Regulation Analysis using Restricted Boltzmann Machines

Network Modeling Seminar, 10/1/2013

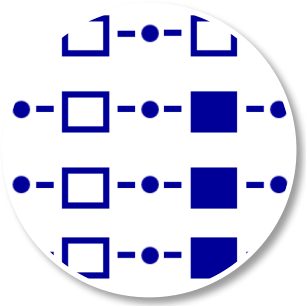
Patrick Michl

RUPRECHT-KARLS-
UNIVERSITÄT
HEIDELBERG

BioQuant
MODEL base of LIFE

dkfz.
GERMAN
CANCER RESEARCH CENTER
IN THE HELMHOLTZ ASSOCIATION

Agenda



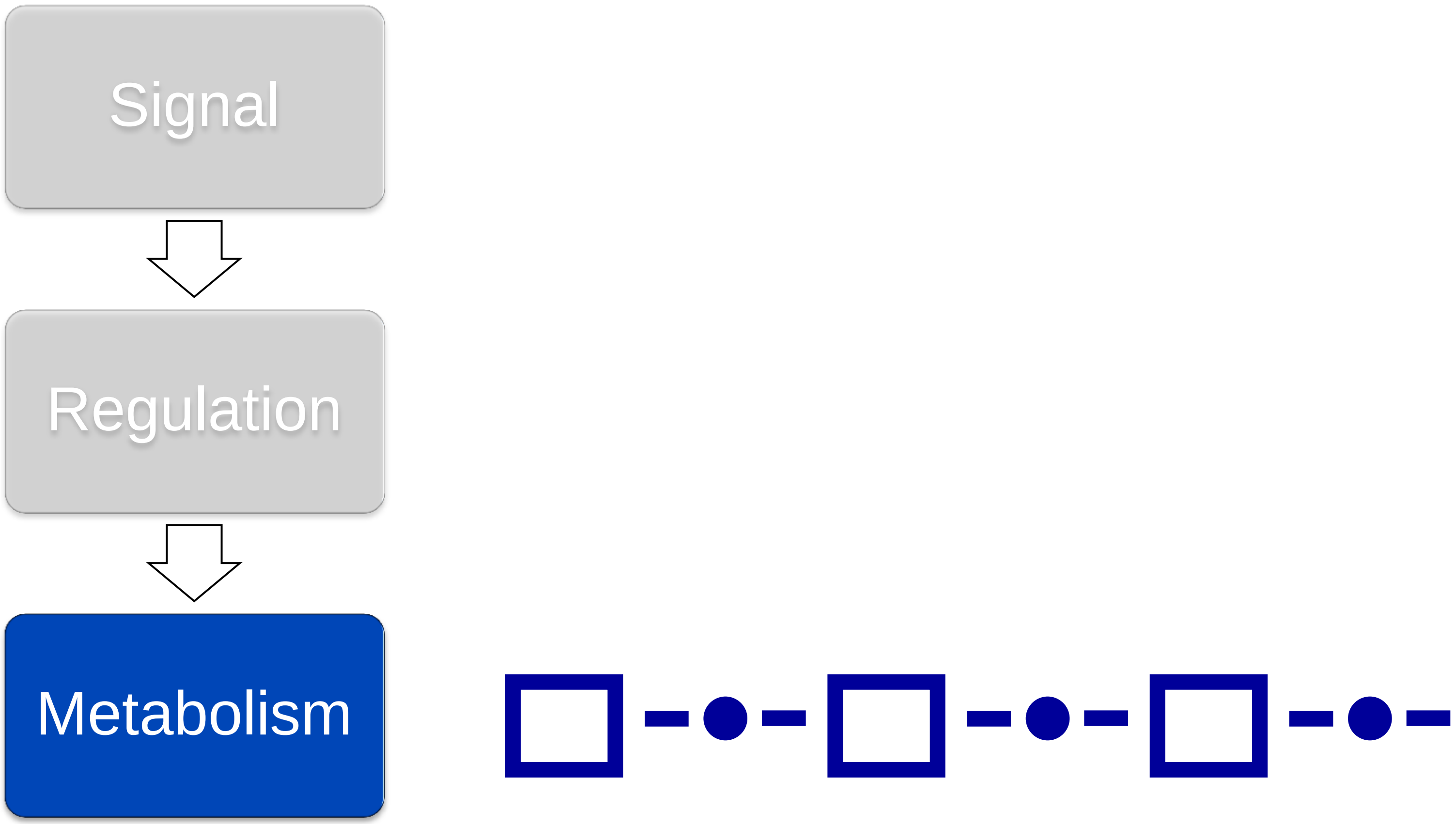
Biological Problem

