

Bayesian network models for gene regulation

Florian Markowetz

Max Planck Institute for Molecular Genetics
– Computational Molecular Biology –
Berlin, Germany

<http://compdiag.molgen.mpg.de/>

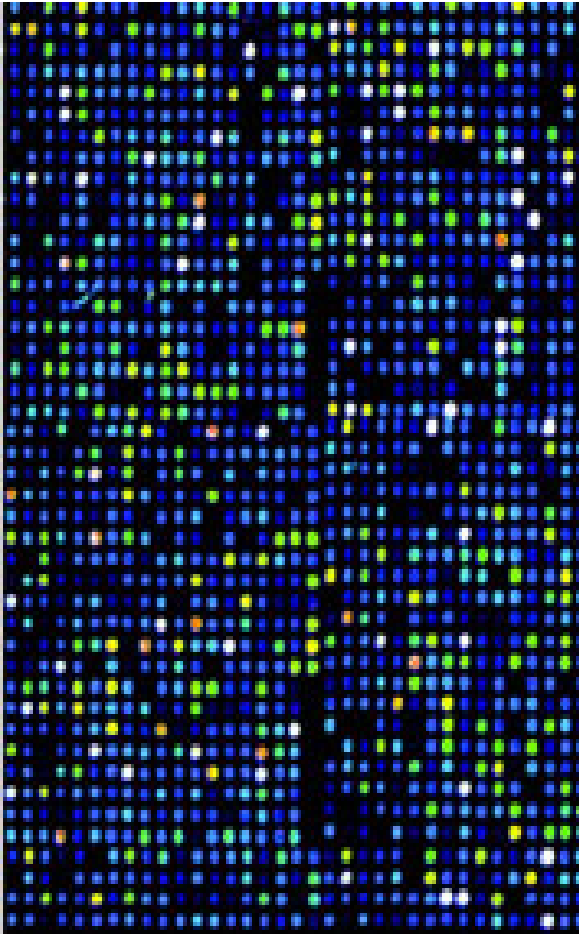


Lectures on Microarray Analysis

Free University Berlin

Winter term 03/04

— Genetic networks —



- Microarrays provide a snapshot of gene expression in a cell. Genes are not expressed independently, they regulate each others activity.

Goal: Reconstruct the gene regulation network!

- We want a **probabilistic method** that can handle noisy data.
- **Causality, not correlation!** Is the effect of a mutated gene on a target direct, or mediated by other genes? What is the nature of the interaction between genes (e.g. does gene A inhibit gene B)?



— Bayesian network —

A **Bayesian network** for $\mathbf{X} = \{X_1, \dots, X_n\}$ consists of

1. a **network structure** \mathcal{S}

- directed acyclic graph (DAG),
- nodes \leftrightarrow variables = genes,

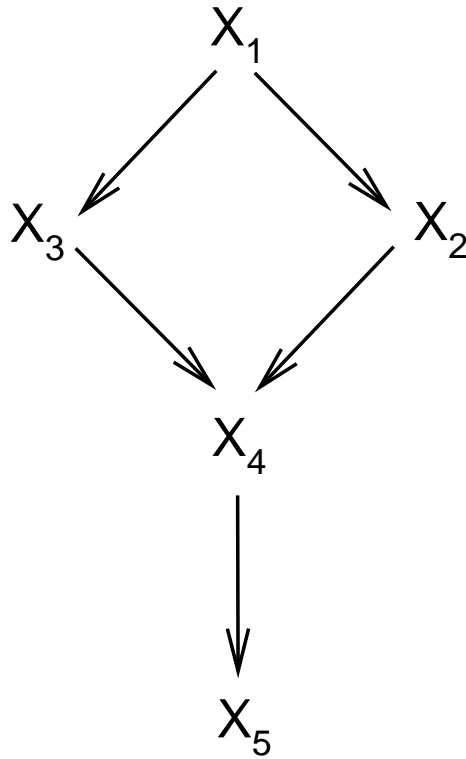
2. a set of **probability distributions** \mathcal{P}

- locally: conditional distribution of a variable X_i given its parents $pa(i)$ in the graph \mathcal{S} :

$$\mathcal{P} = \{ P(X_i \mid X_{pa(i)}) \}$$



— Bayesian networks *cont'd* —



$(\mathcal{S}, \mathcal{P})$ encode the joint distribution $P(\mathbf{X})$:

$$\begin{aligned} P(\mathbf{X}) &= \prod_{i=1}^n P(X_i | X_1, \dots, X_{i-1}) \\ &= \prod_{i=1}^n P(X_i | X_{pa(i)}) \end{aligned}$$

The DAG structure \mathcal{S} depends on the ordering of the variables.



