

# Network Modeling in Systems Biology with R

## Tutorial 1

Markov chain Monte Carlo (MCMC) sampling  
of Gaussian Bayesian networks  
for static (steady-state) data

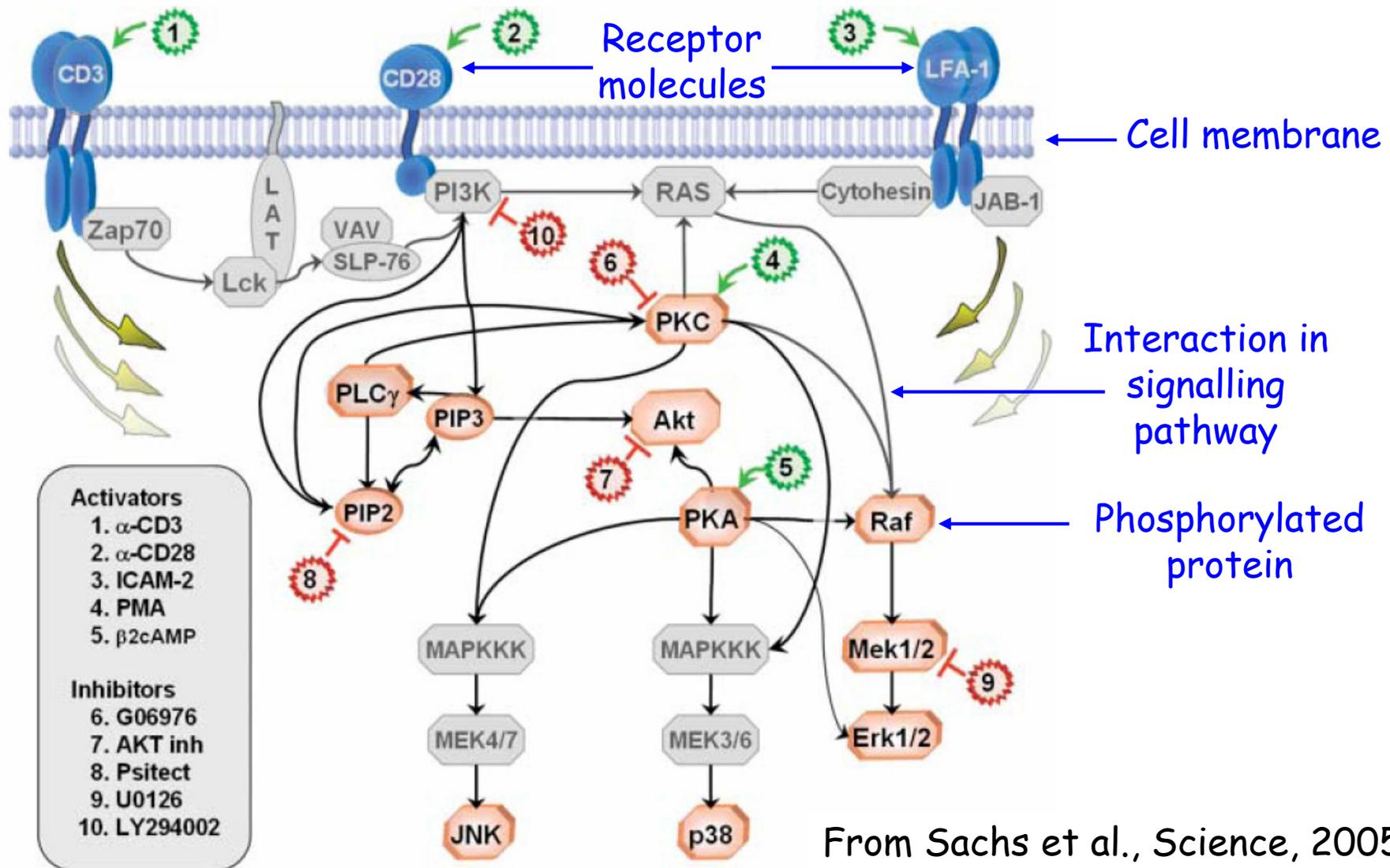
**Statistical Computing 2010, Reisensburg**

**21-June-2010**

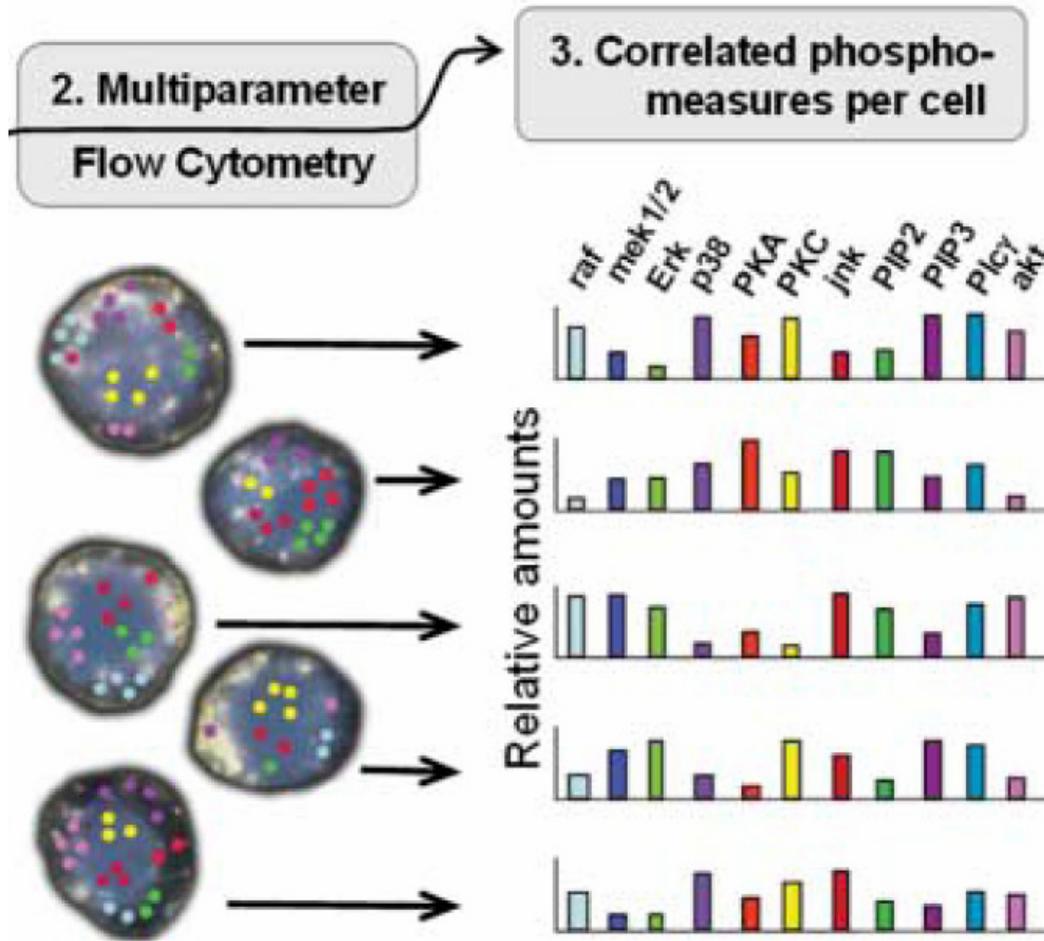
**Miriam Lohr and Marco Grzegorzczak**  
**Department of Statistics**  
**TU Dortmund University**

 **tu** technische universität  
dortmund

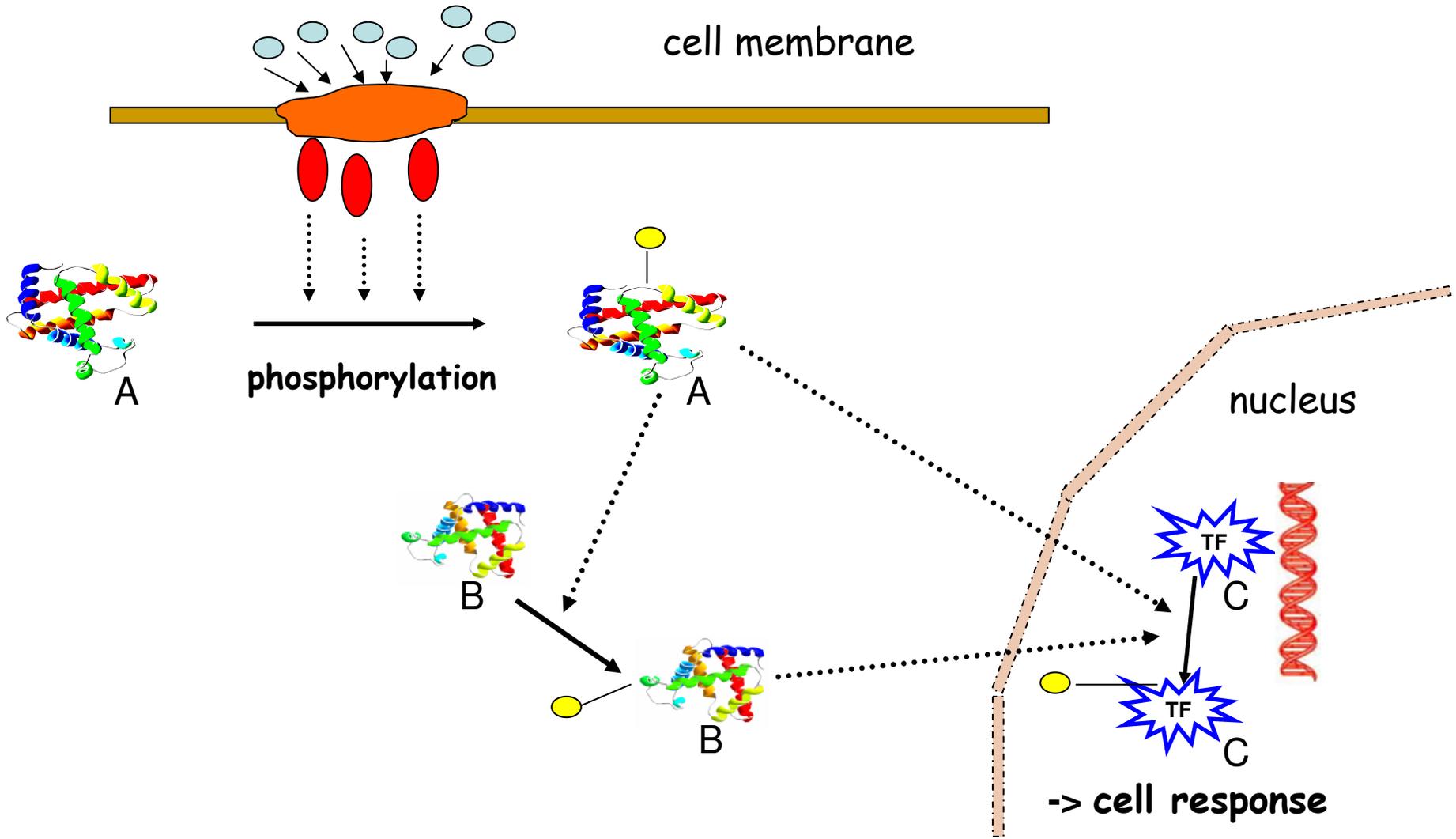
# Regulatory networks and protein signalling pathways



