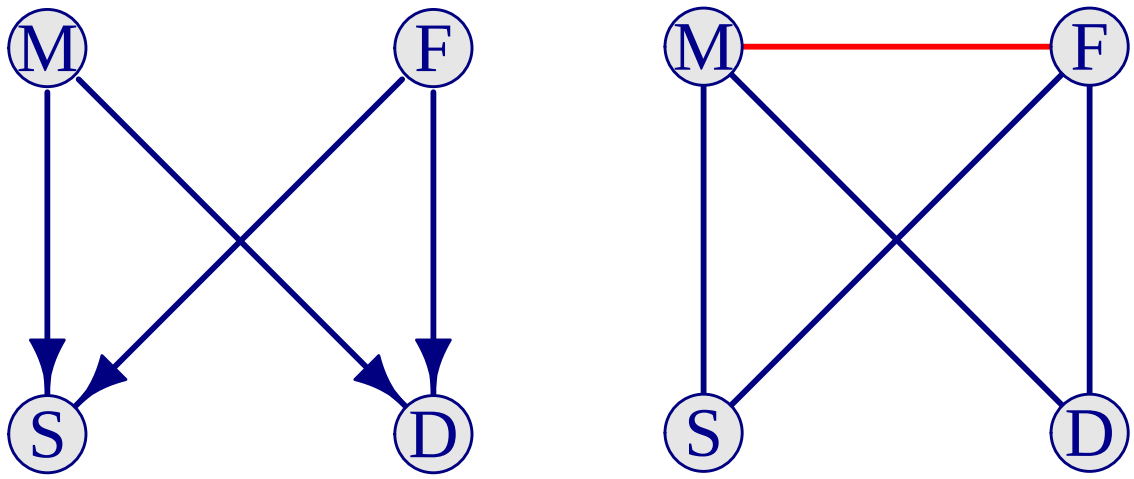
Genetics example, revisited (again)



The moral graph allows us to answer the following questions:

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

Genetics example, revisited (again)



The moral graph allows us to answer the following questions:

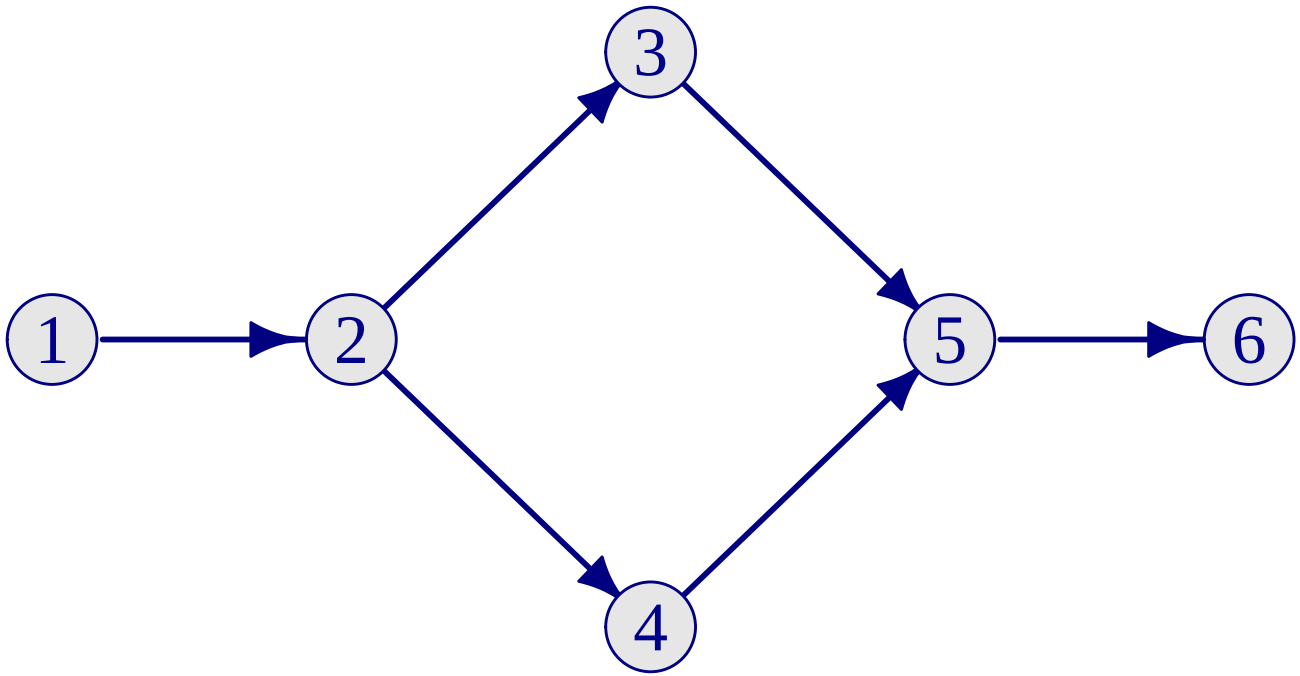
- Is $S \perp\!\!\!\perp D \mid \{M, F\}$
- Is $M \perp\!\!\!\perp F \mid \{S, D\}$, $\mid S$, $\mid D$?

But it does not answer questions like:

- Is $M \perp\!\!\!\perp F$?
- Is $M \perp\!\!\!\perp F \mid S$ or $\mid D$?

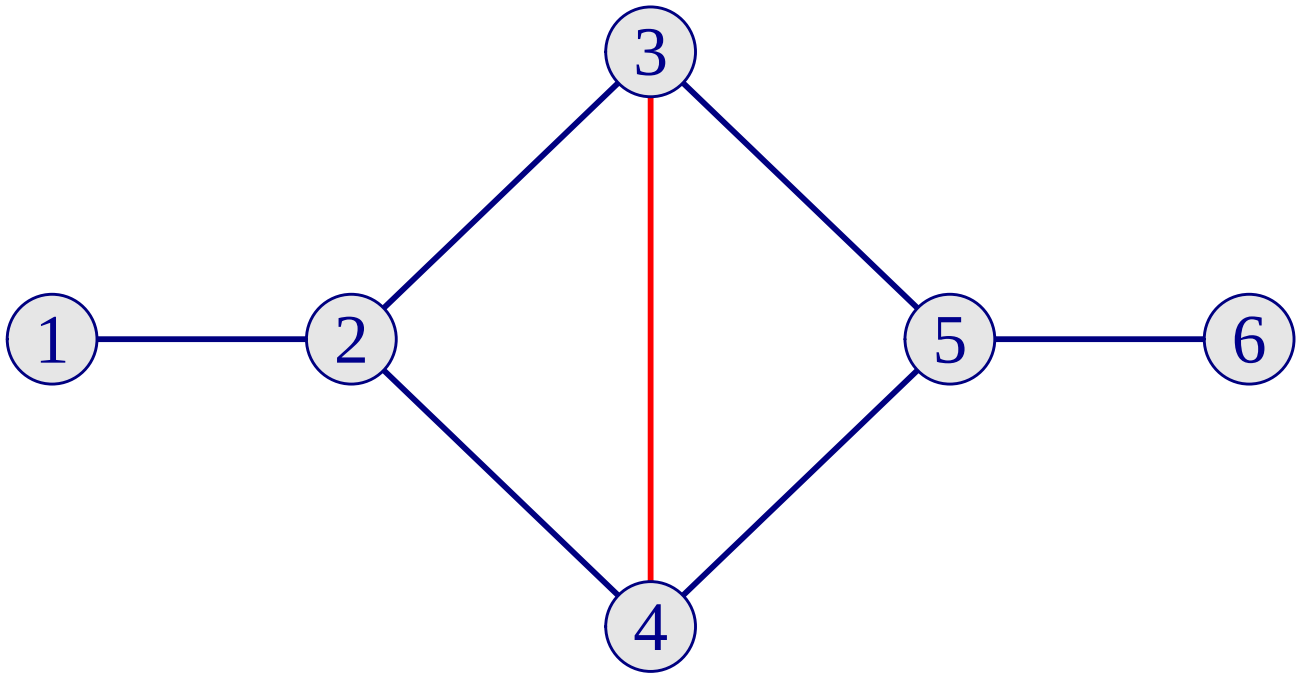
A More Complex Example

What nodes are conditionally independent given all other nodes?



A More Complex Example

What nodes are conditionally independent given all other nodes?