— Conditional independence —

Independence of two random variables

$$X \perp Z \Leftrightarrow P(X, Z) = P(X) \cdot P(Z)$$



Conditional Independence given a third

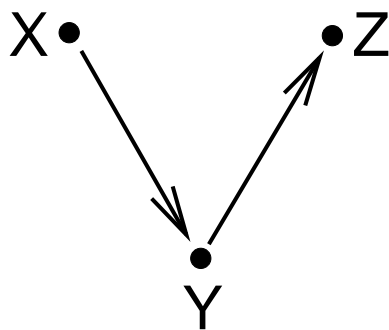
$$\begin{aligned} X \perp Z \mid Y &\Leftrightarrow P(X, Z \mid Y) = P(X \mid Y) \cdot P(Z \mid Y) \\ &\Leftrightarrow P(X \mid Y, Z) = P(X \mid Y) \end{aligned}$$

If I know Y , then Z does not add to my knowledge of X .



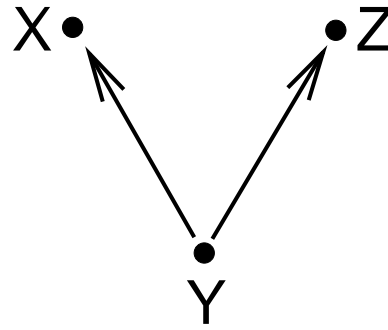
— Conditional independence in the graph —

Three possibilities how a path from X to Z passes through Y :



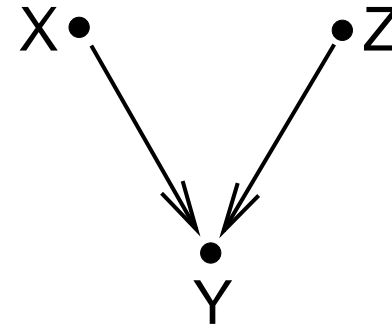
chain

- linear -



fork

- diverging -



collider

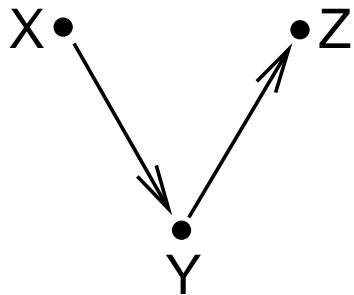
- converging -

In which cases holds $X \perp Z \mid Y$?

Does Y block the path from X to Z ?



— Blocking a path I —



Chain/linear.

Observing Y blocks the path from X to Z :

$$X \perp Z \mid Y$$

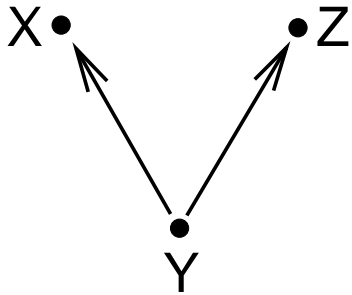
But without the knowledge of Y the path remains open:

$$X \not\perp Z \mid \emptyset$$

$$P(X, Z \mid Y) = \frac{P(X, Y, Z)}{P(Y)} = \frac{P(X)P(Y \mid X)P(Z \mid Y)}{P(Y)} = P(X \mid Y)P(Z \mid Y)$$



— Blocking a path II —



Fork/diverging.

Observing Y blocks the path from X to Z :

$$X \perp Z \mid Y$$

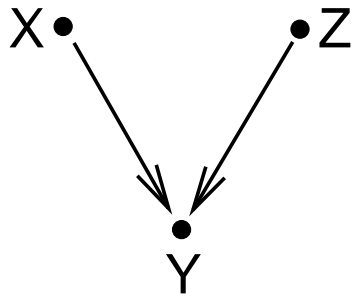
But without the knowledge of Y the path remains open

$$X \not\perp Z \mid \emptyset$$

$$P(X, Z \mid Y) = \frac{P(X, Y, Z)}{P(Y)} = \frac{P(X \mid Y)P(Y)P(Z \mid Y)}{P(Y)} = P(X \mid Y)P(Z \mid Y)$$



— Blocking a path III —



Collider/converging.

X and Z are independent

$$X \perp Z \mid \emptyset,$$

but knowledge of Y results in *explaining away* (or *selection bias*):

$$X \not\perp Z \mid Y$$

$$P(X, Y, Z) = P(X)P(Y|X, Z)P(Z) = P(X)P(Z)\frac{P(X, Y, Z)}{P(X, Z)}$$



— d-separation —

These three observations can be combined to the definition of *d-separation*.