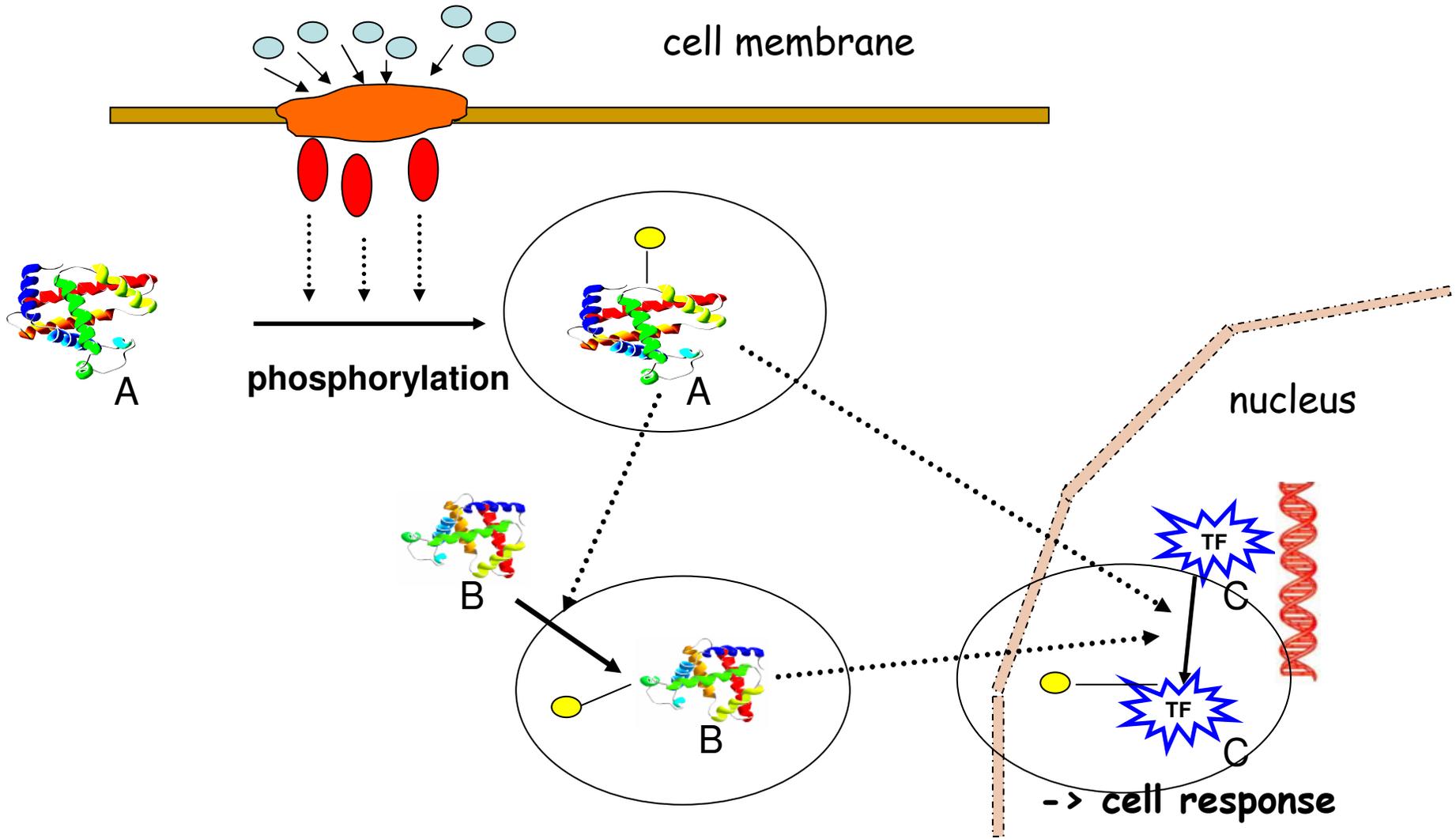
# Flow cytometry technology



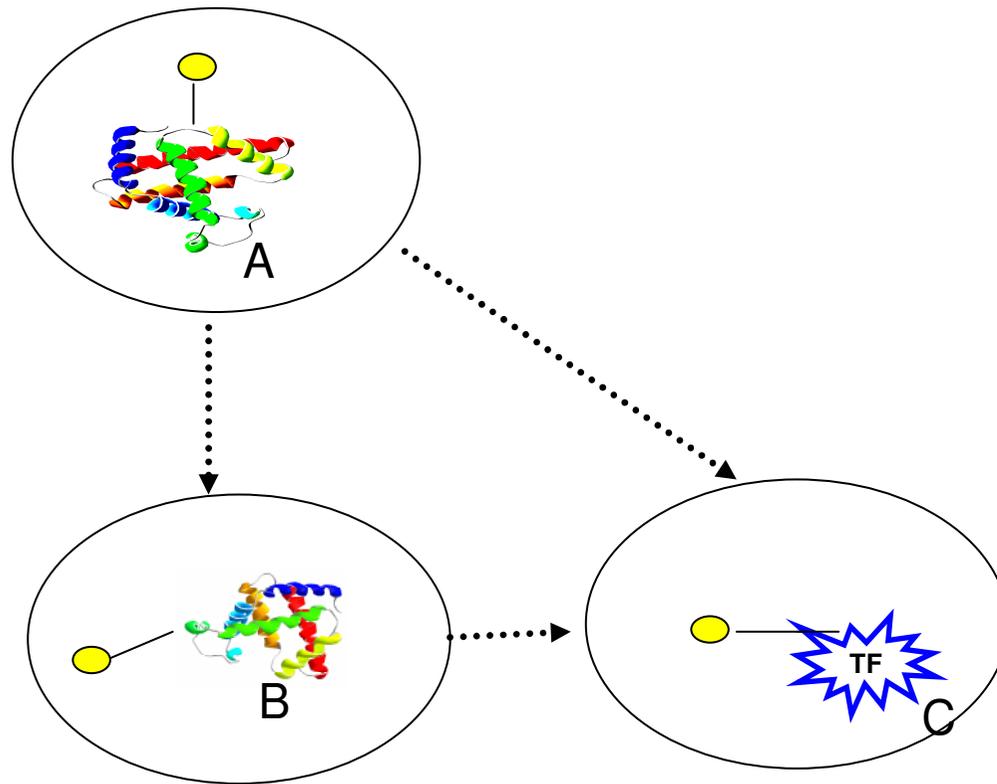
# Protein activation cascade



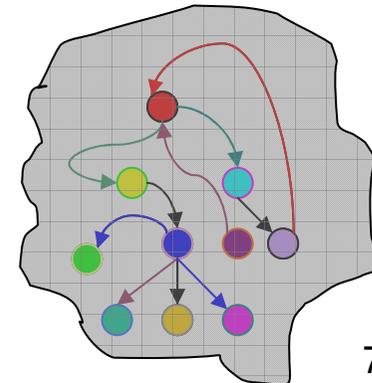
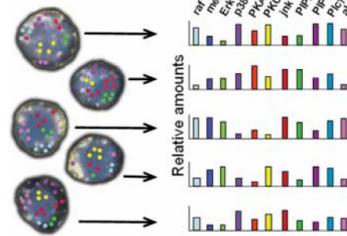
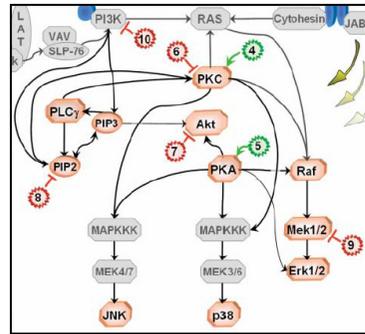
# Protein activation cascade



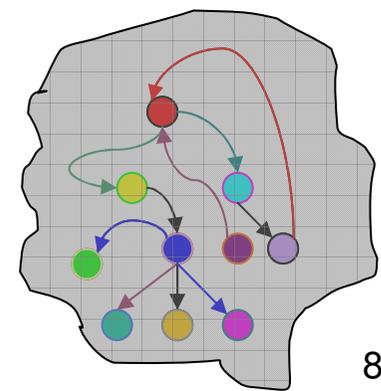
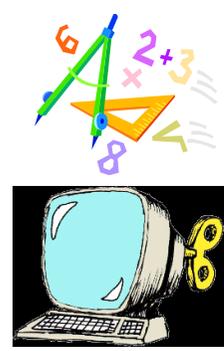
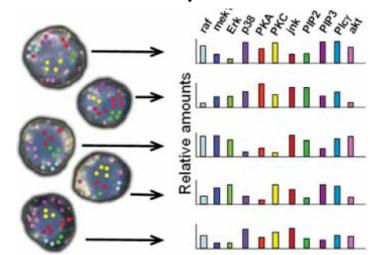
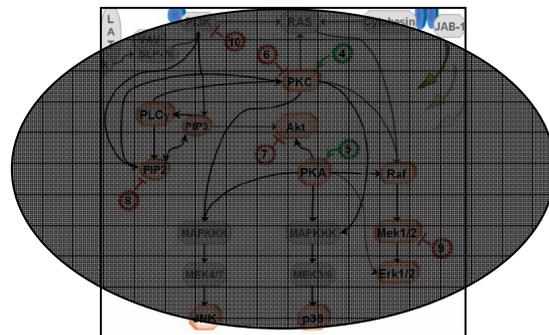
# Protein activation cascade



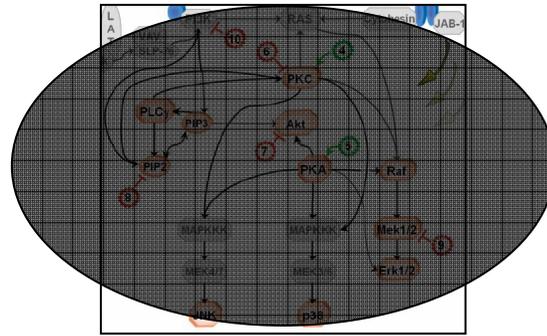
# True regulatory pathway/network



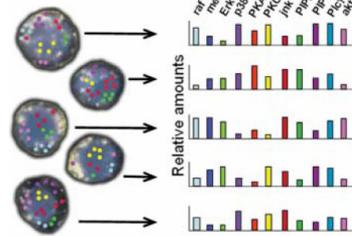
possibly completely unknown



possibly completely unknown

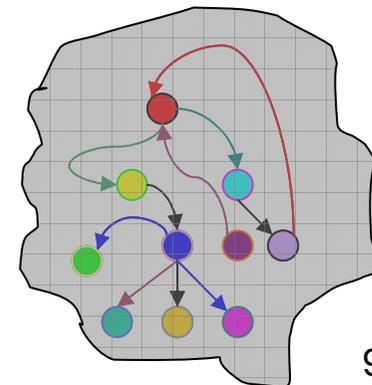
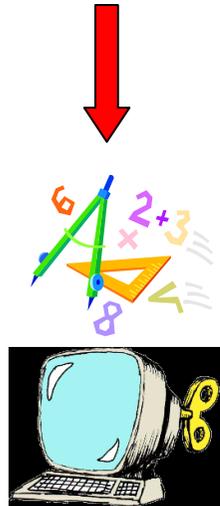


E.g.: Flow cytometry experiments

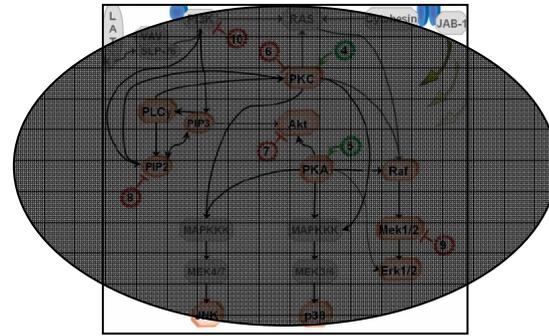


data

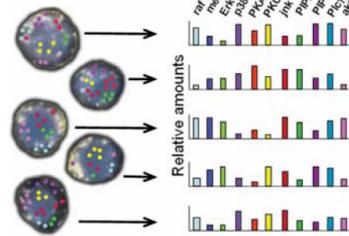
Here: Concentrations of (phosphorylated) proteins