Structural Equation Models

Structural Equation Models

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

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

Structural Equation Models

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

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

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

Structural Equation Models

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

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

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

Structural Equation Models

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

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

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

- For Gaussian random variables, f_i is linear

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

Structural Equation Models

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

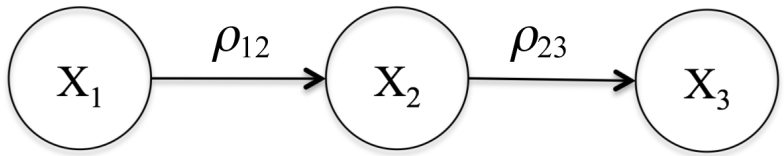
$$X_j = f_j(pa_j, \gamma_j), \quad j = 1, \ldots, p$$

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

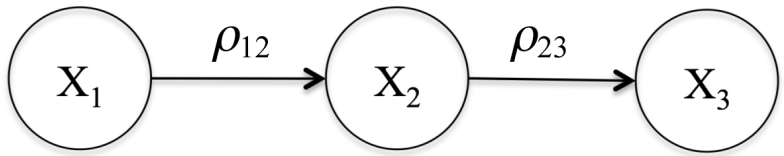
$$X_j = \sum_{j' \in pa_j} \rho_{jj'} X_{j'} + \gamma_j, \quad j = 1, \ldots, p$$

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

A Toy Example



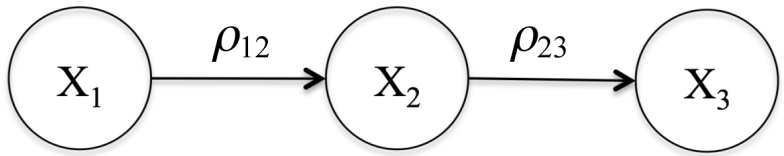
A Toy Example



Assuming normality we can write:

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

A Toy Example



Assuming normality we can write:

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

For non-Gaussian variables, equations involve non-linear functions.

Estimation of DAGs in Biological Settings

- Estimation of DAGs is (in general) computationally very hard (in fact, it's NP-hard): there are $\sim 2^{p^2}$ DAGs with p nodes!
- Three different types of biological data can be used for estimation of directed graphs:
  - i) observational data: steady-state data, or data comparing normal & cancer cells
  - ii) time-course data: time-course gene expression data
  - iii) perturbation data: data from knockouts experiments
- We will only cover (i), but note that (ii) and (iii) provide more informative data.

Estimation of DAGs from Observational Data

Broadly, two (three?) groups of algorithms for estimation of DAGs:

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

Constraint-Based Methods

Constraint-Based Methods

Idea: for each node pair j, j', is there a set S s.t. $X_j \perp\!\!\!\perp X_{j'} \mid S$?

Constraint-Based Methods

Idea: for each node pair j, j', is there a set S s.t. $X_j \perp\!\!\!\perp X_{j'} \mid S$?

- S can have 0 to $p-2$ members!
- For each pair of variables (all $p(p-1)/2$ of them), we need to look at all possible subsets of remaining variables!!
- usually stop at some $k \ll p$

Constraint-Based Methods

Constraint-Based Methods

Idea: for each node pair j, j', is there a set S s.t. $X_j \perp\!\!\!\perp X_{j'} \mid S$?

Constraint-Based Methods

Idea: for each node pair j, j', is there a set S s.t. $X_j \perp\!\!\!\perp X_{j'} \mid S$?

Need a conditional independence test (to test if $X \perp\!\!\!\perp Y \mid S$)

- For Gaussian data, we can use partial correlation
- For Binary data, we can use logOR
- In general, we can use conditional mutual information

Constraint-Based Methods

Idea: for each node pair j, j', is there a set S s.t. $X_j \perp\!\!\!\perp X_{j'} \mid S$?

Need a conditional independence test (to test if $X \perp\!\!\!\perp Y \mid S$)

- For Gaussian data, we can use partial correlation
- For Binary data, we can use logOR
- In general, we can use conditional mutual information

Some notes:

- Conditional independence is symmetric $\Rightarrow$ undirected graph!!
- These methods find the structure/skeleton of the DAG (we'll talk about direction later)

PC Algorithm (Spirtes et al, 1993)

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

PC Algorithm (Spirtes et al, 1993)

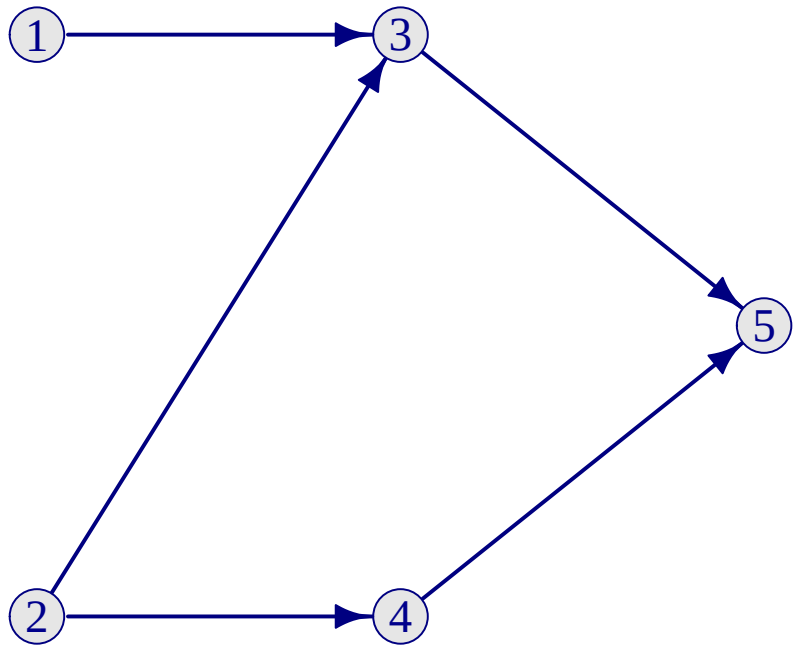
Start with a complete undirected graph, and set $i = 0$

Repeat

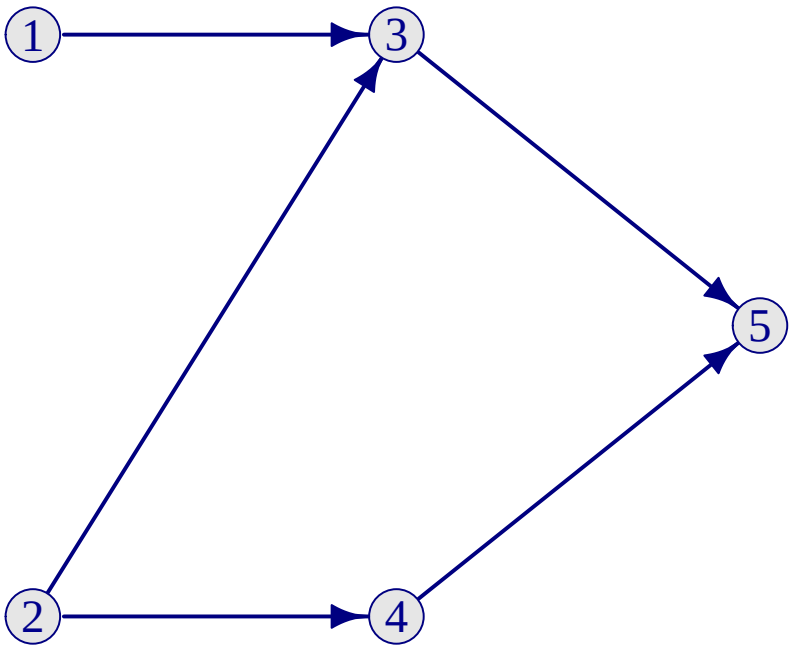
- For each $j \in V$
- For each $j' \in \mathrm{ne}(j)$
- Determine if $\exists S \subset \mathrm{ne}(j) \backslash \{j'\}$ with $|S| = i$
  - Test for CI: is $X_j \perp\!\!\!\perp X_{j'} \mid S$?
  - If such an S exists, then set $S_{jj'} = S$, remove $j - j'$ edge
- $i = i + 1$