Definition: A path p in a DAG G is said to be **d-separated (or blocked)** by a set of nodes Y if and only if

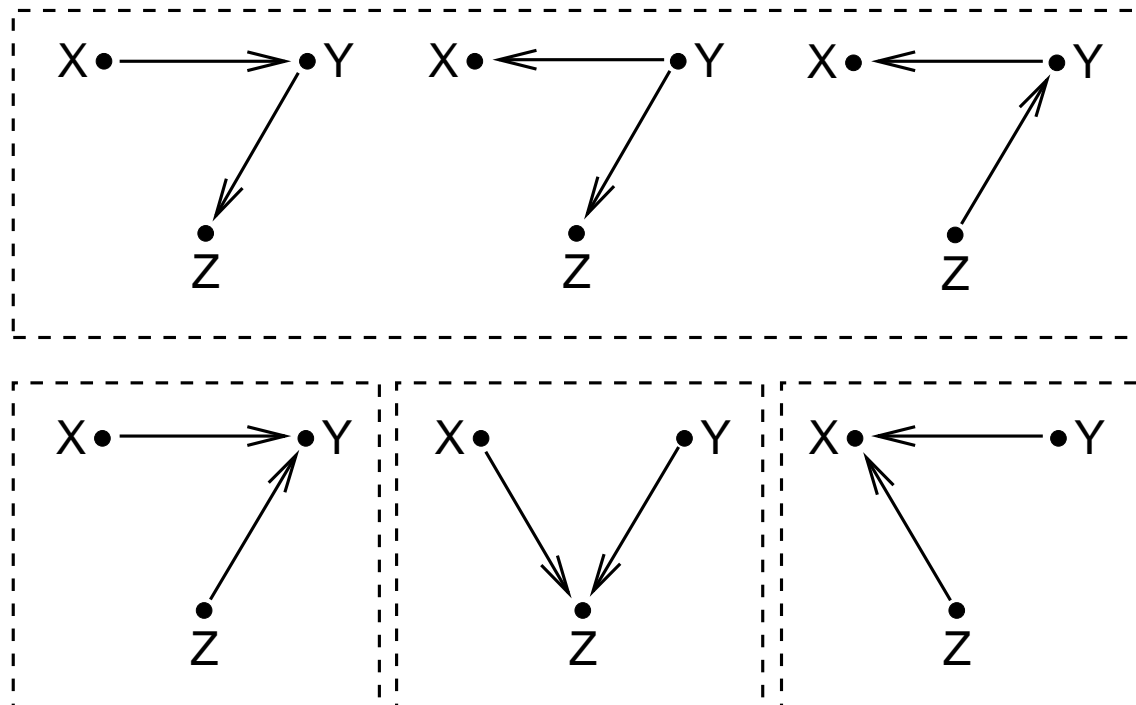
1. p contains a **chain** $i \rightarrow m \rightarrow j$ or a **fork** $i \leftarrow m \rightarrow j$ such that the middle node m is in Y , or
2. p contains an **collider** $i \rightarrow m \leftarrow j$ such that the middle node m is **not** in Y and such that no descendent of m is in Y .

A set Z is said to d-separate X from Y if and only if Z blocks every path from a node in X to a node in Y .



— Equivalence of Networks —

Two structures are **equivalent** if both represent the same independence restrictions.



Even with infinitely many observations we can not decide between the DAGs in the same equivalence class.



— Representation of an equivalence class —

Theorem [Verma and Pearl, 1990]

Two DAGs are equivalent iff they have the same skeleton and the same v-structures.

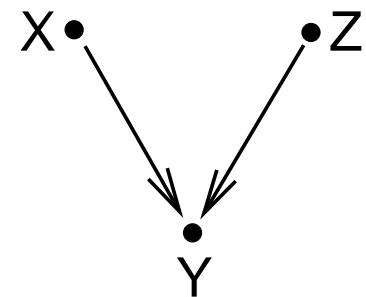
- **skeleton:**

the **undirected graph** resulting from ignoring the directionality of all edges.

- **v-structure:**

v-structures are **immoral families**: a child with unmarried parents. Formally, an ordered triple of nodes (X, Y, Z) in a graph such that

- (1) $X \longrightarrow Y$ and $Z \longrightarrow Y$
- (2) X and Z are not adjacent.



Using the Theorem we can uniquely represent a equivalence class of DAGs by a **partially directed graph (PDAG)**.