possibly completely unknown



E.g.: Flow cytometry experiments

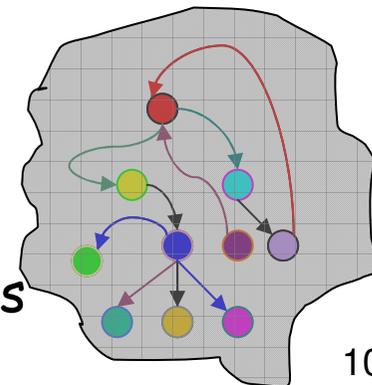


data data



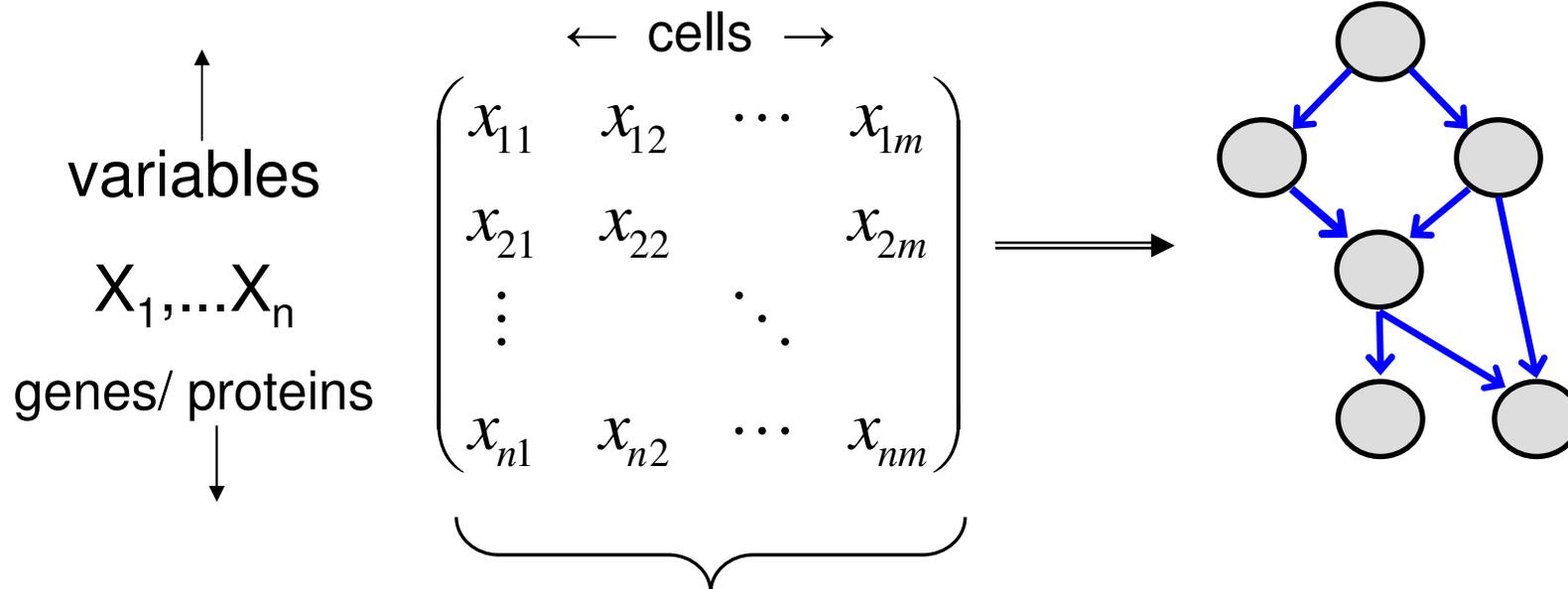
Machine Learning

statistical methods



# Statistical Task

Extract a network from a data matrix

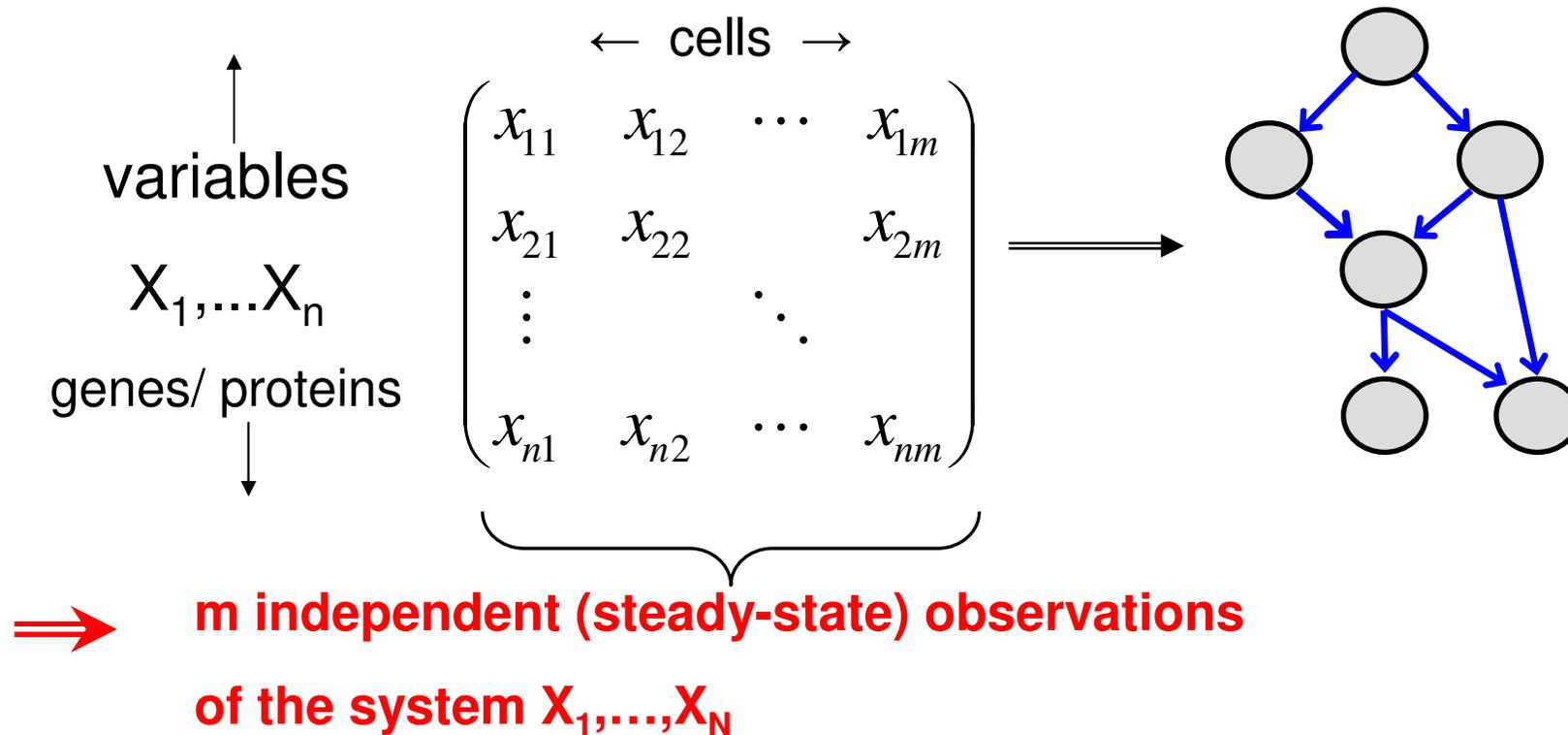


Either  $m$  independent (steady-state) observations  
of the system  $X_1, \dots, X_N$

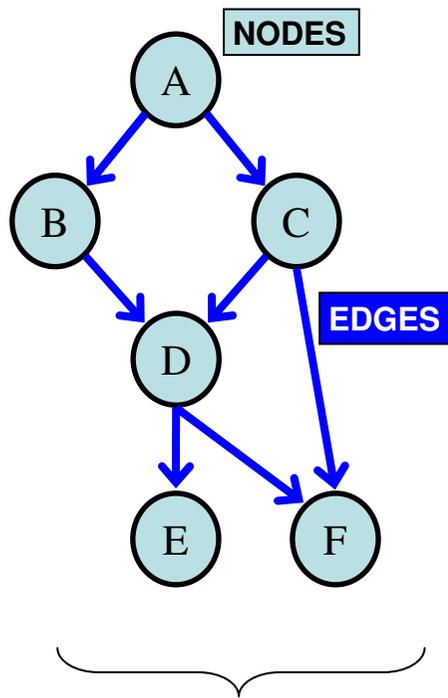
Or time series of the system of length  $m$ :  $(X_1, \dots, X_n)_{t=1, \dots, m}$

# Statistical Task

Extract a network from a data matrix



# Static Bayesian networks

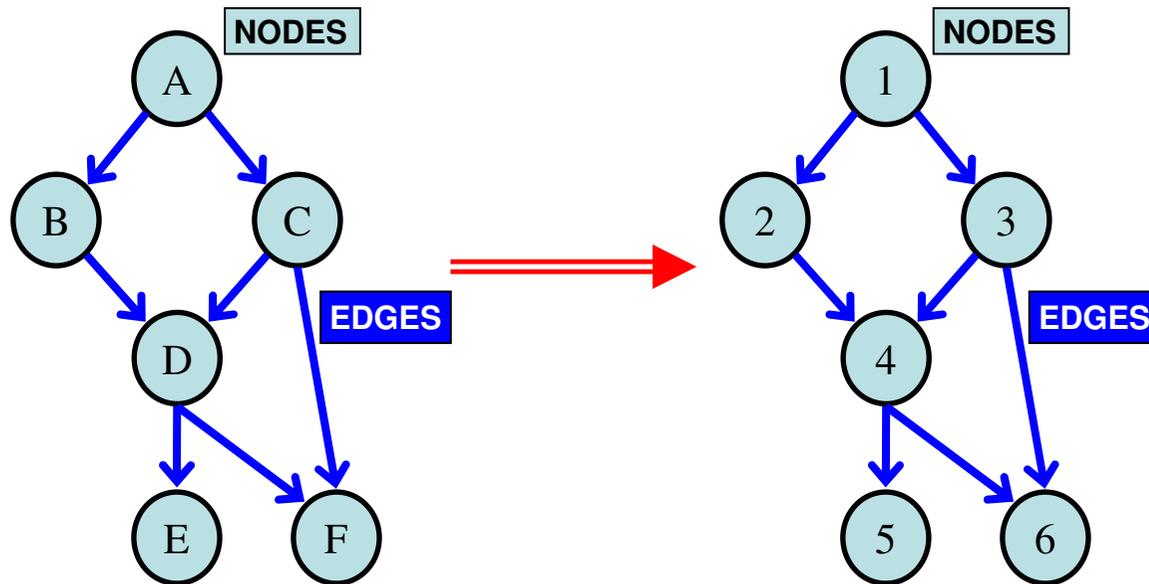


- Marriage between **graph theory** and **probability theory**.
- Directed acyclic graph (**DAG**) represents conditional independence relations.
- **Markov assumption** leads to a factorization of the joint probability distribution:

$$P(A, B, C, D, E, F)$$

$$= P(A) \cdot P(B|A) \cdot P(C|A) \cdot P(D|B, C) \cdot P(E|D) \cdot P(F|C, D)$$

# Incidence Matrix of a DAG

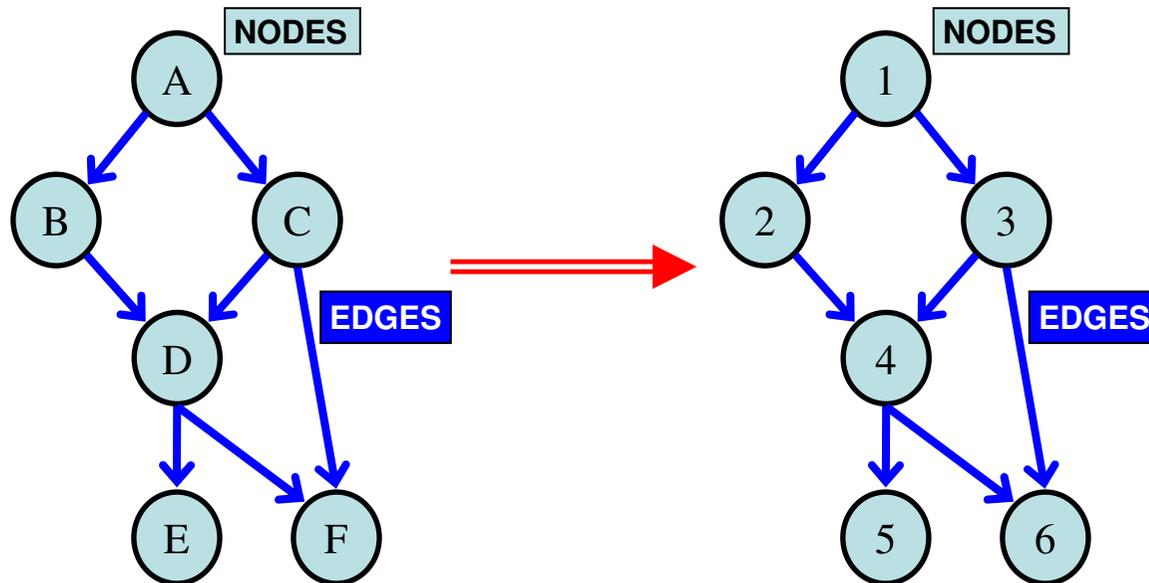


Number the variables/nodes  
1=A, 2=B, 3=C, etc.

$I(i,j)=1$  if there is an edge from node  $i$  to node  $j$   
 $I(i,j)=0$  if there is **no** edge from node  $i$  to node  $j$

$$I = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

# Ancestor Matrix of a DAG

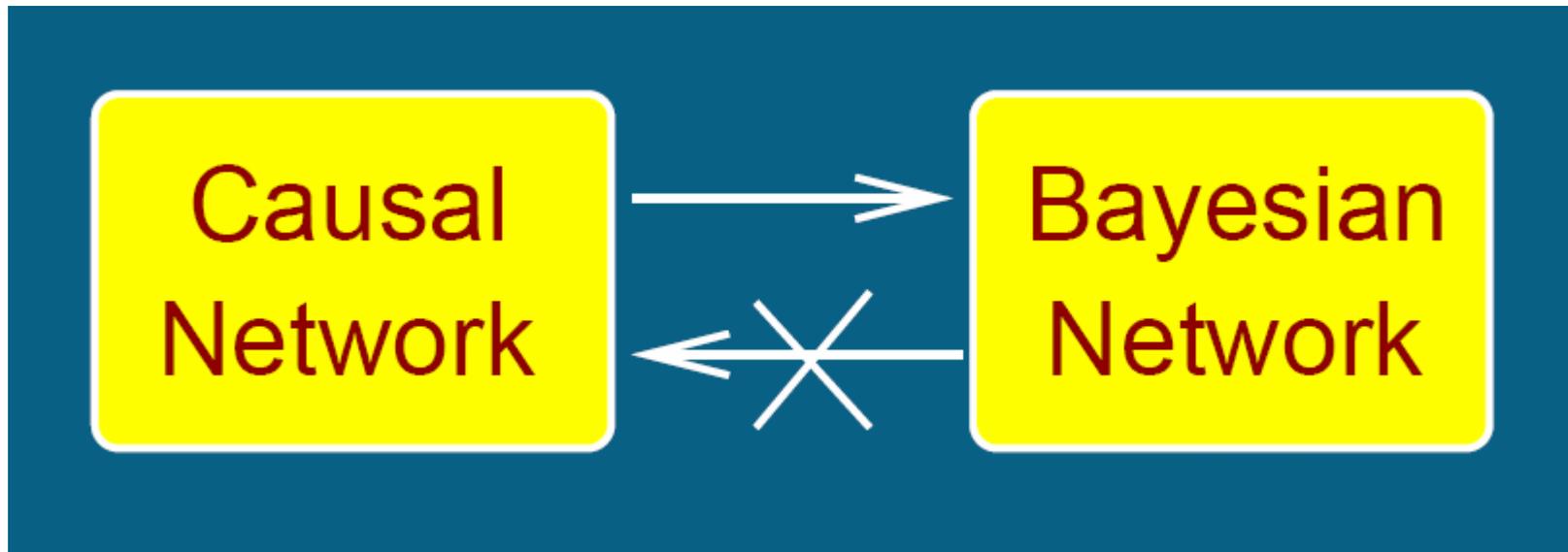


Number the variables/nodes  
1=A, 2=B, 3=C, etc.

$A(j,i)=1$  if there is a path from node  $i$  to node  $j$   
 $A(j,i)=0$  if there is no path from node  $i$  to node  $j$

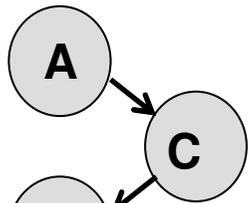
$$A = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 \end{pmatrix}$$

# Bayesian networks versus causal networks



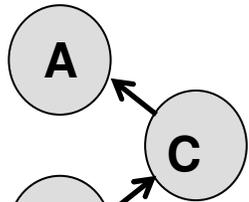
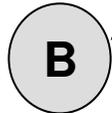
**Bayesian networks** represent conditional (in)dependency relations - **not** necessarily **causal interactions**.