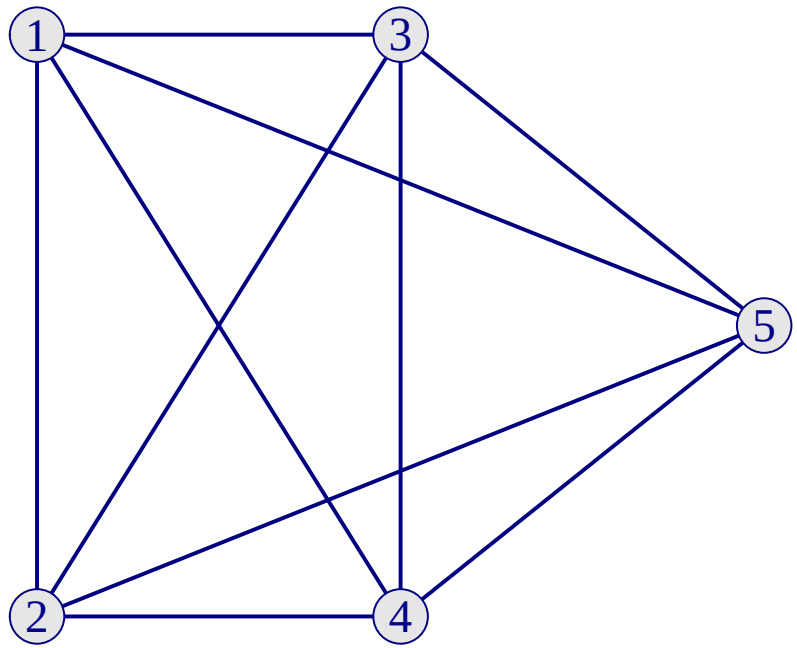
Until $|\mathrm{ne}(j)| < i$ for all j

Example

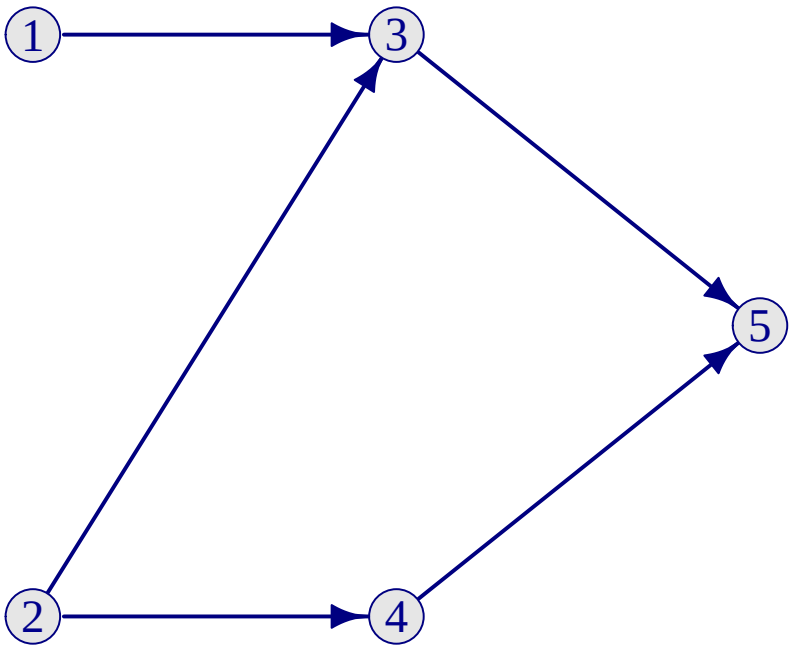


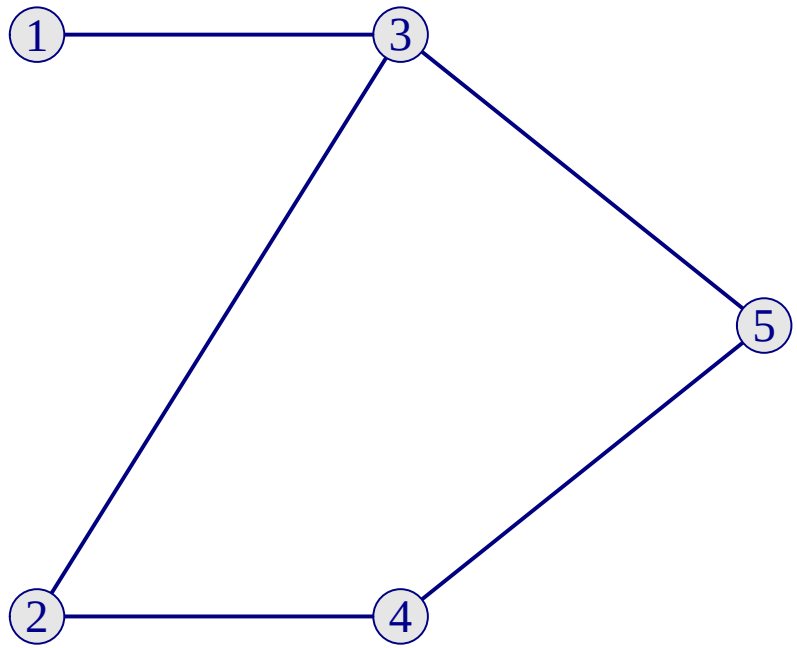
Example



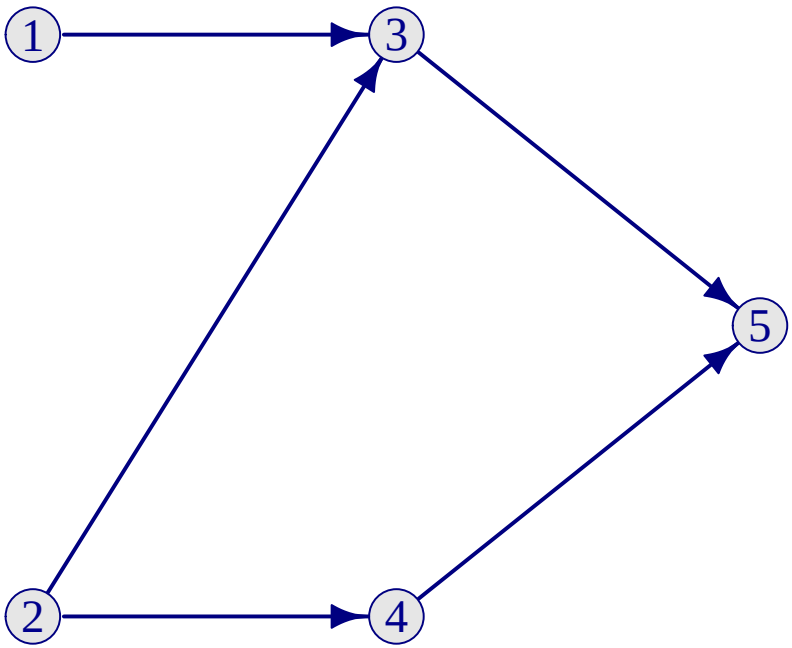


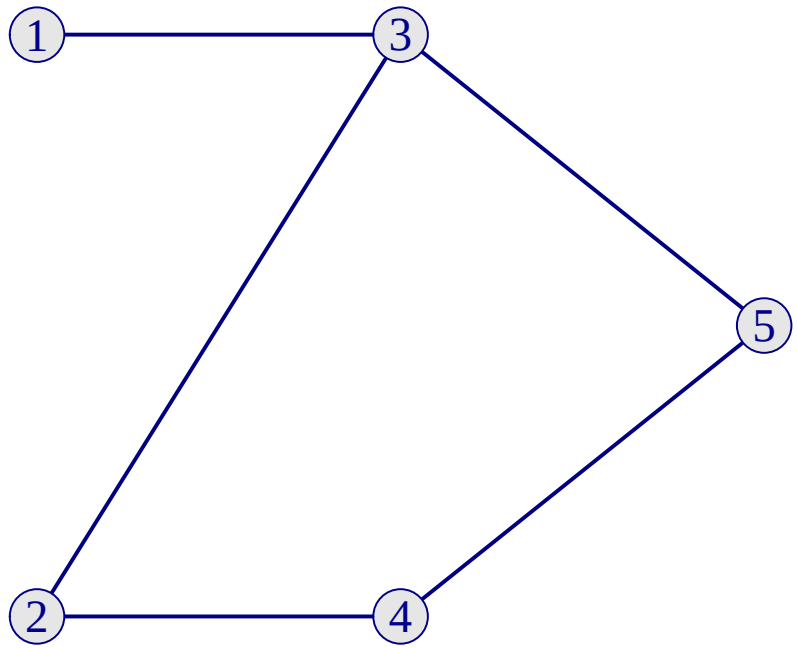
Example





Example





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

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

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

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

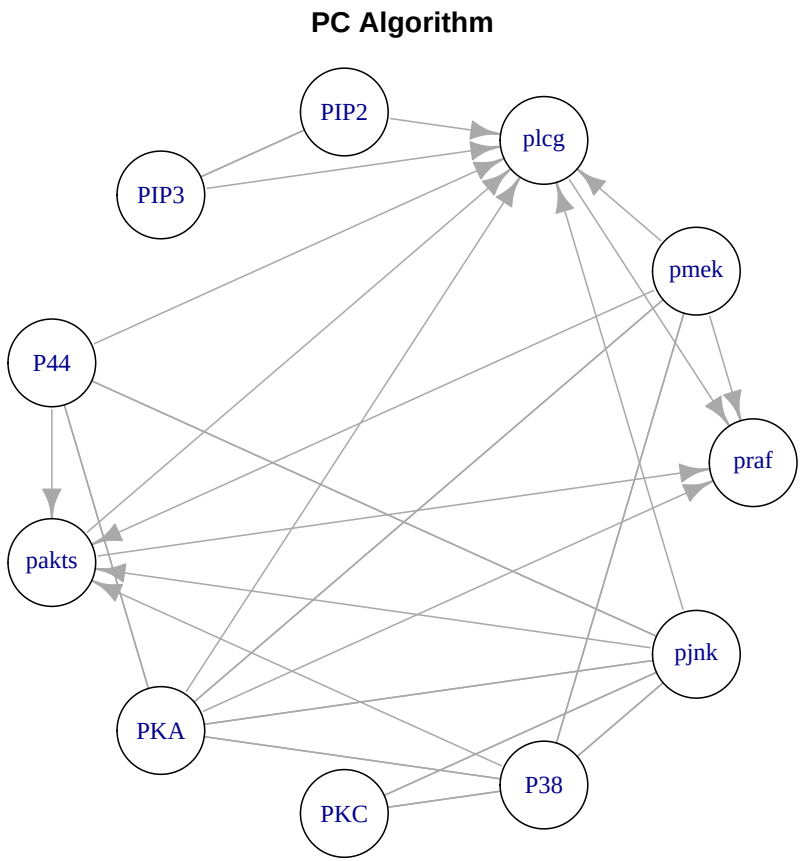
$\quad\quad\quad\quad S_{2,5} = \{3, 4\}$