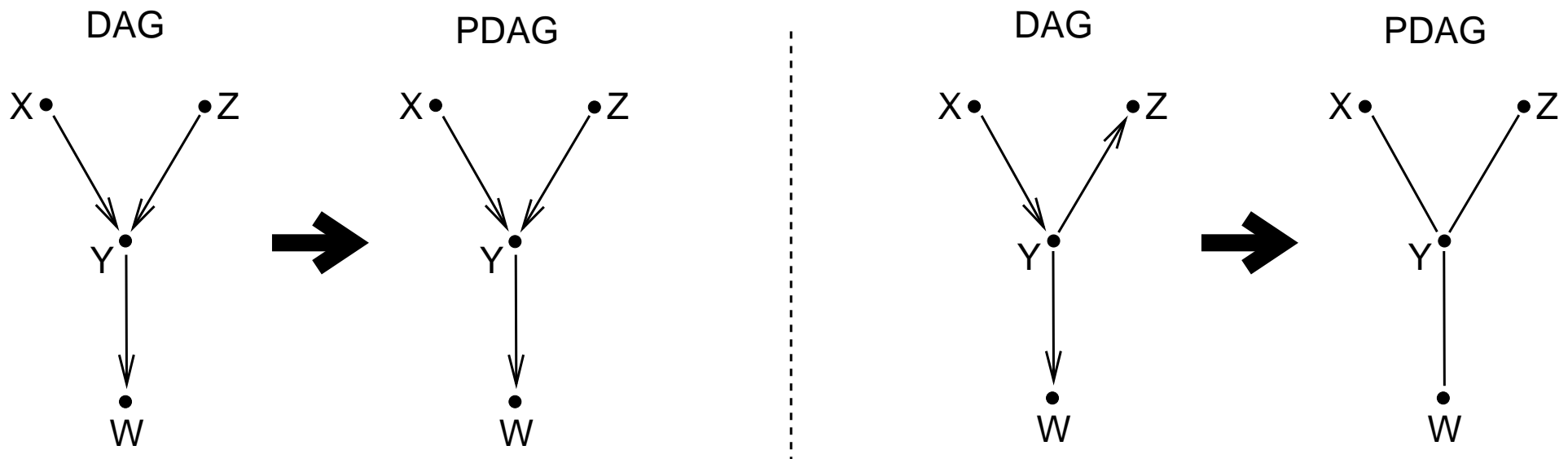


— DAG-2-PDAG —

Construction:

The PDAG identifying the equivalence class of a given DAG contains

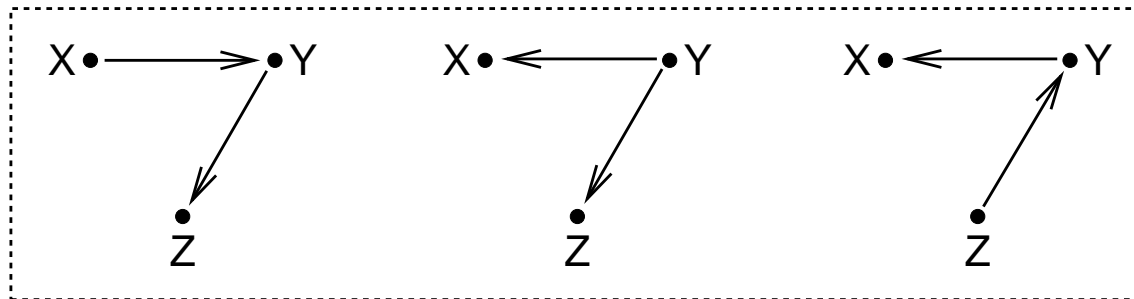
- a **directed edge** for every edge participating in or preventing a v-structure, and
- an **undirected edge** for all other edges.



— Causal vs. probabilistic networks —

A **Bayesian Network** models the distribution of observations. A **Causal Network** models the distribution of observations *and* effects of interventions.

Example:



All three networks show the same dependency structure: $X \perp Z \mid Y$

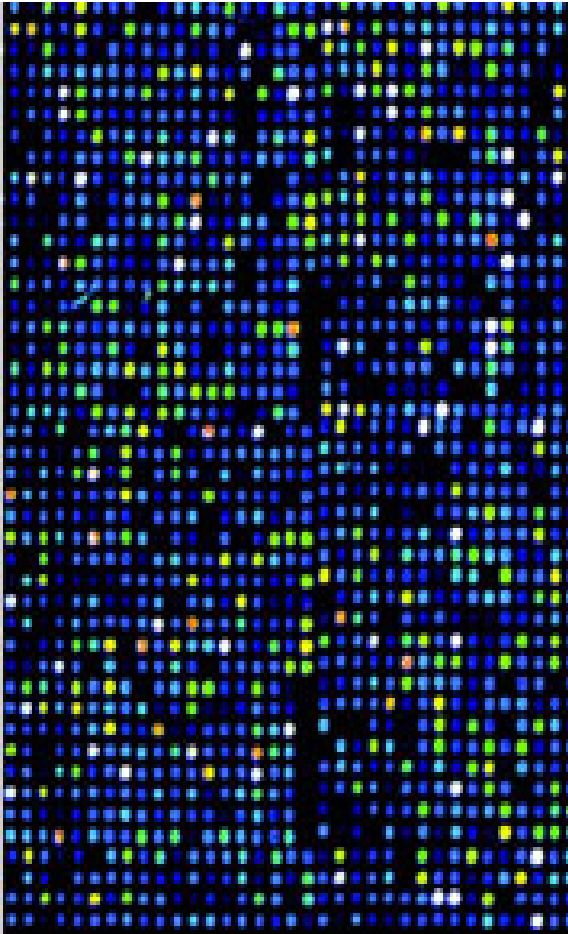
They are equivalent as Bayesian networks ...

... but the 'flow of causality' is different!

The perturbation of a node has different effects in each of the three graphs.



— Observation and Intervention —

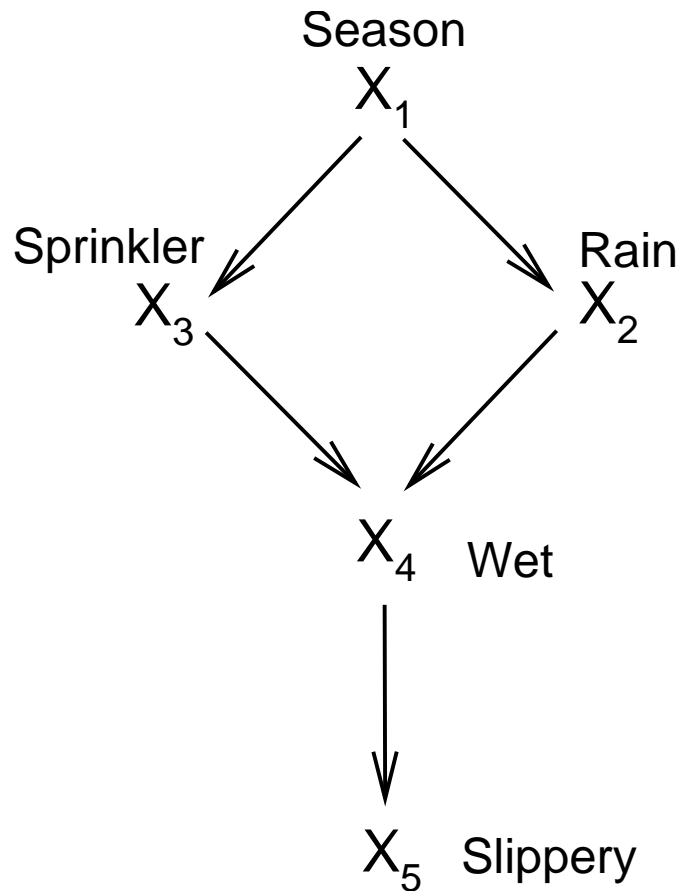


What effect does this result have on the reconstruction of genetic networks by BN?

- Arrows in the BN do not necessarily represent causal influence! From observations alone we can only learn whole equivalence classes, in general not a single DAG.
- But biologists not only observe, they also **intervene, perturb, disrupt** the gene network, e. g., by knock-out experiments.
- **How can we model interventions in Bayesian networks?**