Analysing the regulation of metabolism

Modeling

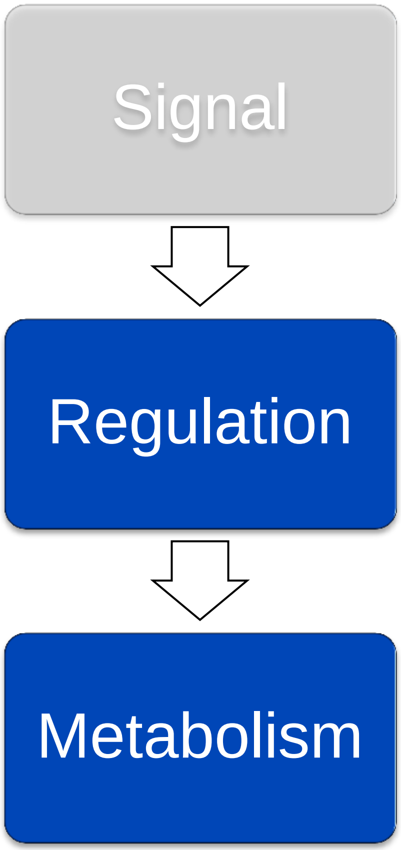
Implementation & Results

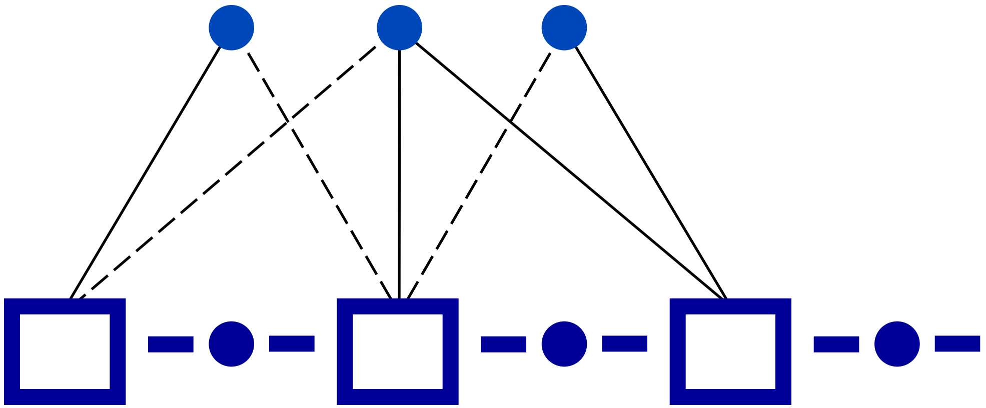
Biological Problem
Analysing the regulation of metabolism



A **linear metabolic pathway** of enzymes (E) …

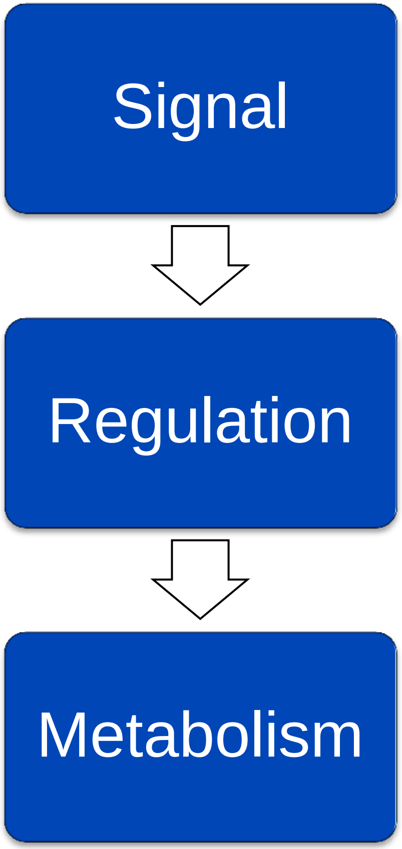
Biological Problem
Analysing the regulation of metabolism