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

Analysis of Protein Flow Cytometry using pcalg

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

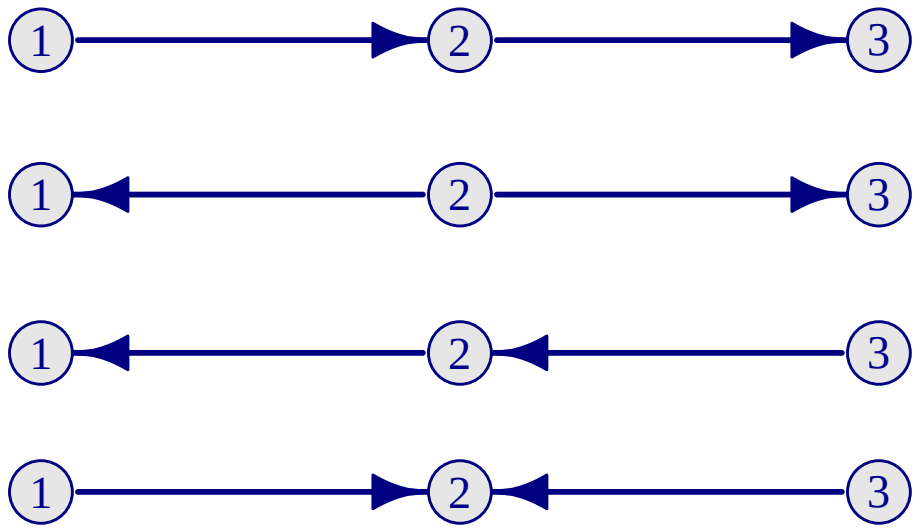
Analysis of Protein Flow Cytometry using `pcalg`



Q: Where did the directions come from? And why are only some of the edges directed?