— An example: the sprinkler network —



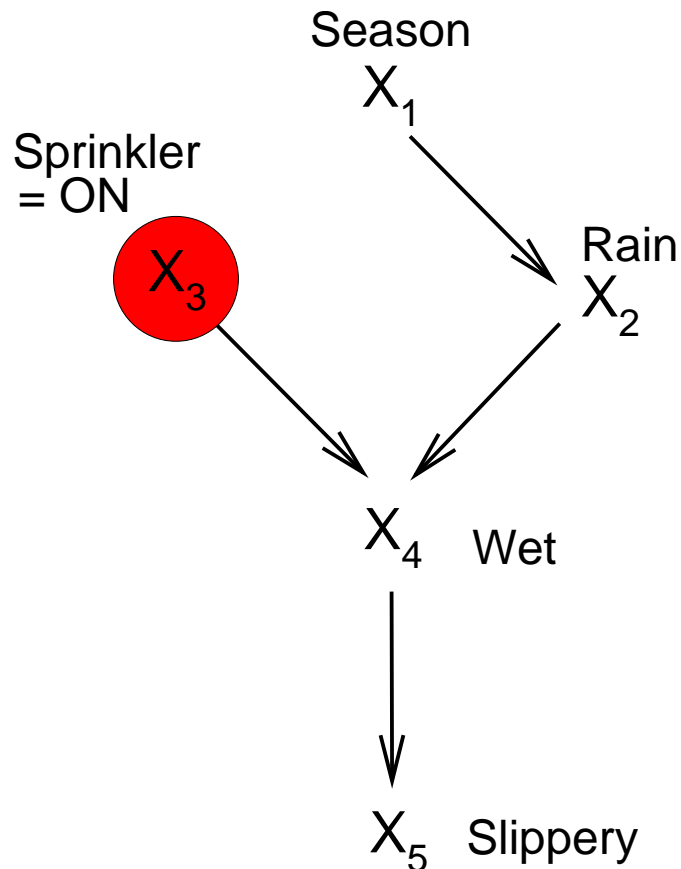
Imagine this being the true causal model we want to induce.

By looking out of the window we can collect observations of the states of all five variables.

What happens if we go out and turn the sprinkler on?



— Human manipulation —



Human intervention is a deterministic manipulation by forces outside the causal network model.

The sprinkler is no longer under the influence of any variables in the model, and thus, the arcs into it should be removed.

Intervention at X_3 :

- cut the edges from $X_{pa(3)}$ to X_3
- and set $P(X_3 \leftarrow \text{ON}) = 1$.



— Bayesian model selection —

The Bayesian way of learning a model structure from data:

1. **Scoring:** introduce a scoring function that evaluates each network with respect to the training data.
2. **Searching:** search for the optimal network according to this score.

The number of graphs grows exponentially in the number of nodes. For more than 5 nodes an exhaustive search is intractable.

Use heuristics: hill-climbing (with random restarts), simulated annealing, ...



— The Bayesian score —

The score evaluates the **posterior probability** of a graph given the data:

$$\begin{aligned} \text{Score}(\text{dag} : \text{data}) &= P(\text{dag} \mid \text{data}) \\ &\propto P(\text{data} \mid \text{dag}) \cdot P(\text{dag}) \end{aligned}$$

where $P(\text{data} \mid \text{dag})$ is the **marginal likelihood**:

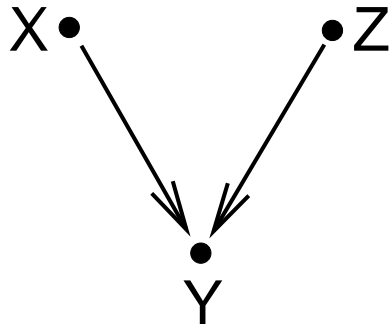
$$P(\text{data} \mid \text{dag}) = \int P(\text{data} \mid \text{dag}, \text{para}) P(\text{para} \mid \text{dag}) \, d\text{para}$$

For **Gaussian or discrete models**, the posterior can be given explicitly and it is **decomposable**:

$$\text{Score}(\text{dag} : \text{data}) = \prod_{\text{nodes}} \text{FamilyScore}(\text{node}, \text{parents} : \text{data}).$$



— Conditional discrete (“tabular”) distribution —



We assume: The real valued micorarray data is discretized.

The distribution of $Y \mid X, Z$ can be written in tabular form.

XZ	Y	
	0	1
00	$\theta_{Y,00,0}$	$\theta_{Y,00,1}$
10	$\theta_{Y,10,0}$	$\theta_{Y,10,1}$
01	$\theta_{Y,01,0}$	$\theta_{Y,01,1}$
11	$\theta_{Y,11,0}$	$\theta_{Y,11,1}$

Each row is a multinomial distribution over the states of Y given a configuration (x, z) of X and Z .

More general, let r_i be the number of states of gene/node X_i . (Often $r_i \equiv r$.) For $P(X_i \mid X_{pa(i)})$ holds

$$\theta_{Y,00,0} = P(Y = 0 \mid (X, Z) = (0, 0))$$

$$\#cols = r_i \quad \text{and} \quad \#rows = \prod_{j \in pa(i)} r_j$$



— The problem —

Learn the dependency structure S of n variables X_1, \dots, X_n .

A dataset D is a collection of m cases: $D = \{C_1, \dots, C_m\}$.

Each case C_h consists in realisations of all the variables: $C_h = (x_1^h, \dots, x_n^h)$.

Our background knowledge is denoted by K . It contains, e. g., the information, how the cases were experientially collected.