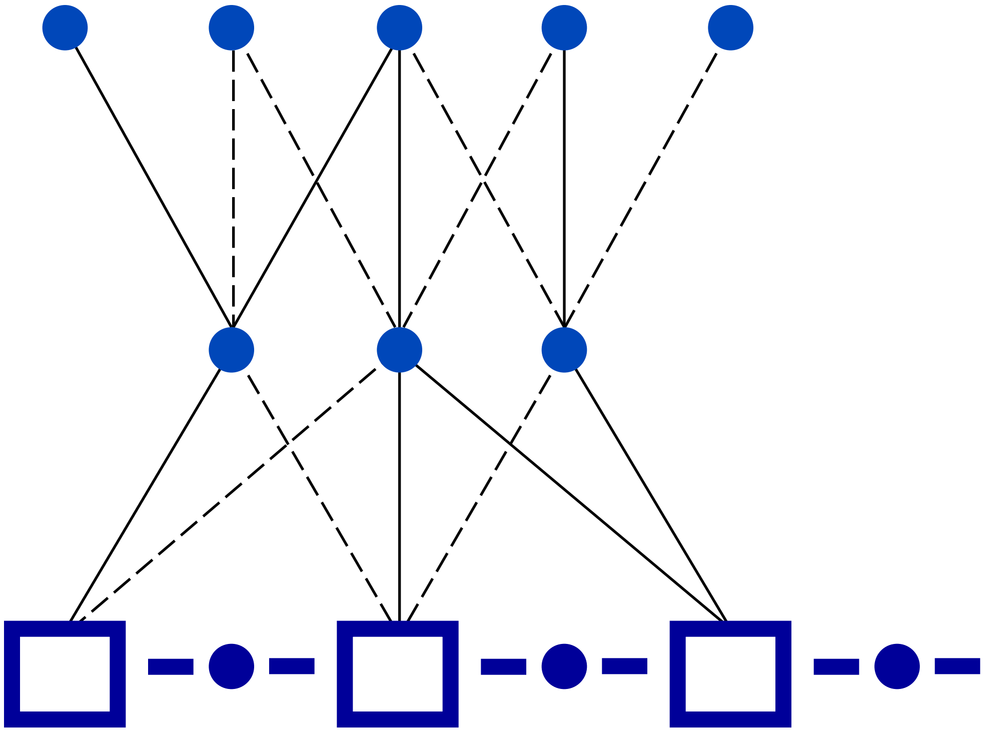




… is regulated by **transcription factors** (TF) …

Biological Problem
Analysing the regulation of metabolism





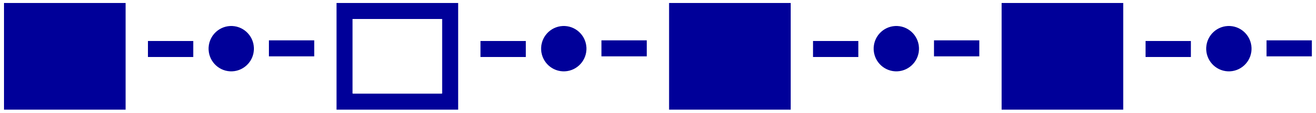
… which respond to **signals** (S)