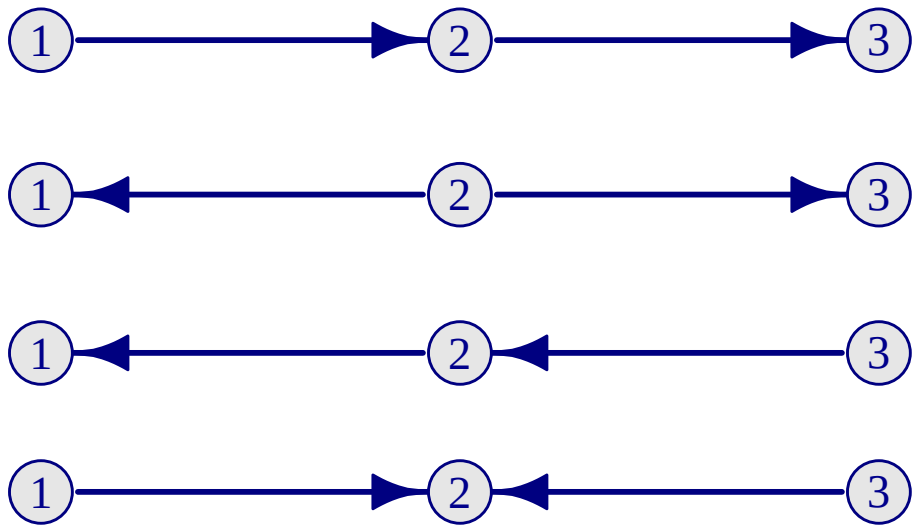
Markov Equivalence

Consider the following 4 graphs



Markov Equivalence

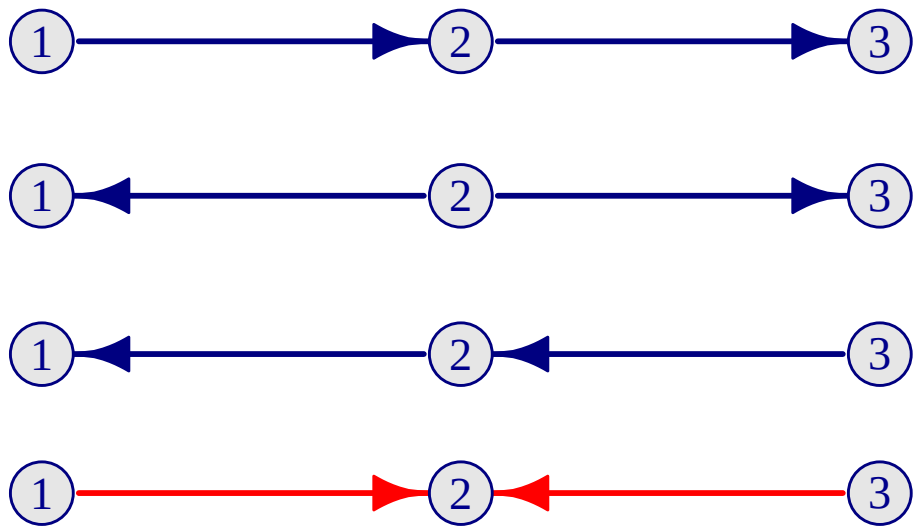
Consider the following 4 graphs



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

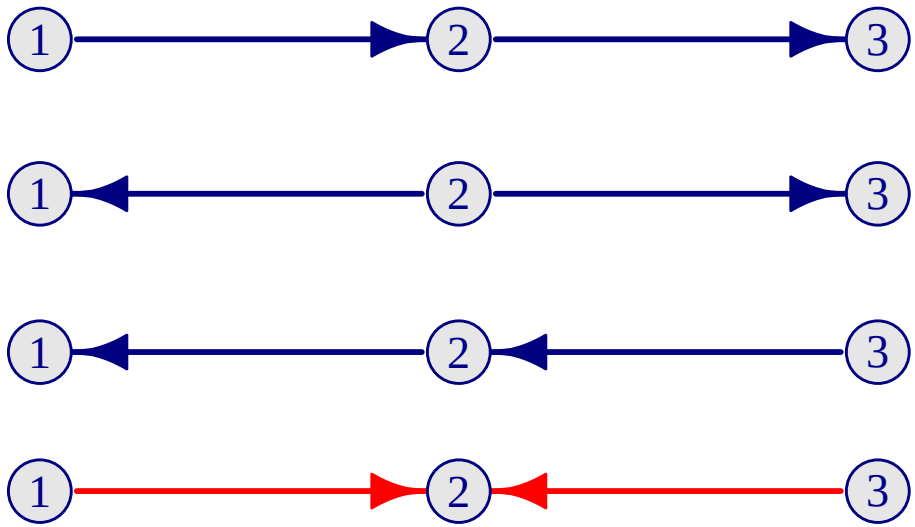
Markov Equivalence

Consider the following 4 graphs



Markov Equivalence

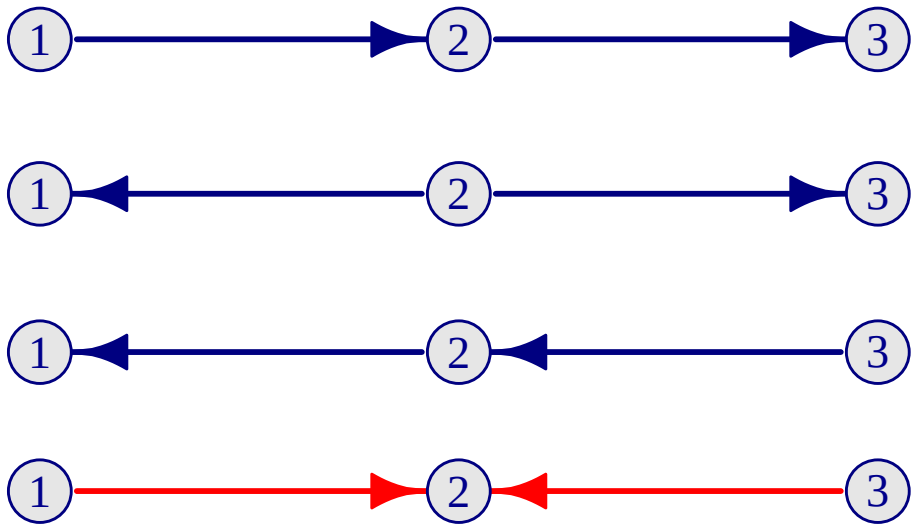
Consider the following 4 graphs



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

Markov Equivalence

Consider the following 4 graphs

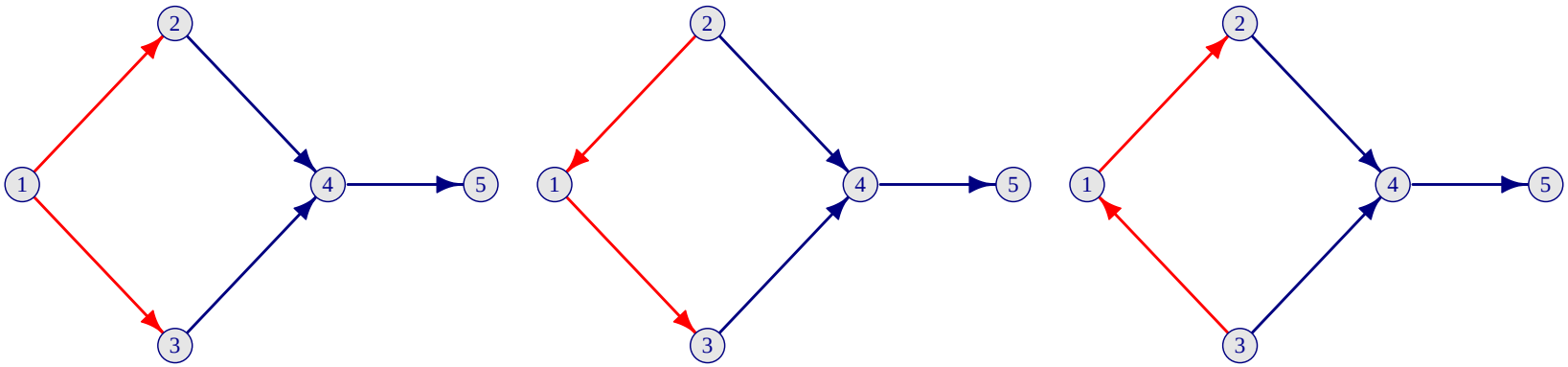


In the first 3 graphs, $X_1 \perp\!\!\!\perp X_3 \mid X_2$!
Two graphs that imply the same CI relationships via d-separation are called Markov equivalent

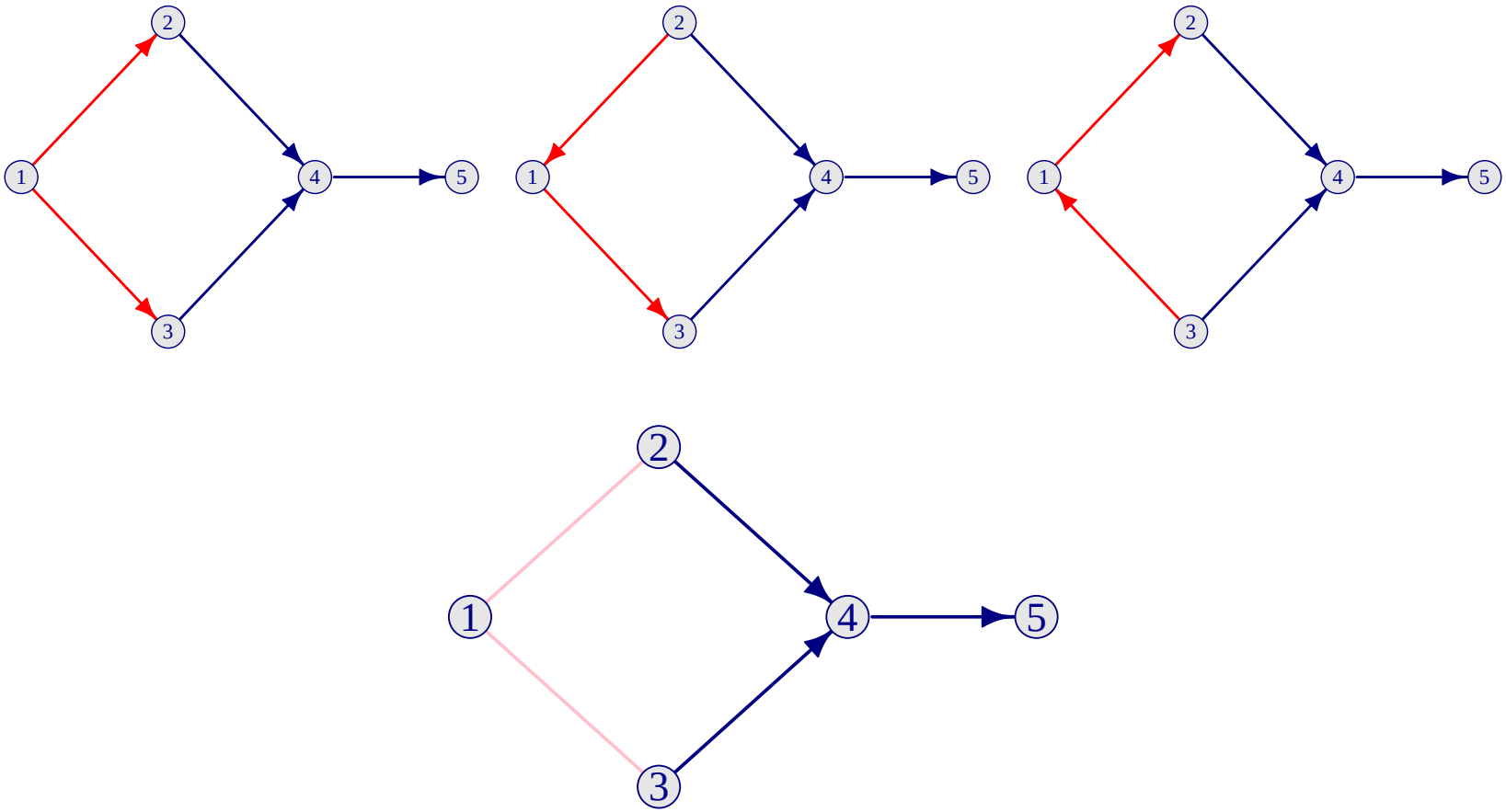
Representation of Markov Equivalence

- Markov equivalent graphs correspond to the same probability distribution and cannot be distinguished from each other based on observational data!
- Therefore, the direction of edges that correspond to Markov equivalent graphs cannot be determined
- These edges are shown as undirected in the graph
- The resulting graph is a CPDAG (completed partially directed acyclic graph), and is the best we can do!

CPDAGs



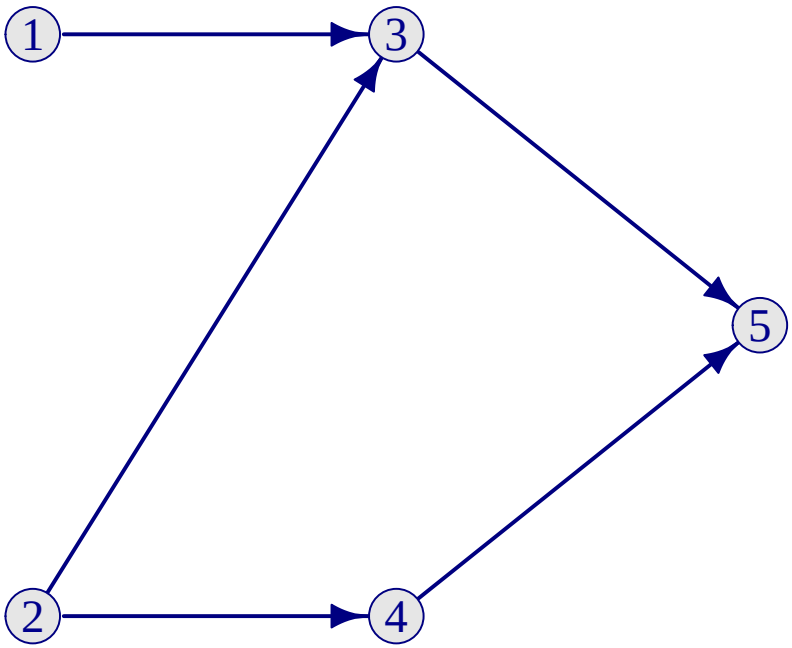
CPDAGs

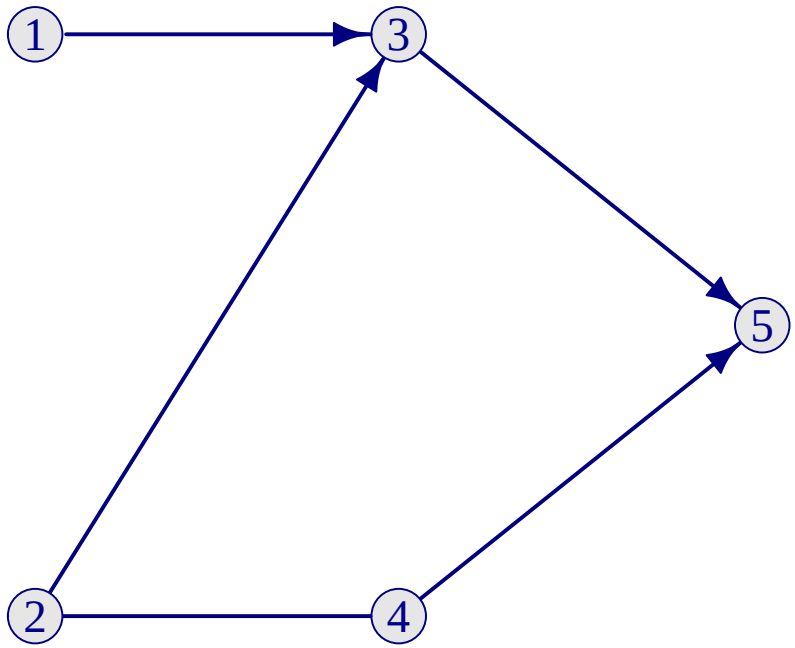


Finding Partial Directions in DAGs

- Partial directions in DAGs can be learned 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 remaining unmarried triplets $i \rightarrow k$—j as $i \rightarrow k \rightarrow j$
  - If $i \rightarrow k \rightarrow j$ and i—j then orient as $i \rightarrow j$
  - If i—m—j and $i \rightarrow k \leftarrow j$ are unmarried colliders and m—k, then orient as $m \rightarrow k$