References:

COOPER AND YOO, *Causal discovery from a mixture of experimental and observational data*, UAI 1999

COOPER AND HERSKOVITS, *A Bayesian method for the induction of probabilistic networks from data*, Machine Learning, 9, 309-347 (1992)



— Assumption 1 —

Causal relationships are represented using causal Bayesian networks.

$$\begin{aligned} P(S \mid D, K) &\propto P(S, D \mid K) \\ &= P(S \mid K) P(D \mid S, K) \\ &= P(S \mid K) \int P(D \mid S, \theta_S, K) P(\theta_S \mid S, K) d\theta_S \end{aligned}$$



— Assumption 2 —

The cases in D are a random sample from the joint distribution given by a causal Bayesian network B with structure S and parameters θ_S .

Cases are independent *conditioned on the generating model*:

$$\begin{aligned} P(D \mid S, \theta_S, K) &= P(C_1, \dots, C_m \mid S, \theta_S, K) \\ &= \prod_{h=1}^m P(C_h \mid S, \theta_S, K), \end{aligned}$$



— Assumption 3 —

For each experimentally manipulated variable X_i in case C_h , the probability $P(C_h \mid S, \theta_S, K)$ is modeled by removing from S the arcs into X_i , and setting $P(X_i = k \mid K) = 1$, where k is the value to which X_i was manipulated.

Consider a case C_h that contains a variable X_i that is manipulated to state k . $P(C_h \mid \theta_S, S, K)$ is inferred by:

1. modify S by removing the arcs into X_i ,
2. remove the parameters θ_S that correspond to the removed arcs in S ,
3. set $P(X_i = k \mid K) = 1$,
4. use this mutilated Bayesian network to infer the probability of the state of the variables in $C_h - \{X_i\}$



— Assumption 4 —

There are no missing data or hidden variables.

$$\begin{aligned} P(D \mid S, \theta_S, K) &= \prod_{h=1}^m P(C_h \mid S, \theta_S, K) \\ &= \prod_{h=1}^m \prod_{i=1}^n P(x_i^h \mid pa_i^h, \theta_S, K) \end{aligned}$$

since $C_h = (x_1^h, \dots, x_n^h)$, where x_i^h is the realisation of node X_i in case h , and pa_i^h denotes the state of the parents of X_i in case h .



— Assumption 5 and 6 —

Variables are discrete.

Parameter independence:

Global: For each causal Bayesian network structure, the parameters (probabilities) associated with one node are prob. independent of the parameters associated with other nodes.

Local: The parameters associated within a node given one instance of its parents are independent of the parameters of that node given other instances of its parent nodes.

Conditional distributions can be written in tabular form.

$$\theta_S = \prod_{i=1}^n \theta_i = \prod_{i=1}^n \prod_{j=1}^{q_i} \theta_{ij}$$



— Assumption 5 and 6 —

$$\begin{aligned} P(D \mid S, \theta_S, K) &= \prod_{h=1}^m \prod_{i=1}^n P(x_i^h \mid pa_i^h, \theta_S, K) \\ &= \prod_{i=1}^n \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} P(x_i = k \mid pa_i = j, \theta_S, K)^{N_{ijk}} \\ &=: \prod_{i=1}^n \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk}^{N_{ijk}} \end{aligned}$$

where

r_i = number of states of X_i ,

q_i = number of joint states of parents of X_i , and

N_{ijk} = number of cases in D in which node X_i is passively observed to have state k when its parents have states as given by j .



— Assumption 7 and 8 —

Parameter modularity: If a node has the same parents in two distinct networks, then the distribution of the parameters associated with this node are identical in both networks.

The prior distribution of parameters associated with each node is Dirichlet.

■ The vector $\theta_{ij} = (\theta_{ijk})_{k=1}^{r_i}$ has a Dirichlet distribution with parameters $\alpha = (\alpha_{ijk})$ if

$$\begin{aligned} P(\theta_{ij} \mid S, K) &= \text{Dir}(\theta_{ij1}, \dots, \theta_{ijr_i} \mid \alpha) \\ &= \frac{\Gamma(\sum_{k=1}^{r_i} \alpha_{ijk})}{\prod_{k=1}^{r_i} \Gamma(\alpha_{ijk})} \prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk}-1} \\ &= \frac{(\alpha_{ij+} - 1)!}{\prod_{k=1}^{r_i} (\alpha_{ijk} - 1)!} \theta_{ij1}^{\alpha_{ij1}-1} \dots \theta_{ijr_i}^{\alpha_{ijr_i}-1}. \end{aligned}$$