Biological Problem
Analysing the regulation of metabolism

P 4

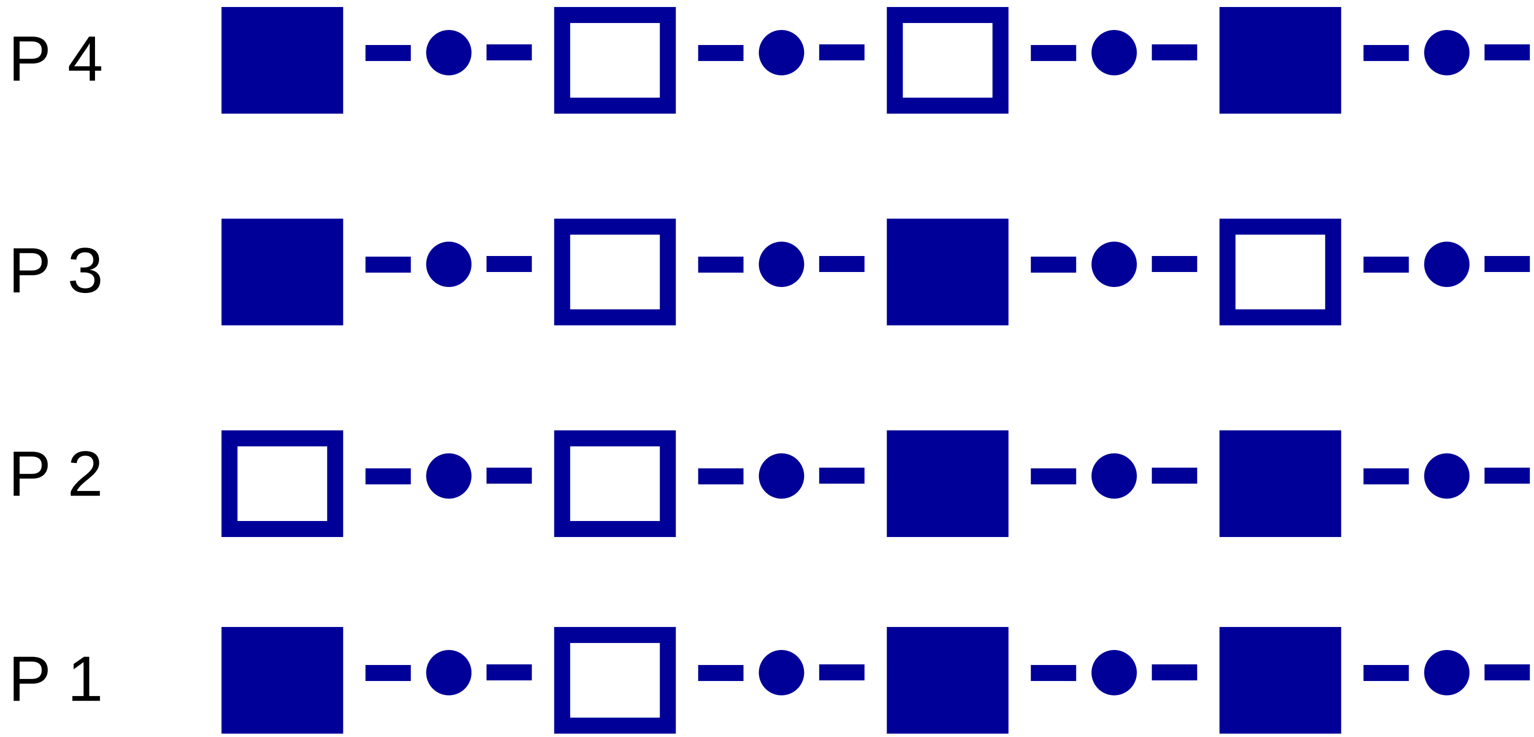
P 3

P 2

P 1 

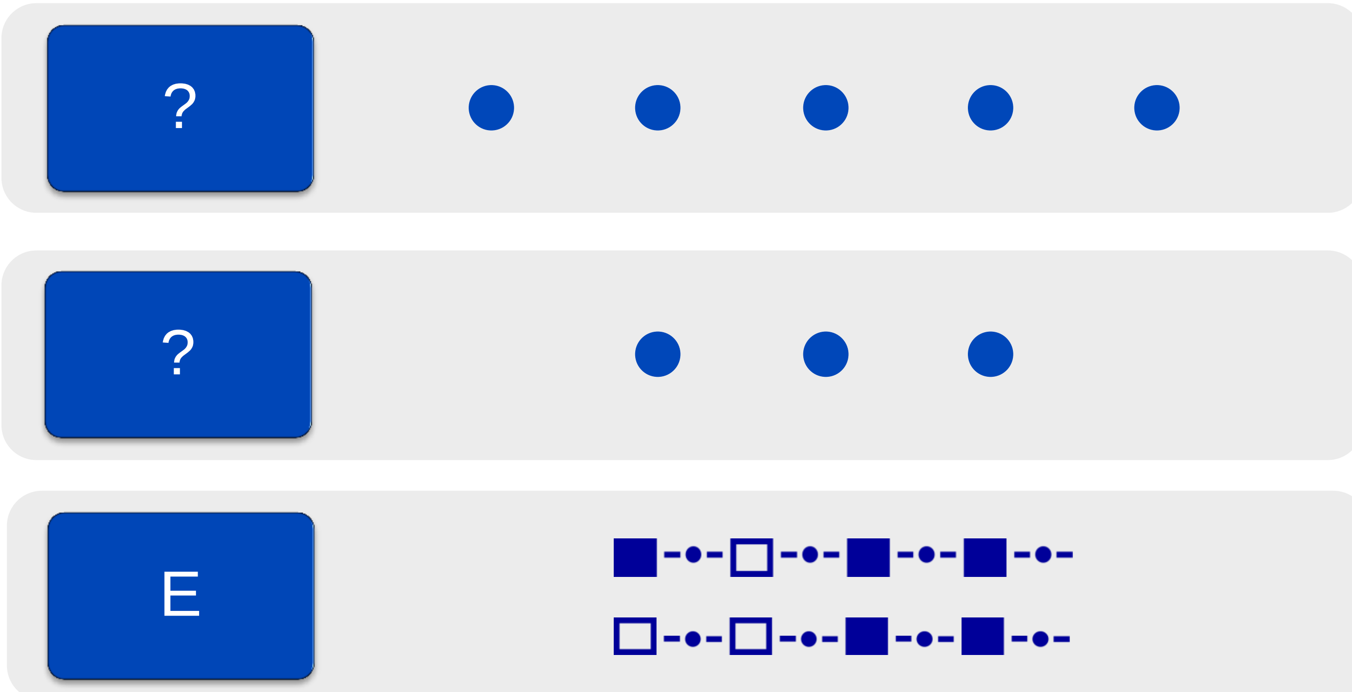
Upregulated linear pathways …

Biological Problem
Analysing the regulation of metabolism