Example





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

$S_{1,4} = \emptyset$

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

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

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

The bnlearn package

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

Analysis of Protein Flow Cytometry using bnlearn

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

- For GS need to choose α (alpha), the false positive probability for selecting edges
- gs (and other structure-based methods) find a PCDAG
- hc gives a directed graph (with highest score)
  - Multiple criteria implemented 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
```

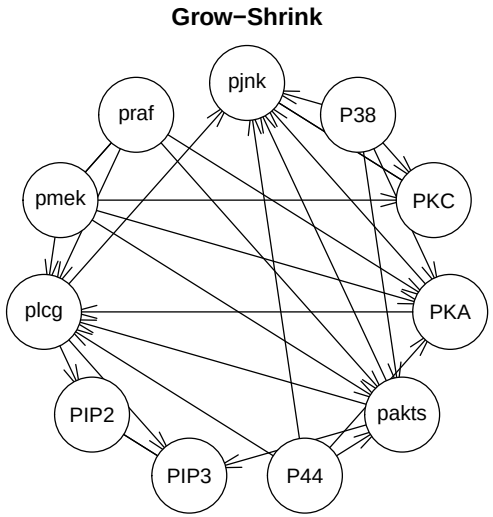
Analysis of Protein Flow Cytometry using bnlearn

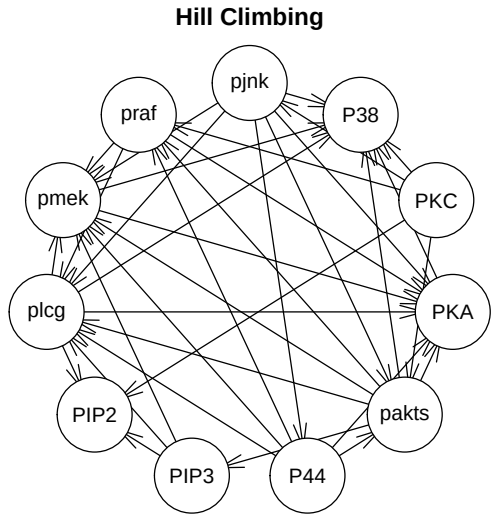
```
> dag2
  Bayesian network learned via Score-based methods

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

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

Analysis of Protein Flow Cytometry using `bnlearn`

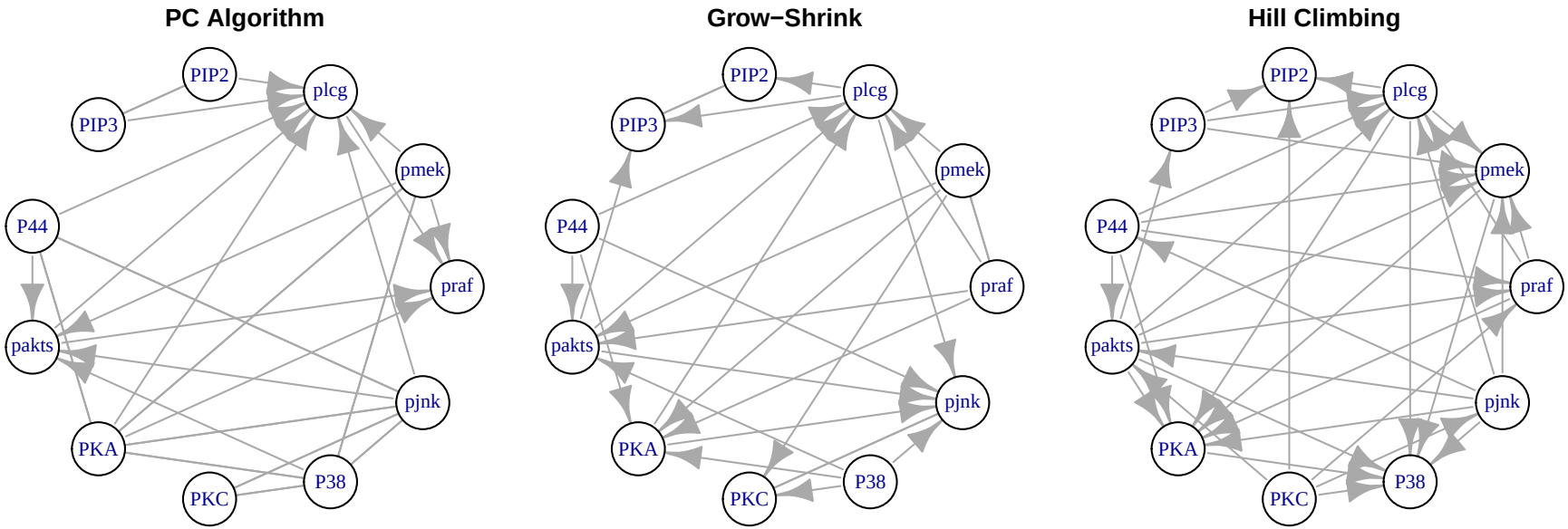




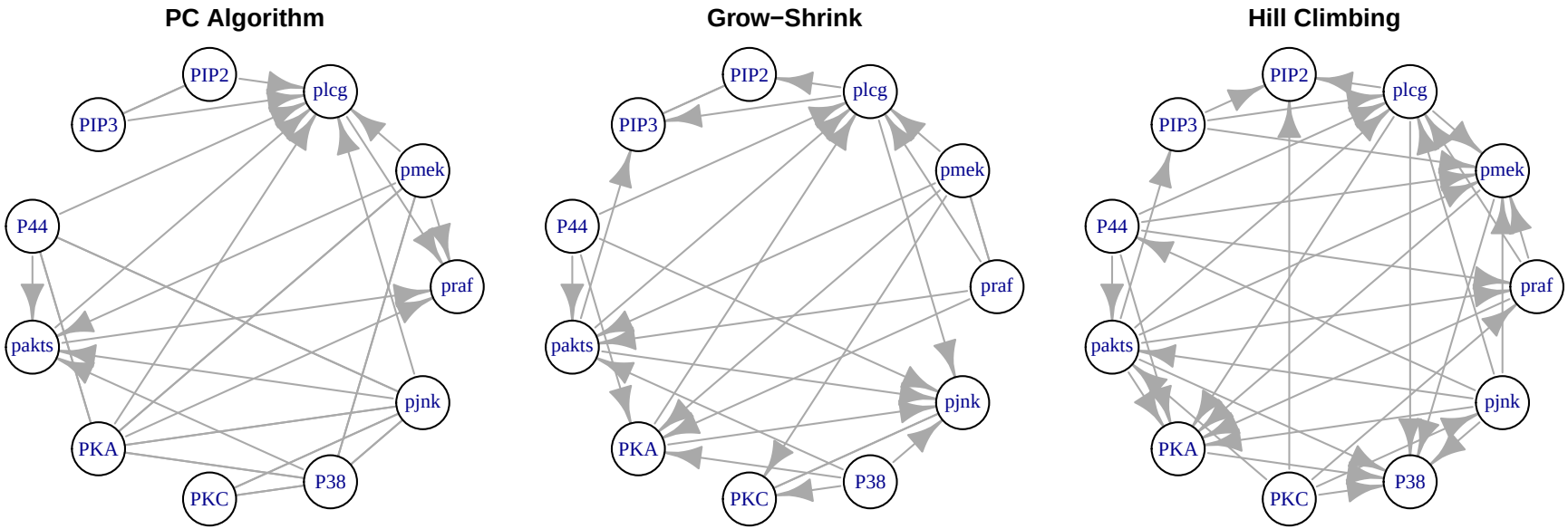
The two graphs are quite different

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

Comparison of Results for Protein Flow Cytometry Data



Comparison of Results for Protein Flow Cytometry Data



- The estimated graphs are quite different!
- Constrained-based methods seem to have more similarities (at least in terms of structure)
- The estimate from HC has more edges; we can change the criterion, but cannot directly control the sparsity

Penalized Likelihood Estimation of DAGs

- Recall that causal relationships (and probability distributions) in DAGs can be represented with structural equation models

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

- And, for Gaussian random variables,

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

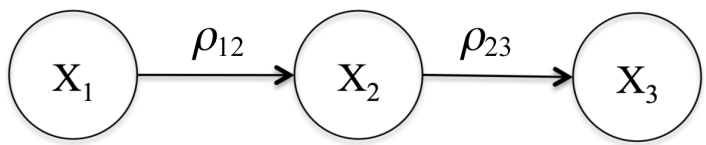
Penalized Likelihood Estimation of DAGs

- Recall that causal relationships (and probability distributions) in DAGs can be represented with structural equation models