# Equivalence classes of BNs



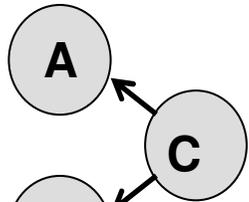
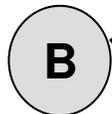
$$P(A) \cdot P(B|C) \cdot P(C|A)$$

$$= P(A) \cdot P(B, C) \cdot P(C)^{-1} \cdot P(C, A) \cdot P(A)^{-1}$$



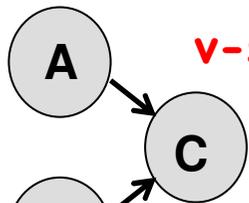
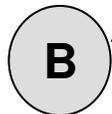
$$= P(C|B) \cdot P(B) \cdot P(C)^{-1} \cdot P(A|C) \cdot P(C)$$

$$= P(A|C) \cdot P(B) \cdot P(C|B)$$



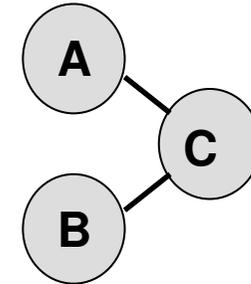
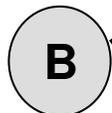
$$= P(A|C) \cdot P(B, C)$$

$$= P(A|C) \cdot P(B|C) \cdot P(C)$$



**v-structure**

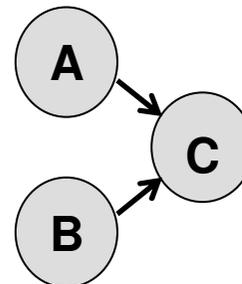
$$P(A) \cdot P(B) \cdot P(C|A, B)$$



$$P(A, B) \neq P(A) \cdot P(B)$$

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

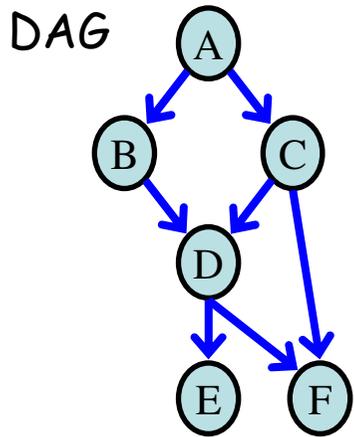
**completed partially  
directed graphs  
(CPDAGs)**



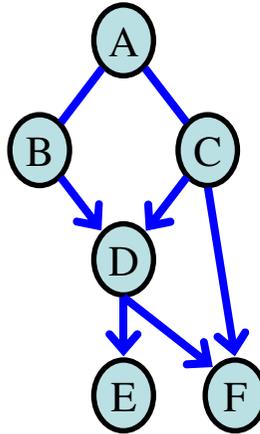
$$P(A, B) = P(A) \cdot P(B)$$

$$P(A, B|C) \neq P(A|C) \cdot P(B|C)$$

# CPDAG representations

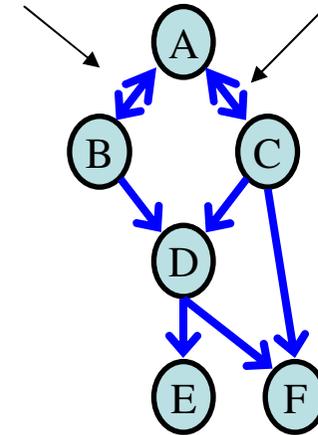


CPDAG



interpretation

superposition



$$DAG = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$CPDAG = \begin{pmatrix} 0 & -1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$EDGES = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

# Static Bayesian networks

$$\begin{aligned} P(\text{graph} | \text{data}) &= \frac{P(\text{data} | \text{graph}) \cdot P(\text{graph})}{P(\text{data})} \propto P(\text{data} | \text{graph}) \cdot P(\text{graph}) \\ &= P(\text{graph}) \cdot \int P(\text{data}, \theta(\text{graph}) | \text{graph}) d\theta(\text{graph}) \end{aligned}$$

# Static Bayesian networks

$$\begin{aligned} P(\text{graph} | \text{data}) &= \frac{P(\text{data} | \text{graph}) \cdot P(\text{graph})}{P(\text{data})} \propto P(\text{data} | \text{graph}) \cdot P(\text{graph}) \\ &= P(\text{graph}) \cdot \int P(\text{data}, \theta(\text{graph}) | \text{graph}) d\theta(\text{graph}) \end{aligned}$$

**Parameterisation: Gaussian BGe scoring metric:**

$$\text{data} \sim N(\mu, \Sigma)$$

with the (conjugate) normal-Wishart distribution for the parameters

$$\mu \sim N(\mu^*, (vW)^{-1}) \text{ and } W \sim \text{Wishart}(T_0)$$

# Static Bayesian networks

$$P(\text{graph} | \text{data}) = \frac{P(\text{data} | \text{graph}) \cdot P(\text{graph})}{P(\text{data})} \propto P(\text{data} | \text{graph}) \cdot P(\text{graph})$$

$$= P(\text{graph}) \cdot \int P(\text{data}, \theta(\text{graph}) | \text{graph}) d\theta(\text{graph})$$

**BGe metric:** closed form solution

# Static Bayesian networks

$$P(\text{graph} \mid \text{data}) = \frac{P(\text{data} \mid \text{graph}) \cdot P(\text{graph})}{P(\text{data})} \propto P(\text{data} \mid \text{graph}) \cdot P(\text{graph})$$

$$= \underbrace{P(\text{graph})}_{\text{uniform distribution}} \cdot \underbrace{\int P(\text{data}, \theta(\text{graph}) \mid \text{graph}) d\theta(\text{graph})}_{\text{BGe metric: closed form solution}}$$

uniform distribution

BGe metric: closed form solution

$$= \text{score}_{\text{BGe}}(\text{graph} \mid \text{data})$$

# Learning the network/graph structure

graph  $\rightarrow$   $score_{BGe}(\text{graph})$

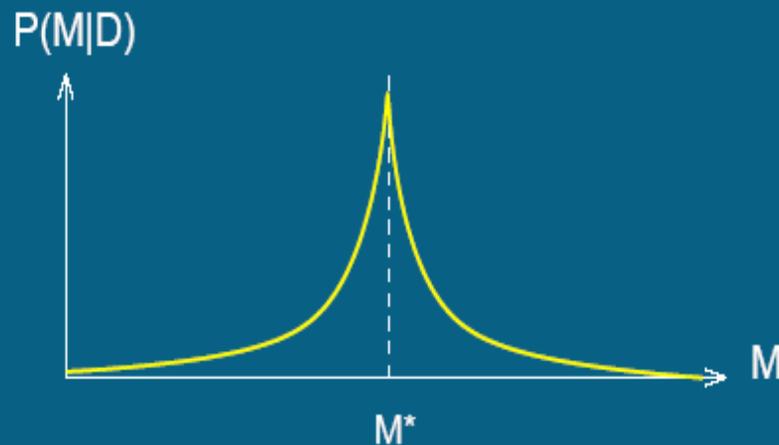
n	4	6	8	10
#DAGs	543	$3,7 \cdot 10^6$	$7,8 \cdot 10^{11}$	$4,2 \cdot 10^{18}$

**Idea:** Heuristically searching for the graph  $M^*$  that is most supported by the data  
 $P(M^* | \text{data}) > P(\text{graph} | \text{data})$ ,  
e.g.: greedy search algorithm

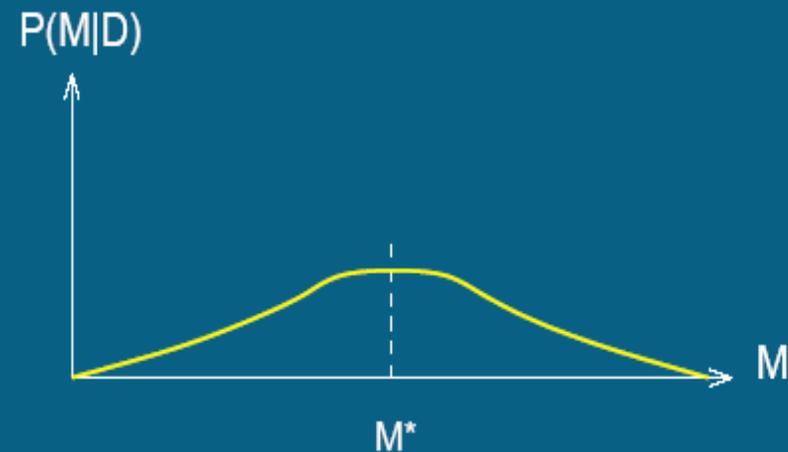
# Learning the network/graph structure

Distribution of  $P(\text{graph}|\text{data})$

Data are sparse  $\rightarrow$  Intrinsic uncertainty of inference



Large data set  $D$ :  
Best network structure  $M^*$  well defined



Small data set  $D$ :  
Intrinsic uncertainty about  $M^*$

# MCMC sampling of Bayesian networks

Better idea: Bayesian model averaging via **Markov Chain**

**Monte Carlo** (MCMC) simulations

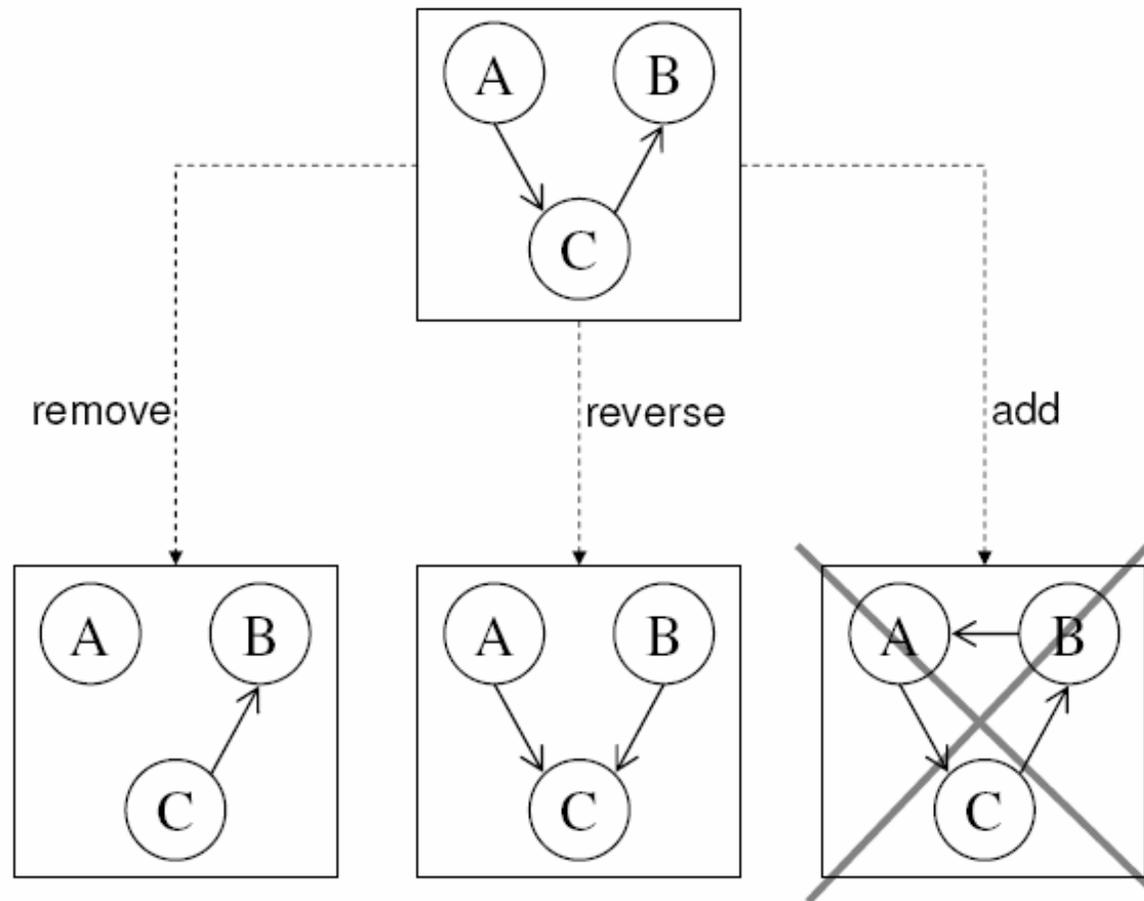
Construct and simulate a Markov Chain  $(M_t)_t$  in the space of DAGs whose distribution converges to the graph posterior distribution as stationary distribution, i.e.:

$$P(M_t = \text{graph} | \text{data}) \xrightarrow{t \rightarrow \infty} P(\text{graph} | \text{data})$$

to generate a DAG sample:  $G_1, G_2, G_3, \dots, G_T$

# Structure MCMC sampling scheme

(based on single edge operations)



# Metropolis Hastings sampler

A Metropolis Hastings MCMC sampling scheme consists of two parts.

- (i) Given a graph  $G_{old}$ , a new graph is proposed with a **proposal probability**  $Q(G_{new} | G_{old})$ .
- (ii) The new graph is accepted with an **acceptance probability**  $A(G_{new} | G)$ , or rejected otherwise.

In the **structure MCMC sampling scheme** a neighbour graph, that is a graph  $G_{new}$  that can be reached from  $G_{old}$  by one single edge operation, is randomly drawn from a discrete uniform distribution in the proposal move (i).

And in step (ii) the new graph is accepted with probability:

$$A(G_{new} | G_{old}) = \min \left\{ 1, \frac{P(D | G_{new}) P(G_{new})}{P(D | G_{old}) P(G_{old})} \cdot \frac{Q(G_{old} | G_{new})}{Q(G_{new} | G_{old})} \right\}$$

# Metropolis Hastings sampler

A Metropolis Hastings MCMC sampling scheme consists of two parts.

- (i) Given a graph  $G_{old}$ , a new graph is proposed with a **proposal probability**  $Q(G_{new} | G_{old})$ .
- (ii) The new graph is accepted with an **acceptance probability**  $A(G_{new} | G)$ , or rejected otherwise.

In the **structure MCMC sampling scheme** a neighbour graph, that is a graph  $G_{new}$  that can be reached from  $G_{old}$  by one single edge operation, is randomly drawn from a discrete uniform distribution in the proposal move (i).