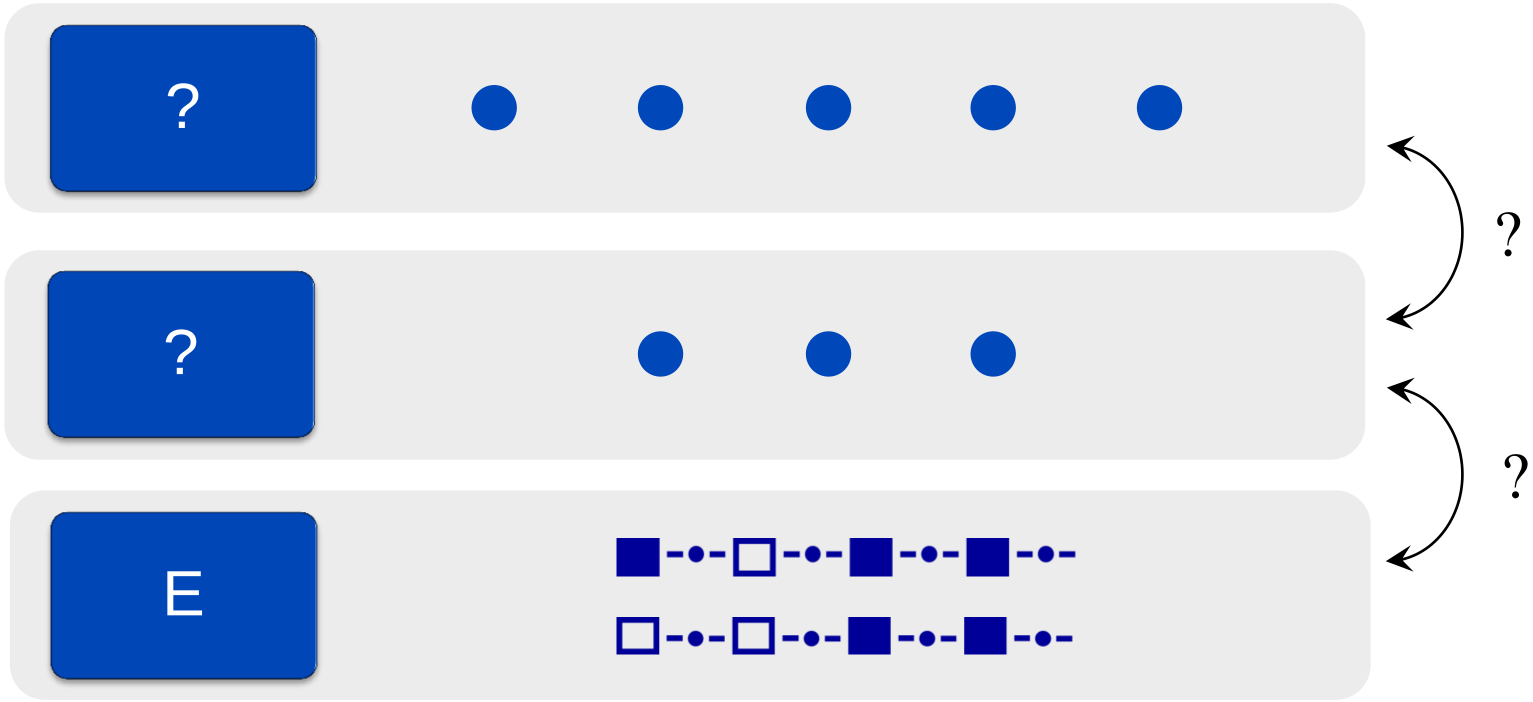
… can appear in **different patterns**

Biological Problem
Analysing the regulation of metabolism



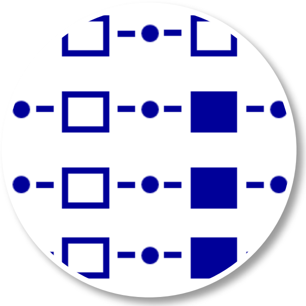
Which transcription factors and signals cause this patterns …