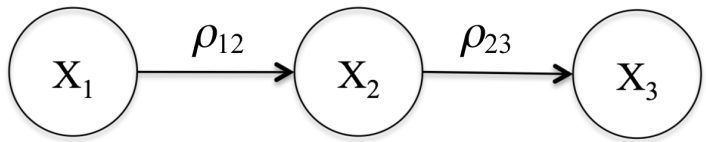
$$X_i = f_i(\mathrm{pa}_i, \gamma_i), \qquad i = 1, \cdots, p$$

- And, for Gaussian random variables,

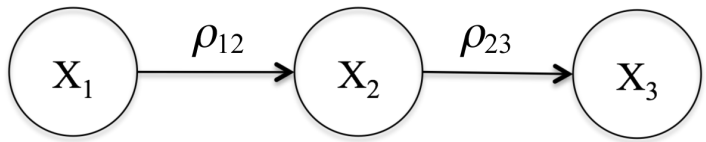
$$X_i = \sum_{j \in \mathrm{pa}_i} \rho_{ji} X_j + \gamma_i, \qquad i = 1, \cdots, p$$



Penalized Likelihood Estimation of DAGs

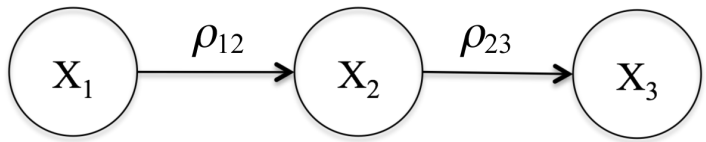


Penalized Likelihood Estimation of DAGs



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

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

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

  `after some math...`

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

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

[1] Shojaie & Michailidis (2010)

Penalized Likelihood Estimation of DAGs

- In high dimensions, we can solve a penalized version of this problem, e.g. by adding a lasso penalty $\lambda \sum_{i<j} |A_{ij}|$
- Also, can solve the problem as $(p-1)$ lasso regression, where each variable is regressed on prior variables in the ordering:

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

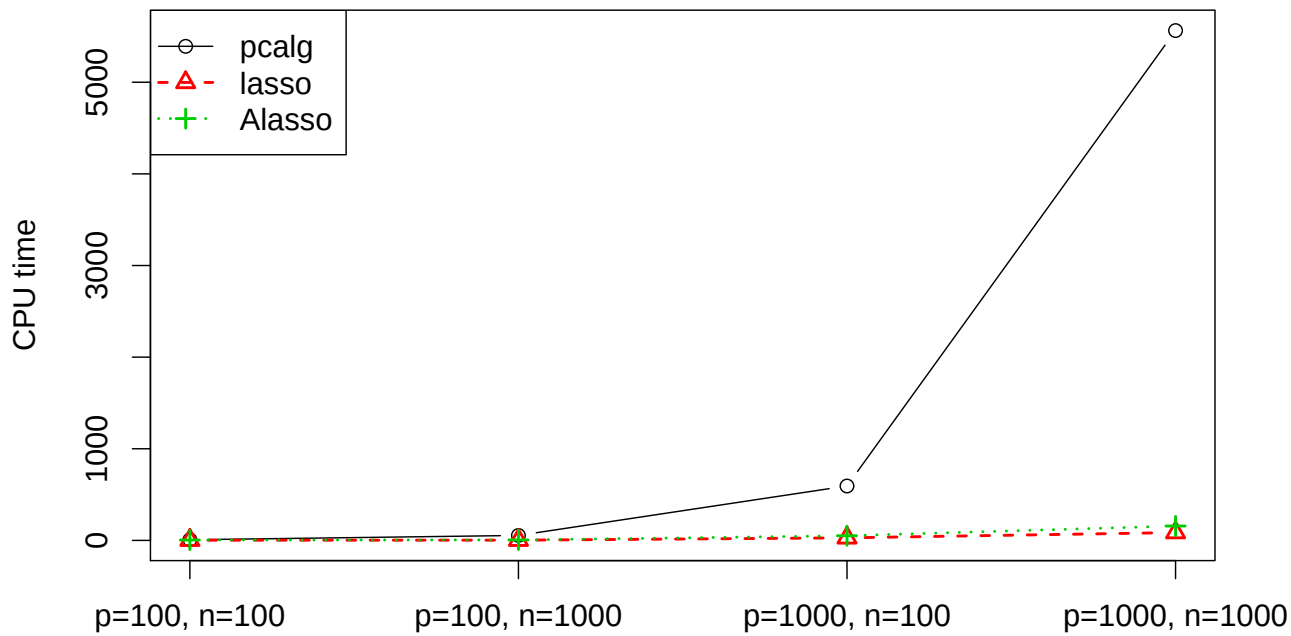
- As in `glasso`, λ is a tuning parameter that controls the amount of sparsity; $\lambda = \frac{2}{\sqrt{n}} Z_{\alpha/(2p^2)}$ controls a false positive probability at level α

Computational Complexity

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

Computational Complexity

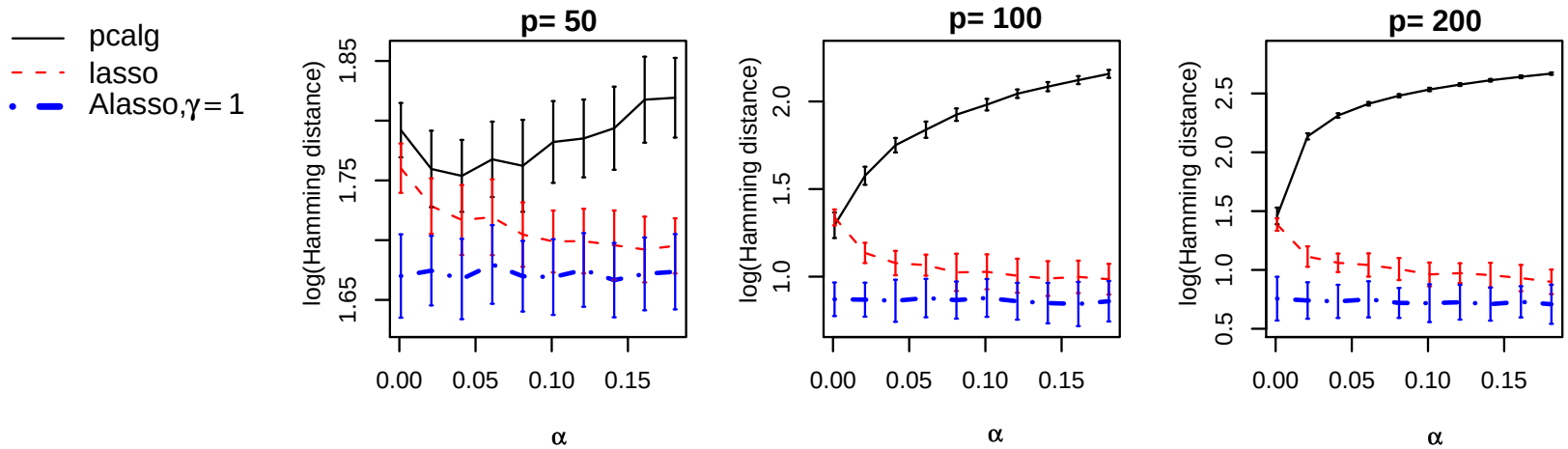
- Compared to `pcalg`, this method runs much faster: $\sim np^2$ operations vs $\sim p^q$ (q is the max degree)
- Can be easily implemented in `R` as $p-1$ regressions using `glmnet`. A more general version is available in the `spacejam` package, which also includes estimation for non-Gaussian data