And in step (ii) the new graph is accepted with probability:

$$A(G_{new} | G_{old}) = \min \left\{ 1, \underbrace{\frac{P(D | G_{new})}{P(D | G_{old})}}_{\text{Likelihood ratio}} \underbrace{\frac{P(G_{new})}{P(G_{old})}}_{\text{Prior ratio}} \cdot \underbrace{\frac{Q(G_{old} | G_{new})}{Q(G_{new} | G_{old})}}_{\text{Hastings ratio}} \right\}$$

# Metropolis Hastings sampler

A Metropolis Hastings MCMC sampling scheme consists of two parts.

- (i) Given a graph  $G_{old}$ , a new graph is proposed with a **proposal probability**  $Q(G_{new}|G_{old})$ .
- (ii) The new graph is accepted with an **acceptance probability**  $A(G_{new}|G)$ , or rejected otherwise.

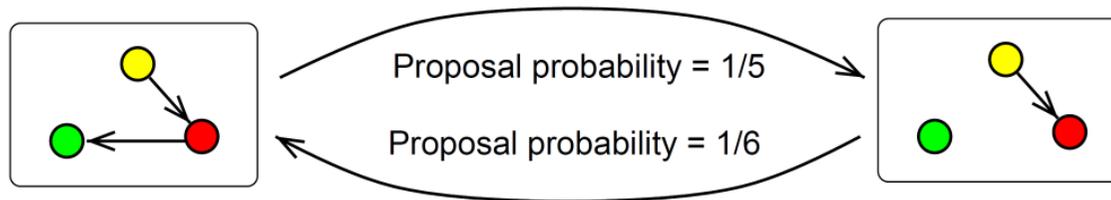
In the **structure MCMC sampling scheme** a neighbour graph, that is a graph  $G_{new}$  that can be reached from  $G_{old}$  by one single edge operation, is randomly drawn from a discrete uniform distribution in the proposal move (i).

And in step (ii) the new graph is accepted with probability:

$$A(G_{new} | G_{old}) = \min \left\{ 1, \underbrace{\frac{Score_{BGe}(G_{new})}{Score_{BGe}(G_{old})}}_{\text{Ratio of Scores}} \cdot \underbrace{\frac{Q(G_{old} | G_{new})}{Q(G_{new} | G_{old})}}_{\text{Hastings ratio}} \right\}$$

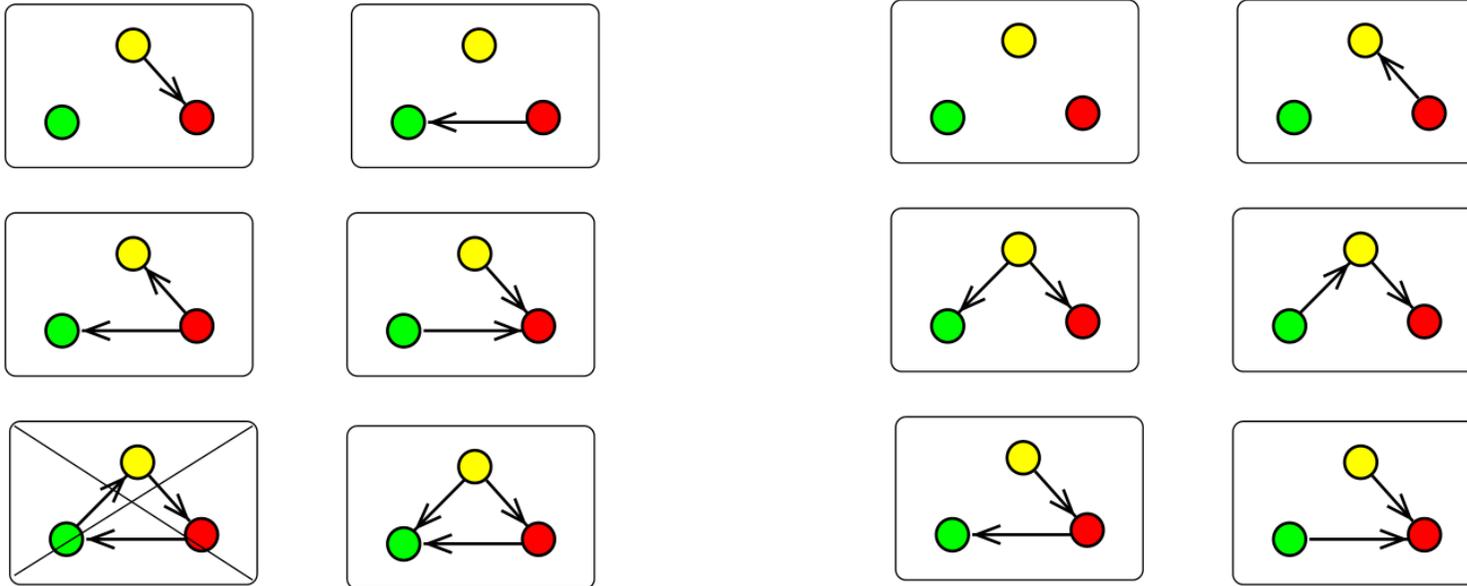
# Hastings ratio

$Q(G_i | G_{i-1}) \neq Q(G_{i-1} | G_i)$  is possible



Neighbourhood

Neighbourhood



# Metropolis Hastings algorithm structure MCMC for Bayesian networks

Initialisation: Start from an arbitrary initial graph  $G$   
(e.g. the empty-seeded graph) and set  $G_0=G$ .

Iteration: For  $i=1,\dots,T$

- Obtain a new graph  $G_i$  from the proposal distribution  $Q(G_i|G_{i-1})$
- Accept the new graph with probability  $A(G_i|G_{i-1})$  where  $A(.,.)$  has to be specified as described above; otherwise reject  $G_{i-1}$  leave the Markov chain state unchanged; symbolically:  $G_i=G_{i-1}$ .

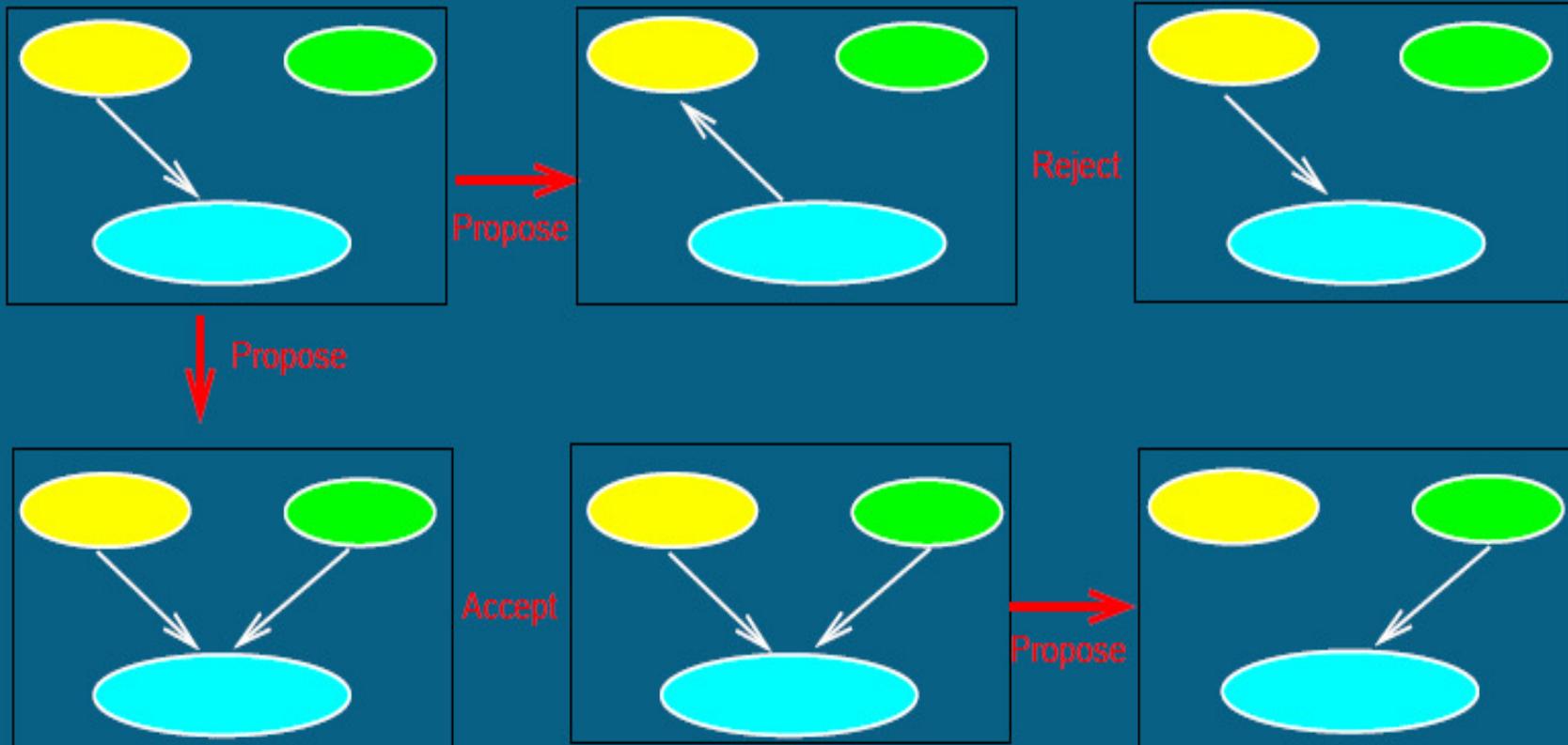
END

Discard an initial ,burn-in' period to allow the Markov chain to reach stationarity, i.e to converge. For example discard the first  $I < T$  MCMC samples.

Output: An MCMC sample from the posterior distribution  $P(G|D)$ , symbolically:

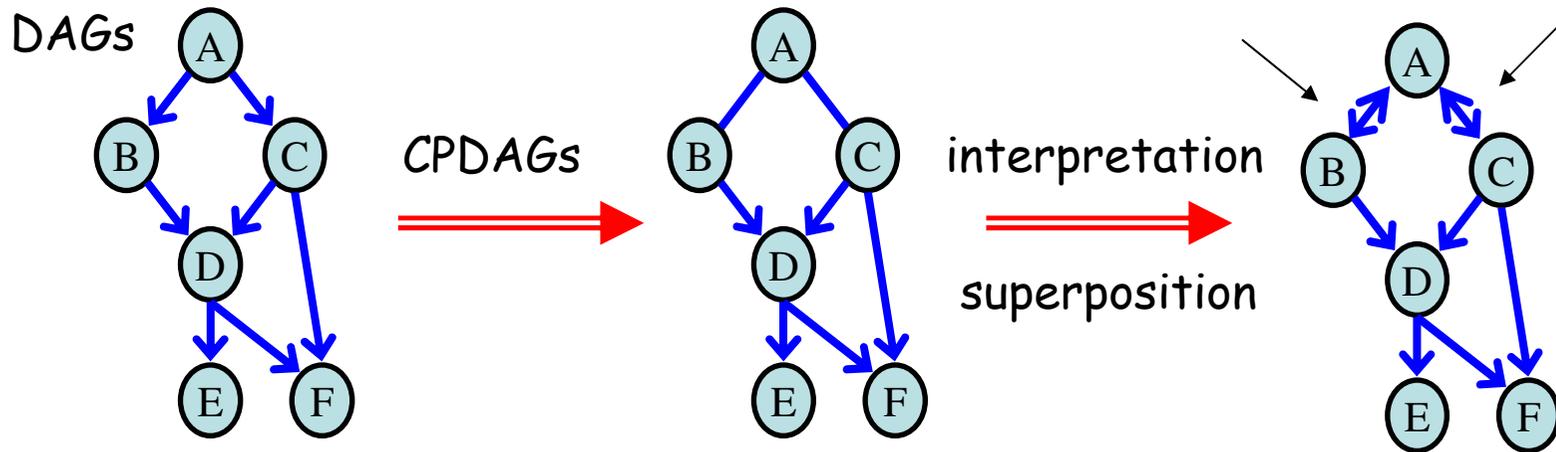
MCMC sample:  $G_{I+1}, \dots, G_T$

# Markov chain Monte Carlo (MCMC)



Acceptance probability:  $\min \left\{ 1, \frac{P(D|M_{new})}{P(D|M_{old})} \times \frac{P(M_{old})}{P(M_{new})} \times \frac{Q(M_{old}|M_{new})}{Q(M_{new}|M_{old})} \right\}$

# Marginal edge posterior probabilities



Use the DAG (CPDAG) sample for estimating the marginal posterior probability of „**directed edge relation features**“

$$\hat{P}(A \rightarrow B) = \frac{1}{(T - I)} \sum_{i=I+1}^T I(G_i)$$

where  $I(G_i)$  is 1 if the CPDAG of  $G_i$  contains the directed edge  $A \rightarrow B$ , and 0 otherwise

# Convergence of MCMC sampling

The DAG sample  $G_1, G_2, G_3, \dots, G_T$  is generated via Markov Chain Monte Carlo (MCMC) simulations so that the Markov Chain  $(M_t)_t$  converges to the graph posterior distribution:

$$P(M_t = \text{graph} | \text{data}) \rightarrow P(\text{graph} | \text{data})$$
$$t \rightarrow \infty$$

# Convergence of MCMC sampling

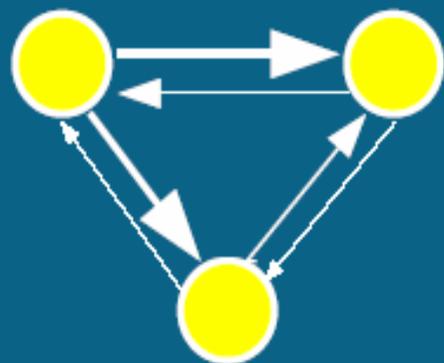
The DAG sample  $G_1, G_2, G_3, \dots, G_T$  is generated via Markov Chain Monte Carlo (MCMC) simulations so that the Markov Chain  $(M_t)_t$  converges to the graph posterior distribution:

$$P(M_t = \text{graph} | \text{data}) \rightarrow P(\text{graph} | \text{data})$$

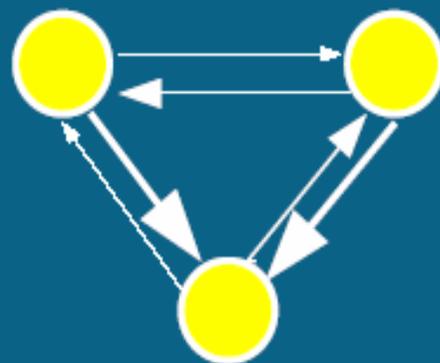
$$t \rightarrow \infty$$

In practice:  $t$  is not infinite!!!