— Putting all assumptions together ... —

$$\begin{aligned}
 P(D \mid S, K) &= \int P(D \mid S, \theta_S, K) \cdot P(\theta_S \mid S, K) \, d\theta_S \\
 &= \int \prod_{i=1}^n \prod_{j=1}^{q_i} \prod_{k=1}^{r_i} \theta_{ijk}^{N_{ijk}} \cdot \frac{\Gamma(\alpha_{ij+})}{\prod_{k=1}^{r_i} \Gamma(\alpha_{ijk})} \prod_{k=1}^{r_i} \theta_{ijk}^{\alpha_{ijk}-1} \, d\theta_S \\
 &= \prod_{i=1}^n \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij+})}{\prod_{k=1}^{r_i} \Gamma(\alpha_{ijk})} \int \prod_{k=1}^{r_i} \theta_{ijk}^{N_{ijk} + \alpha_{ijk} - 1} \, d\theta_{ij} \\
 &= \prod_{i=1}^n \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij+})}{\prod_{k=1}^{r_i} \Gamma(\alpha_{ijk})} \cdot \frac{\prod_{k=1}^{r_i} \Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ij+} + N_{ij+})} \\
 &= \prod_{i=1}^n \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij+})}{\Gamma(\alpha_{ij+} + N_{ij+})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}
 \end{aligned}$$



— Learning from observations —

For **discrete models**, the marginal likelihood can be derived as:

$$P(\text{data} \mid \text{dag}) = \prod_{i=1}^n \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij+})}{\Gamma(\alpha_{ij+} + N_{ij+})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk})}{\Gamma(\alpha_{ijk})}$$

r_i = number of states of node X_i ,

q_i = number of joint states of parents of X_i ,

N_{ijk} = number of times we observe node X_i in state k given parental state j ,

α_{ijk} = parameter of the Dirichlet prior $P(\text{para} \mid \text{dag})$.

Multiplying $P(\text{data} \mid \text{dag})$ with a prior $P(\text{dag})$ reflecting your prior knowledge on network structure yields a scoring metric $\text{Score}(\text{dag} : \text{data})$.



— Learning by hard interventions: knock-outs —

The likelihood for data from hard interventions looks only slightly different:

$$P(\text{data} \mid \text{dag}) = \prod_{i=1}^n \prod_{j=1}^{q_i} \frac{\Gamma(\alpha_{ij+})}{\Gamma(\alpha_{ij+} + N_{ij+}^{\text{obs}})} \prod_{k=1}^{r_i} \frac{\Gamma(\alpha_{ijk} + N_{ijk}^{\text{obs}})}{\Gamma(\alpha_{ijk})}$$

Here N_{ijk}^{obs} is the number of **passive observations** $X_i = k \mid X_{pa(i)} = j$.

The interventions vanish in the calculations, because there $P(X_i \leftarrow k) = 1$.



— Software —

1. for MATLAB: the Bayesian Network Toolbox by Kevin Murphy
<http://www.ai.mit.edu/~murphyk/Software/BNT/bnt.html>
2. for R: the deal package by Susanne Bøttcher and Claus Dethlefsen.
<http://www.math.auc.dk/novo/deal/>
3. The Open Source Probabilistic Networks Library
<https://sourceforge.net/projects/openpnl/>



— What can go wrong? Some Caveats —

1. A method designed to output a graph will eventually output a graph. That does not mean the graph is in any way sensible.
2. Make sure, the DAG assumptions is not totally off the score.
3. There is a high variance in structure learning: if the data or even the preprocessing differ a little bit, a totally different structure may come out.
4. Make sure, the interpretation of the output is not just BioPoetry.
5. “Any method (or statistician) that takes a complex multivariate dataset and, from it, claims to identify one true model, is both naive and misleading.
David Edwards, Introduction to Graphical Modelling, Springer 2000



— Literature —

1. A bibliography of learning causal networks of gene interactions
www.molgen.mpg.de/~markowet/docs/network-bib.pdf
2. Friedman *et al.*, Using Bayesian Networks to Analyze Expression Data, RECOMB 2000.
3. Segal *et al.*, Module networks: identifying regulatory modules and their condition-specific regulators from gene expression data, Nat Genet. 2003 Jun; 34(2): 166-76.
4. Heckerman *et al.*, Learning Bayesian Networks: The Combination of Knowledge and Statistical Data, ML 1995.
5. Judea Pearl, Causality: Models, Reasoning and Inference, CUP 2000.



— Summary —

1. Definition of Bayesian networks
2. Conditional independence and d-separation
3. Equivalence classes of Bayesian networks
4. Modeling perturbations in Bayesian networks
5. How to derive a scoring metric for structure learning



— Thank you! Questions? —