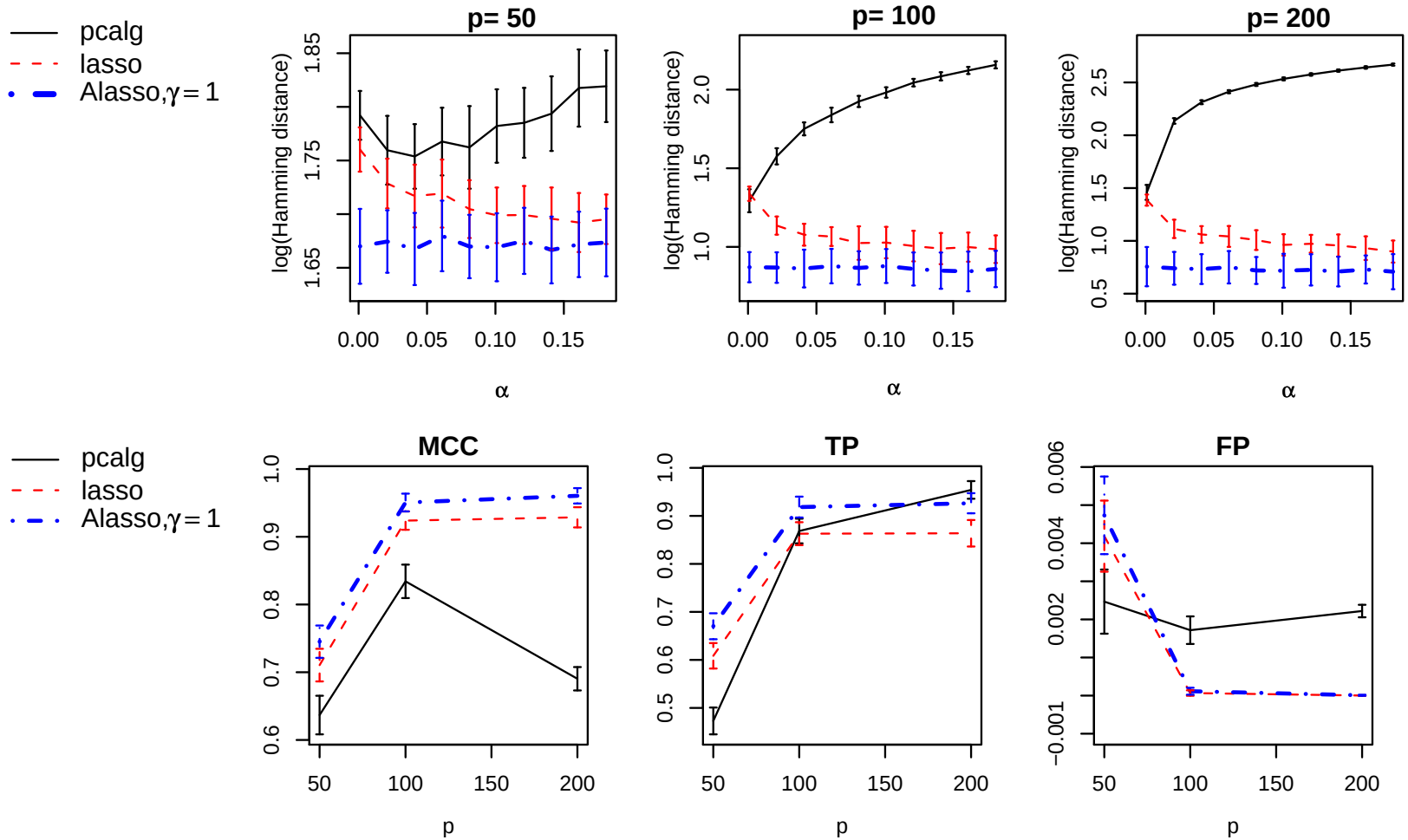
Simulation Studies

- Settings:
  $p = 50, 100, 200$
  $n = 100$
- Performance Criteria
  1. Matthew's Correlation Coefficient (MCC): ranges between -1 (worst fit) and 1 (best fit)
  2. Structural Hamming Distance (SHD): sum of false positive and false negatives
  3. True positive and false positive rates
- Tuning parameter for both PC-Algorithm and penalized likelihood method based on false positive error α

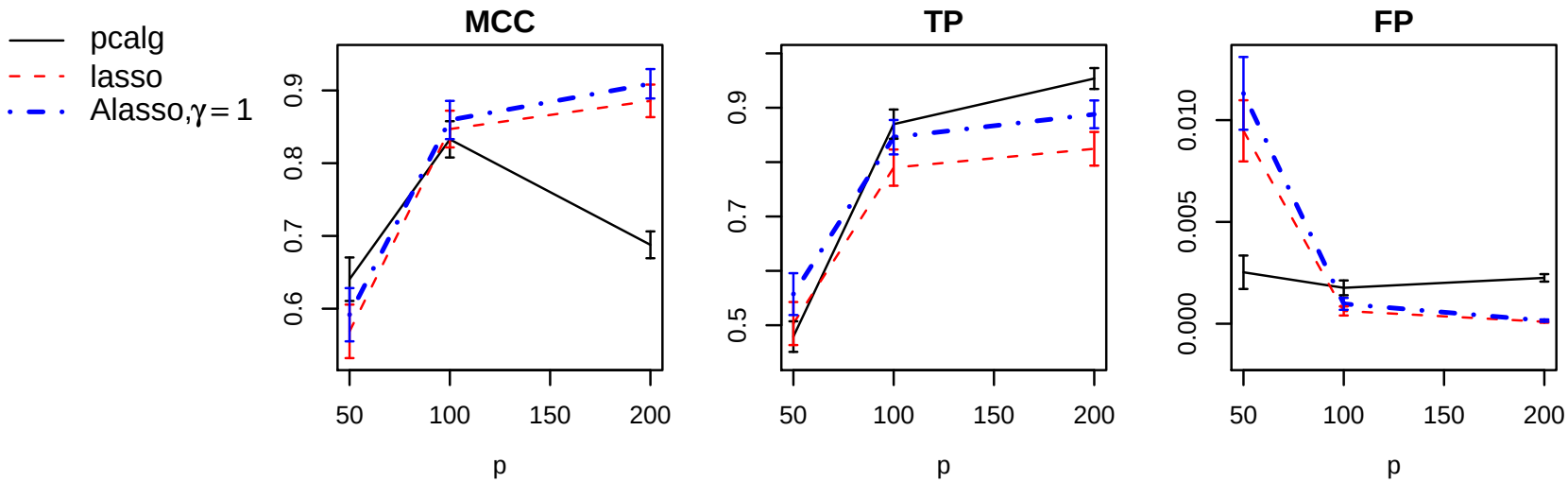
Gaussian Observations



Gaussian Observations



What if we don't know the causal ordering?

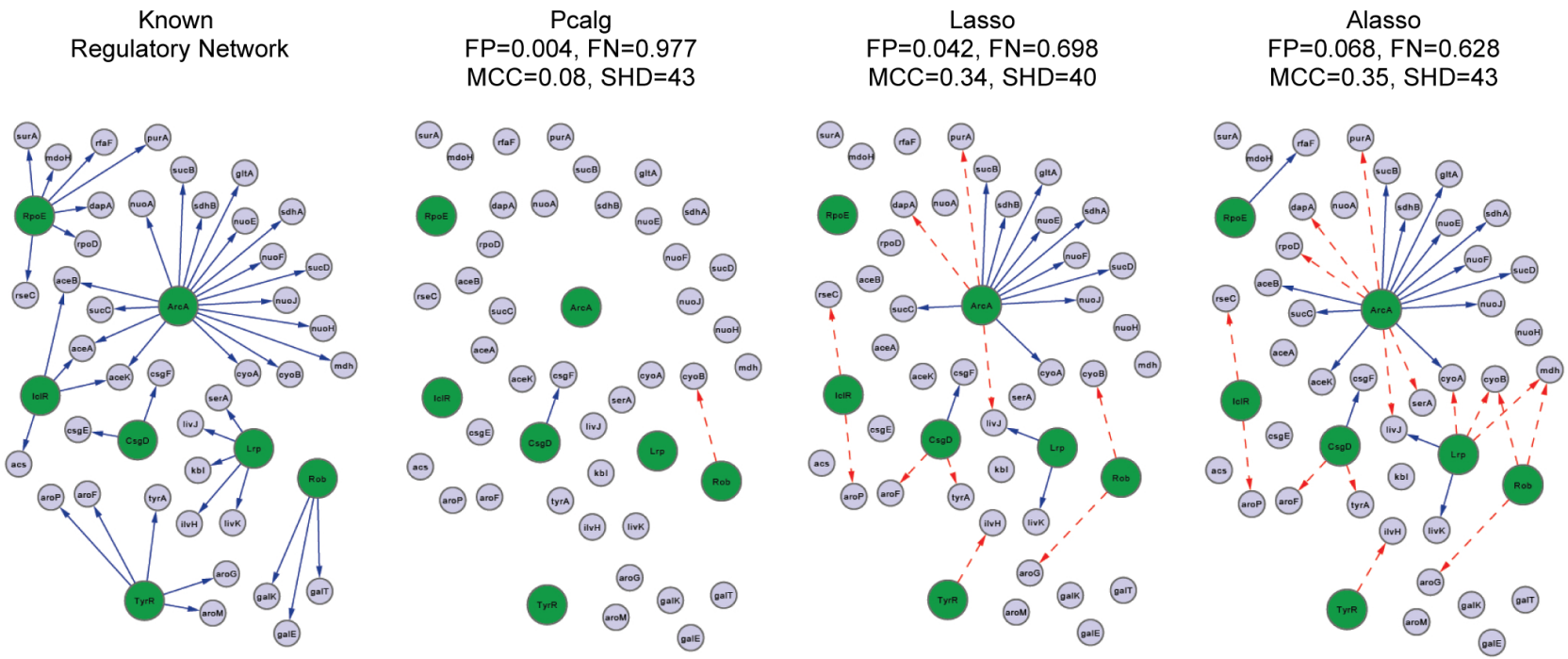


Regulatory Network of E-Coli

- 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



Summary

- Estimation of DAGs from observational data is both conceptually and computationally difficult
- Constraint-based and search-based algorithms become slow in high dimensions
- Also, may not be able to distinguish DAGs from observational data (Markov equivalence)
- Efficient penalized likelihood methods can estimate DAGs if the ordering is known
- Efficient implementations in `R` available for most methods
- Different methods need different tuning parameters...