MCMC simulation 1



MCMC simulation 2

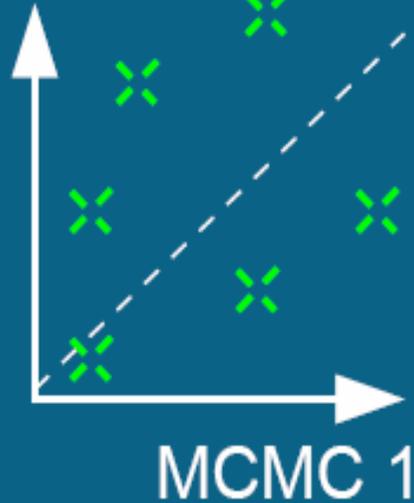


MCMC 2



T infinite

MCMC 2



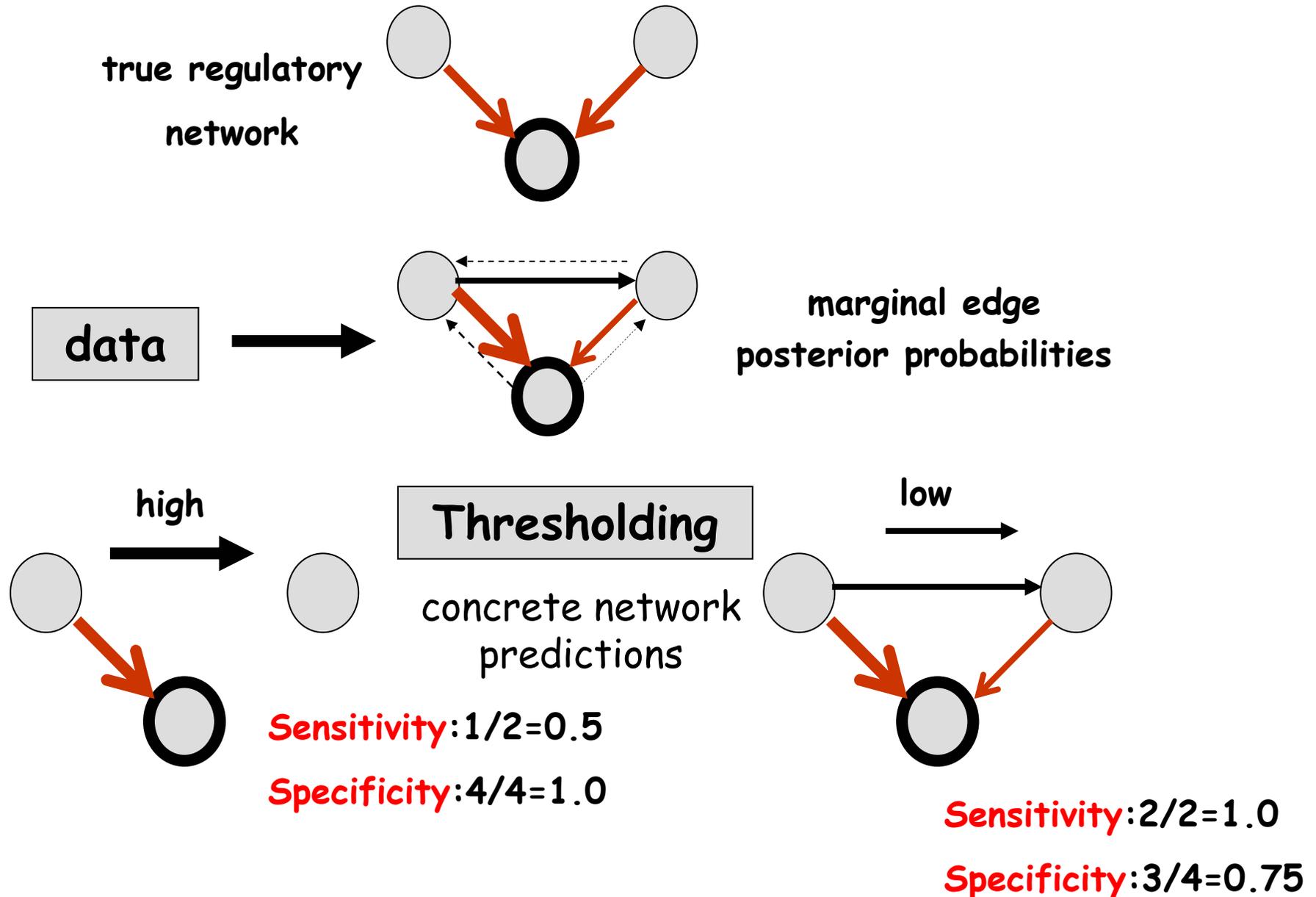
T too short

MCMC 2

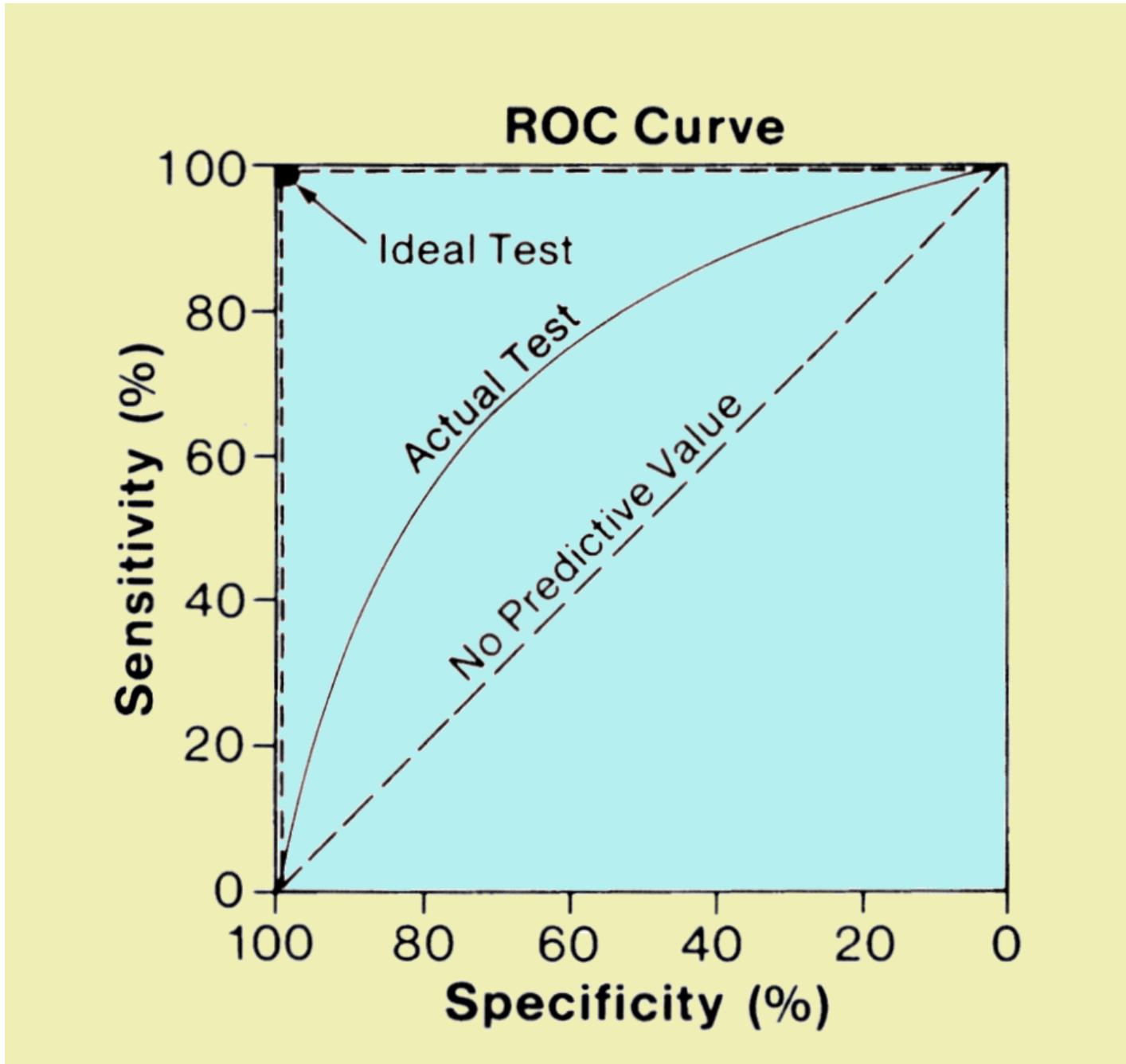


T long enough

# Network reconstruction accuracy

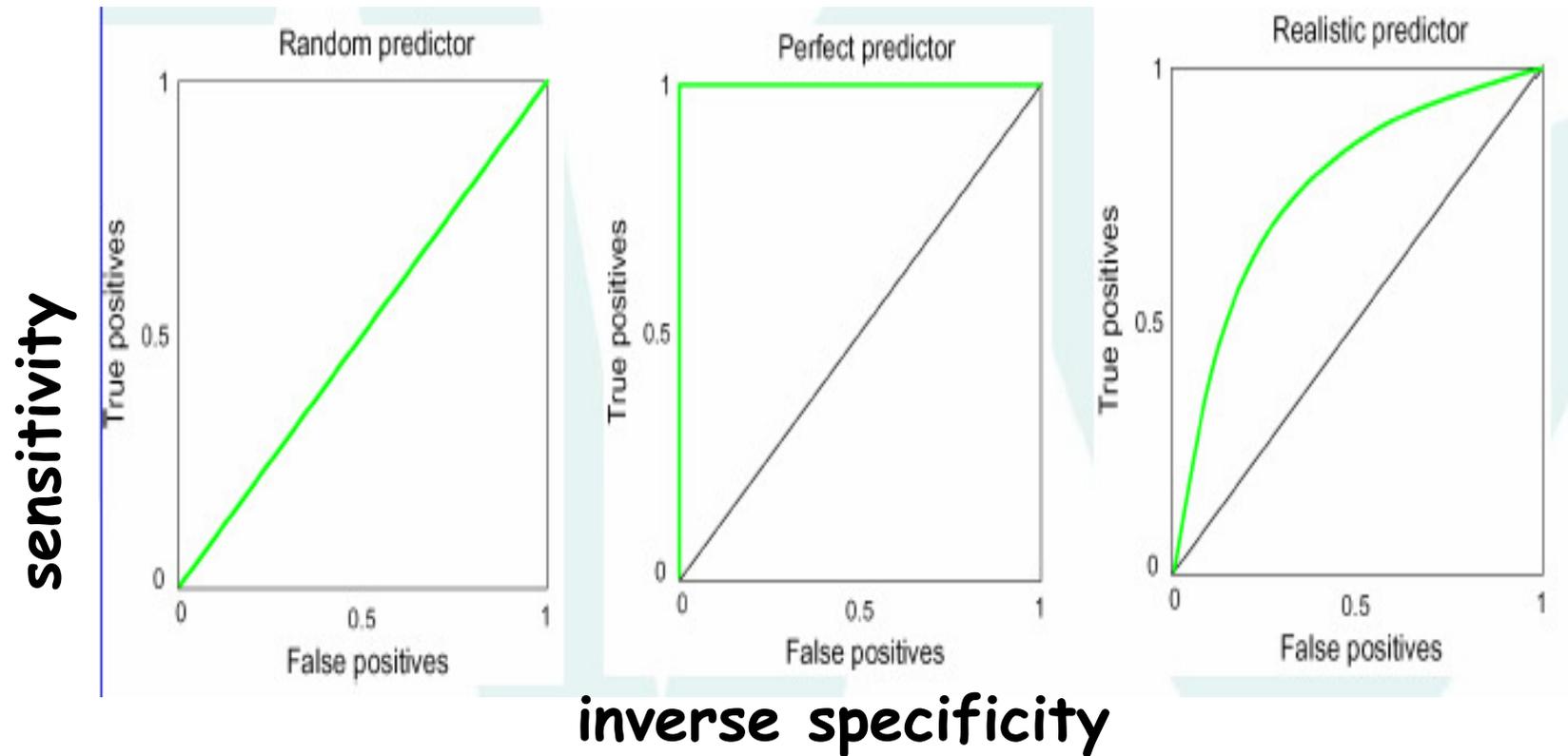


## Receiver Operator Characteristic (ROC) curve



# AUC scores

Area under Receiver Operator Characteristic (ROC) curve



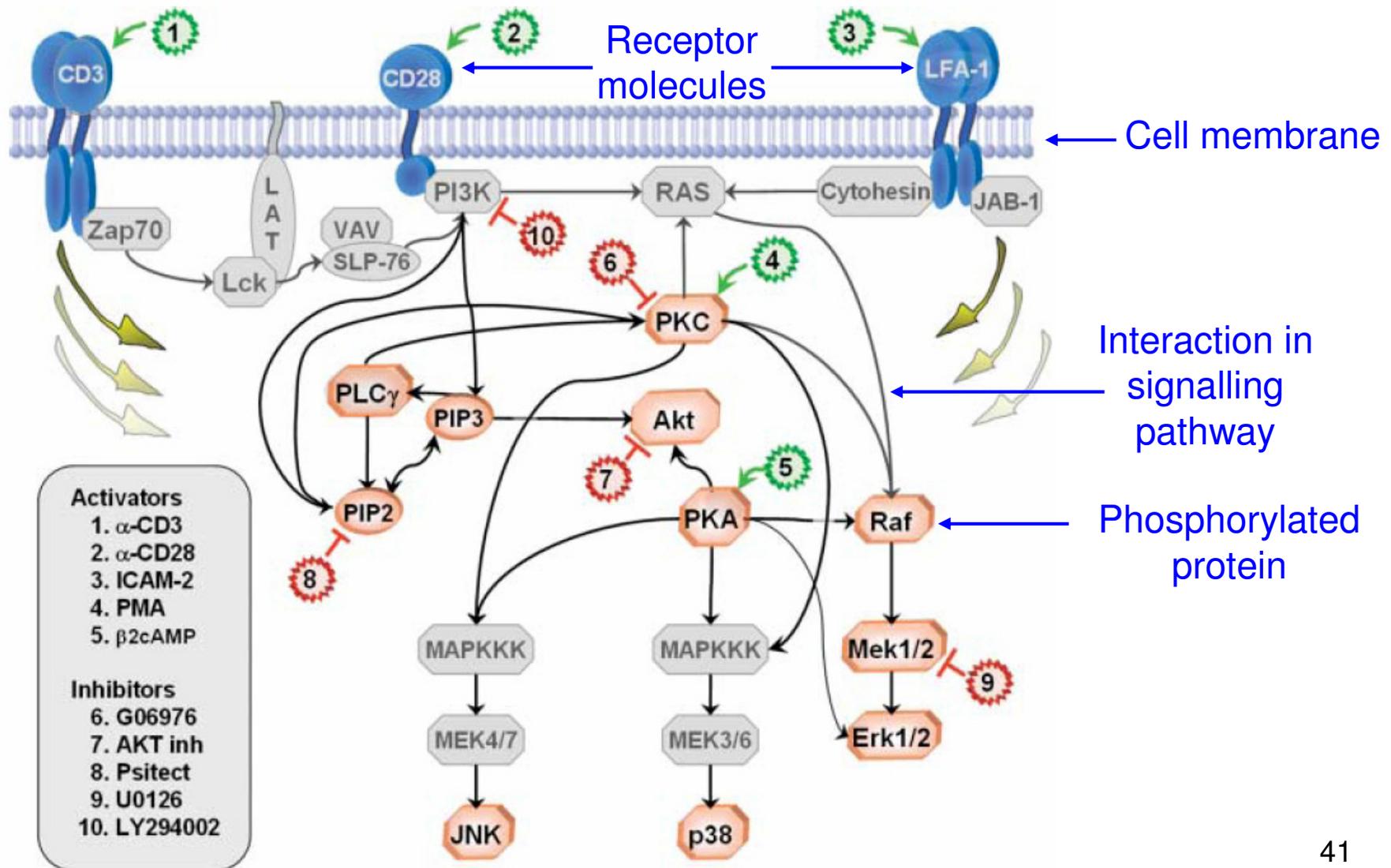
$AUC=0.5$

$AUC=1$

$0.5 < AUC \leq 1$

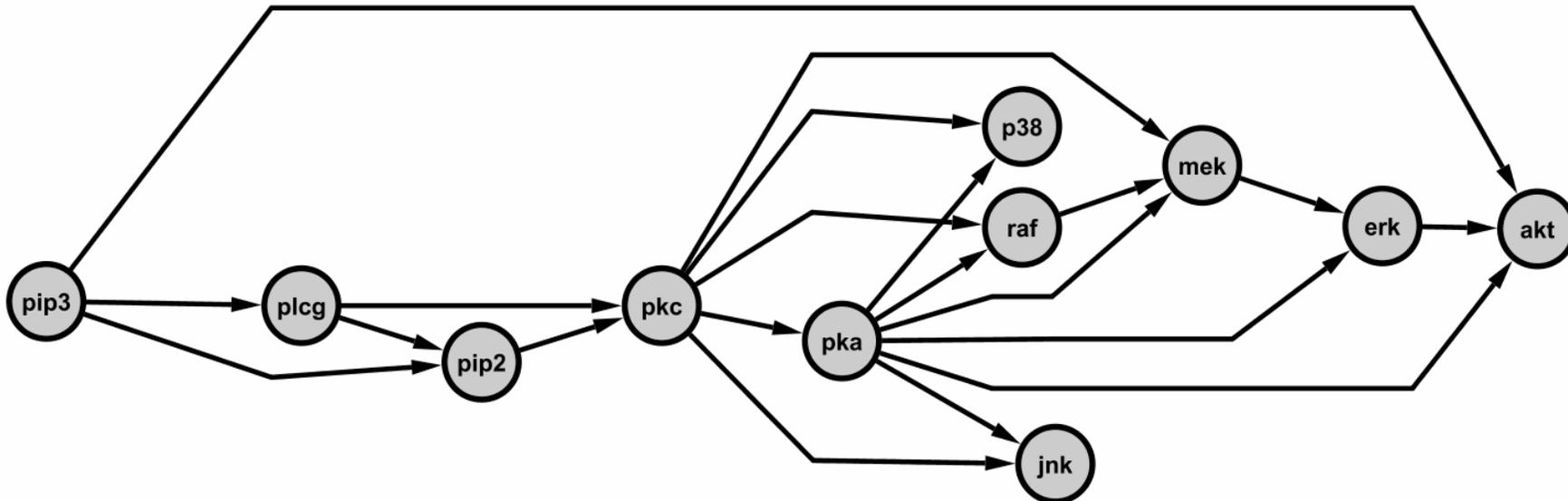
# Outlook to practical application

Gold-standard RAF pathway according to Sachs et al. (2004)



# Outlook

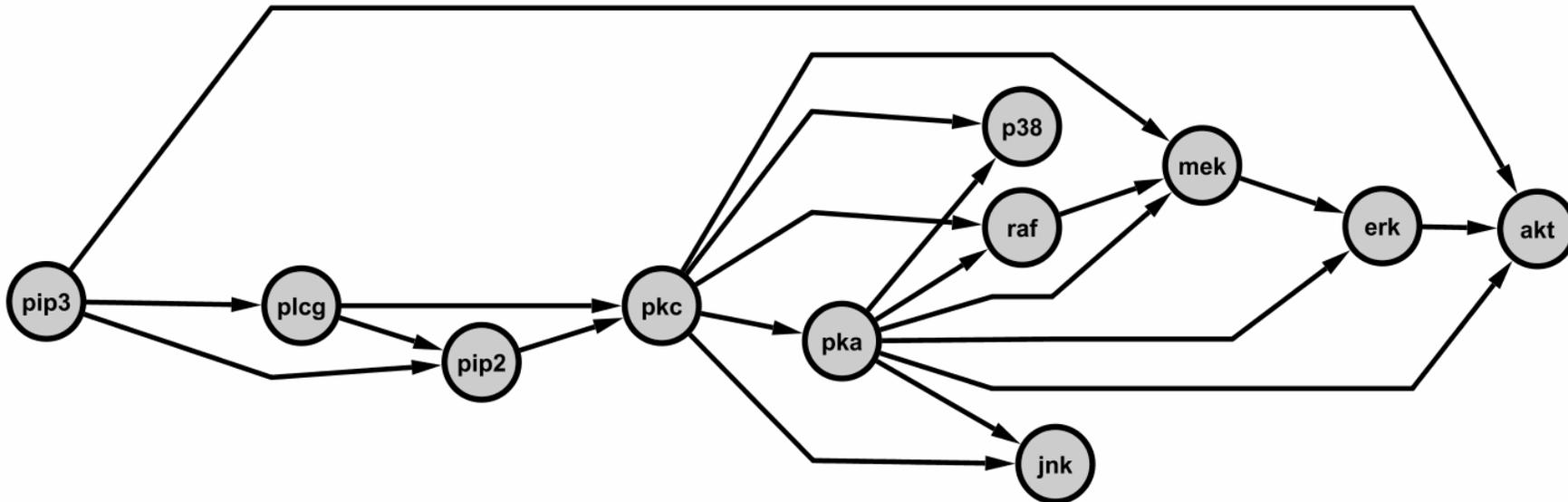
Take the RAF pathway topology



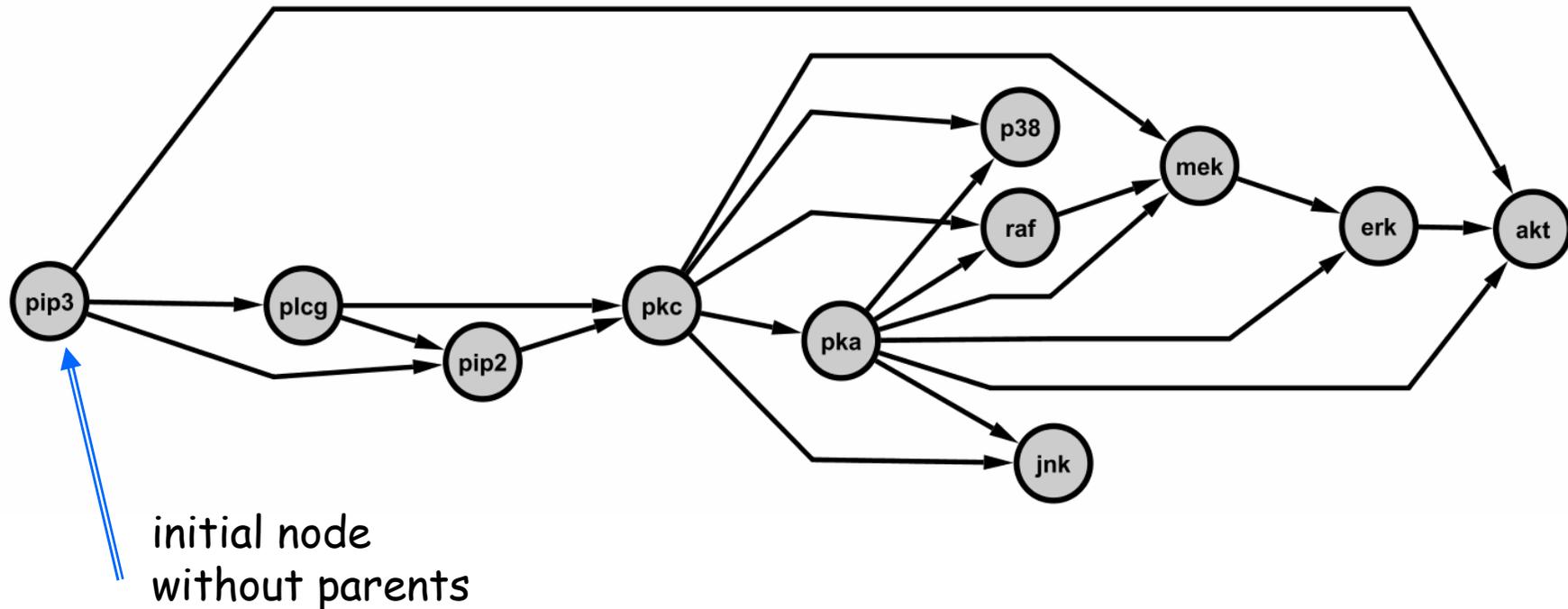
Cellular signalling cascade which consists of **11 phosphorylated proteins** and phospholipids in human immune systems cell  
(true network known from the literature)

# Outlook

Generate synthetic Gaussian network data



# Generate synthetic Gaussian network data

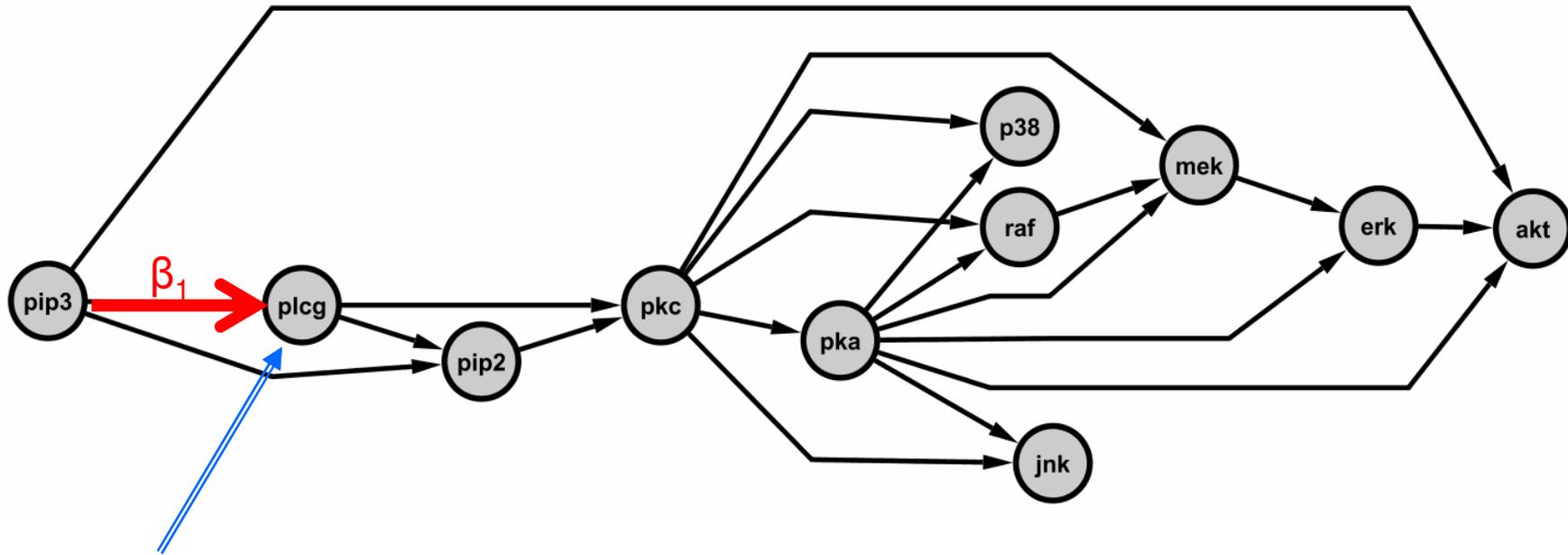


$$pip3 = \varepsilon_{pip3} \quad \text{where } \varepsilon_{pip3} \text{ is a Gaussian} \\ \text{with expectation } \mu=0 \text{ and variance } \sigma^2=1$$

We generate  $m$  independently and identically distributed (iid) realisations for  $pip3$ , and we standardise the  $m$  observations:

```
pip3 <- zscore(pip3):=(pip3-mean(pip3))/std(pip3)
```

# Generate synthetic Gaussian network data



Having sampled  $m$  realisations for parent node **pip3**, we sample the regression coefficient  $\beta_1$  from a uniform distribution on  $[0.5, 2]$  with a randomly drawn sign +/-

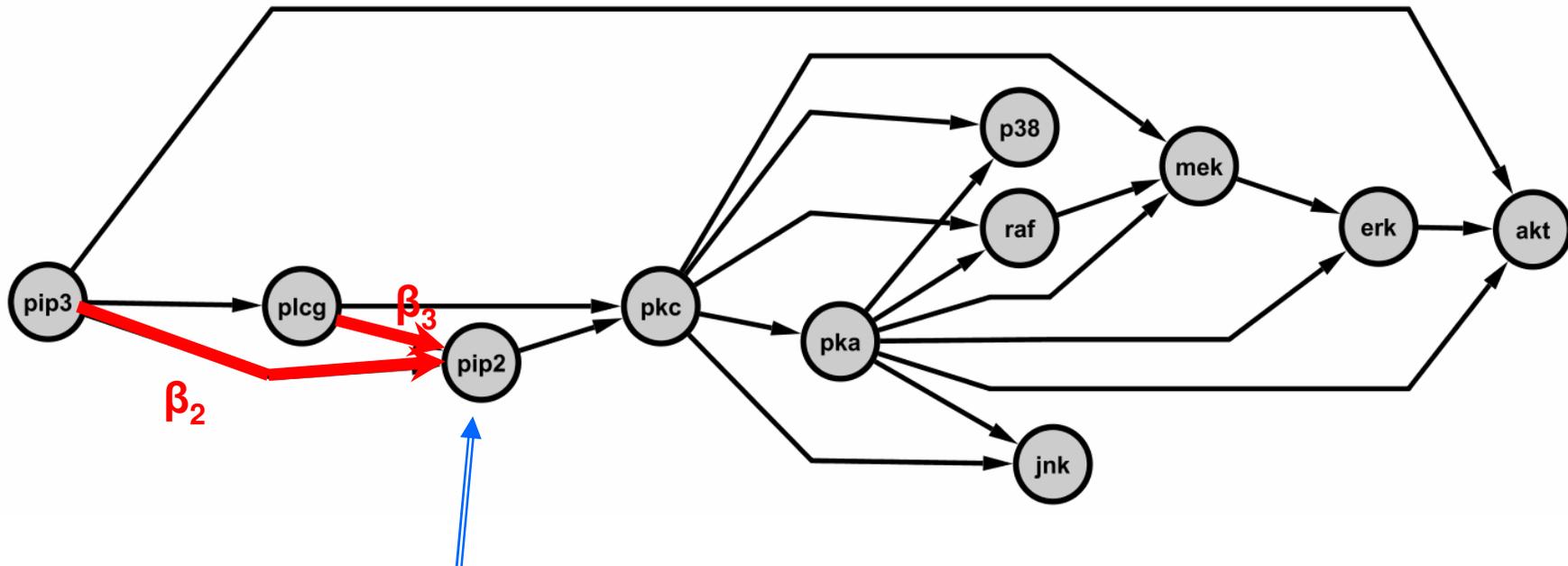
And we generate  $m$  iid realisations for **plcg** as follows :

$$plcg = \beta_1 \cdot pip3 + \varepsilon_{plcg}$$

where the noise term  $\varepsilon_{plcg}$  is a Gaussian with expectation  $\mu=0$  and variance  $\sigma^2$

Standardise the  $m$  values for **plcg**  $\leftarrow$  `zscore(plcg):=(plcg-mean(plcg))/std(plcg)`

# Generate synthetic Gaussian network data



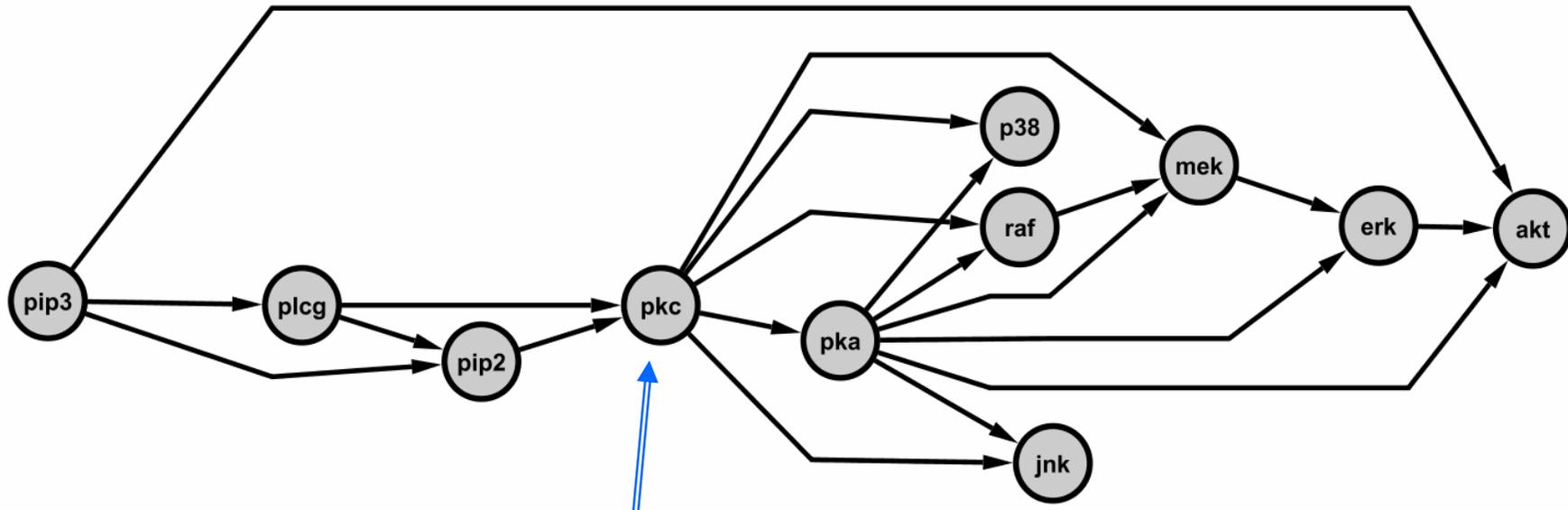
Having sampled  $m$  realisations for **pip3** and **plcg**, we sample both regression coefficients  $\beta_2$  and  $\beta_3$  from a uniform distribution on  $[0.5, 2]$  with randomly drawn signs +/-

And we generate  $m$  iid realisations for **pip2** as follows :

$$pip2 = \beta_2 \cdot pip3 + \beta_3 \cdot plcg + \epsilon_{pip2} \quad \text{where the noise term } \epsilon_{pip2} \text{ is a Gaussian with expectation } \mu=0 \text{ and variance } \sigma^2$$

Standardise the  $m$  values for **pip2** `pip2 <- zscore(pip2):=(pip2-mean(pip2))/std(pip2)`

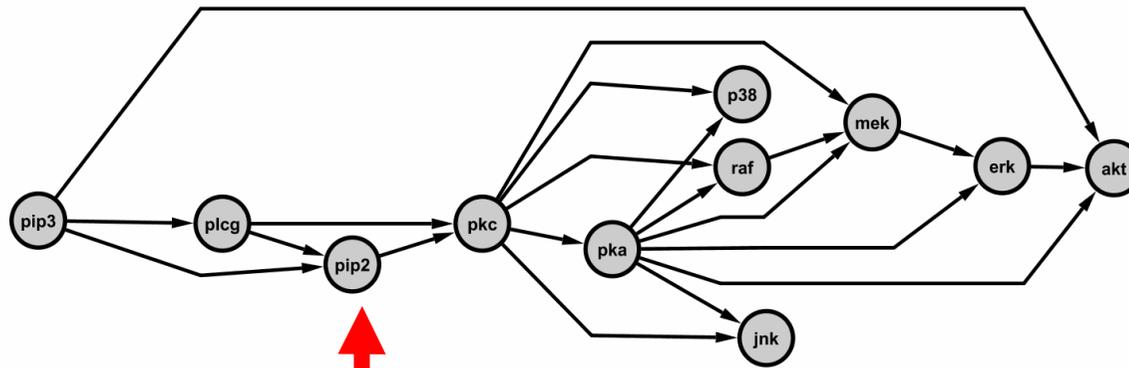
# Generate synthetic Gaussian network data



*We go to the next node, etc.*

Each node is described as a linear combination of its parent nodes. The regression coefficients are randomly sampled, and the noise terms are Gaussian distributed. We standardise the data to avoid that the signals become stronger and stronger. The parameter  $\sigma^2$  can be used to vary the signal-to-noise (SNR) ratio:

# Generate synthetic Gaussian network data

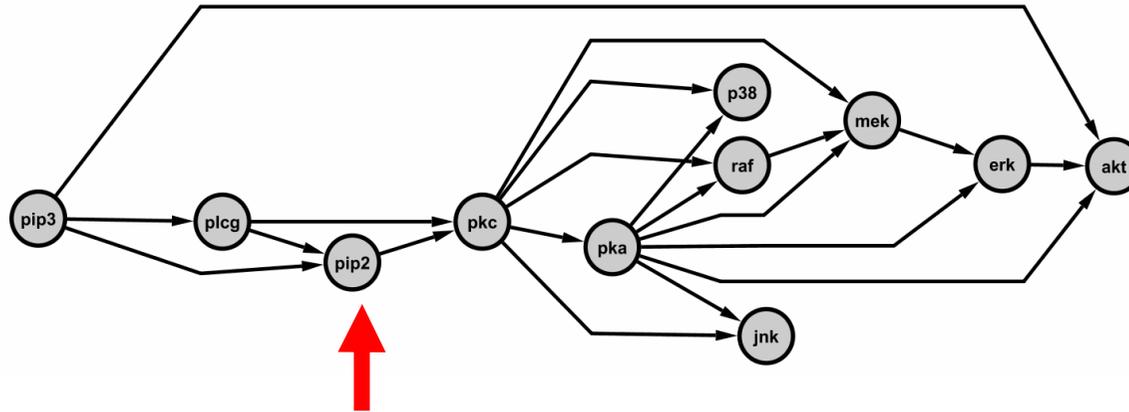


E.g.:  $pip2 = \beta_2 \cdot pip3 + \beta_3 \cdot plcg + \varepsilon_{pip2}$

signal  
from the parents

$$std(signal) = std(\underbrace{\beta_2 \cdot pip3}_{N(0,1)} + \underbrace{\beta_3 \cdot plcg}_{N(0,1)})$$

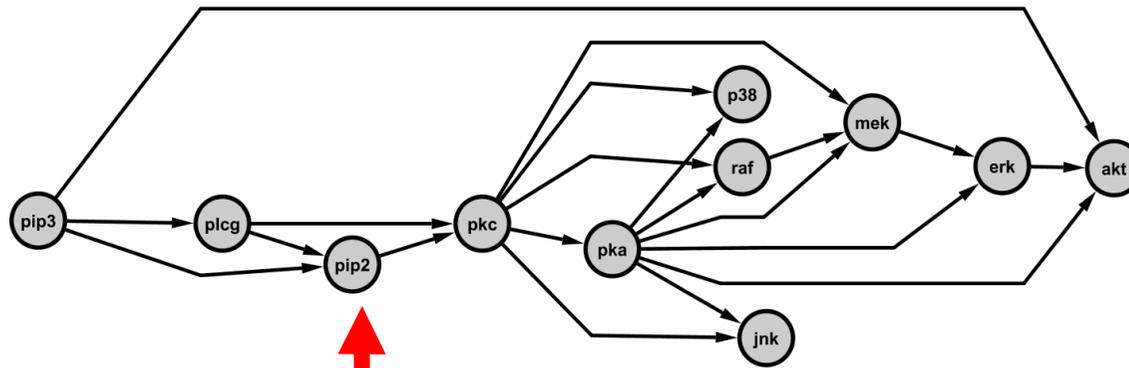
# Generate synthetic Gaussian network data



E.g.:  $pip2 = \beta_2 \cdot pip3 + \beta_3 \cdot plcg + \underbrace{\varepsilon_{pip2}}_{\text{noise}}$

$$std(noise) = std(\varepsilon_{pip3}) = \sigma$$

# Generate synthetic Gaussian network data



E.g.: 
$$pip2 = \underbrace{\beta_2 \cdot pip3 + \beta_3 \cdot plcg}_{\text{signal from the parents}} + \underbrace{\varepsilon_{pip2}}_{\text{noise}}$$

$$SNR = \frac{std(signal)}{std(noise)} = \frac{std(\beta_2 \cdot pip3 + \beta_3 \cdot plcg)}{\sigma}$$

Task: We will try to infer the  
Raf-pathway graph topology  
from a synthetically generated  
data set