Biological Problem
Analysing the regulation of metabolism

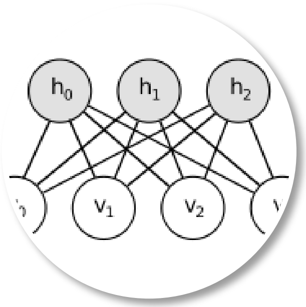


… and how do they interact? (topological structure)

Agenda



Biological Problem
Analysing the regulation of metabolism



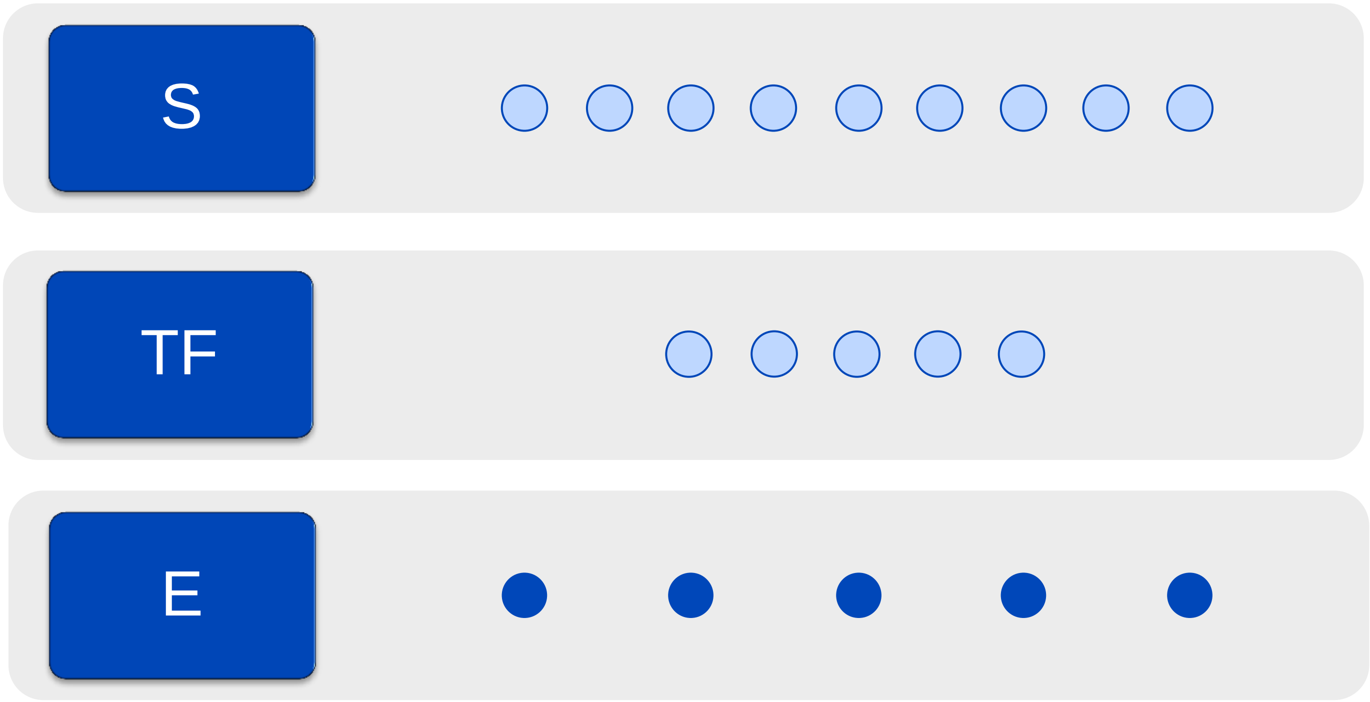
Network Modeling
Restricted Boltzmann Machines (RBM)

Validation & Implementation

Network Modeling
Restricted Boltzmann Machines (RBM)

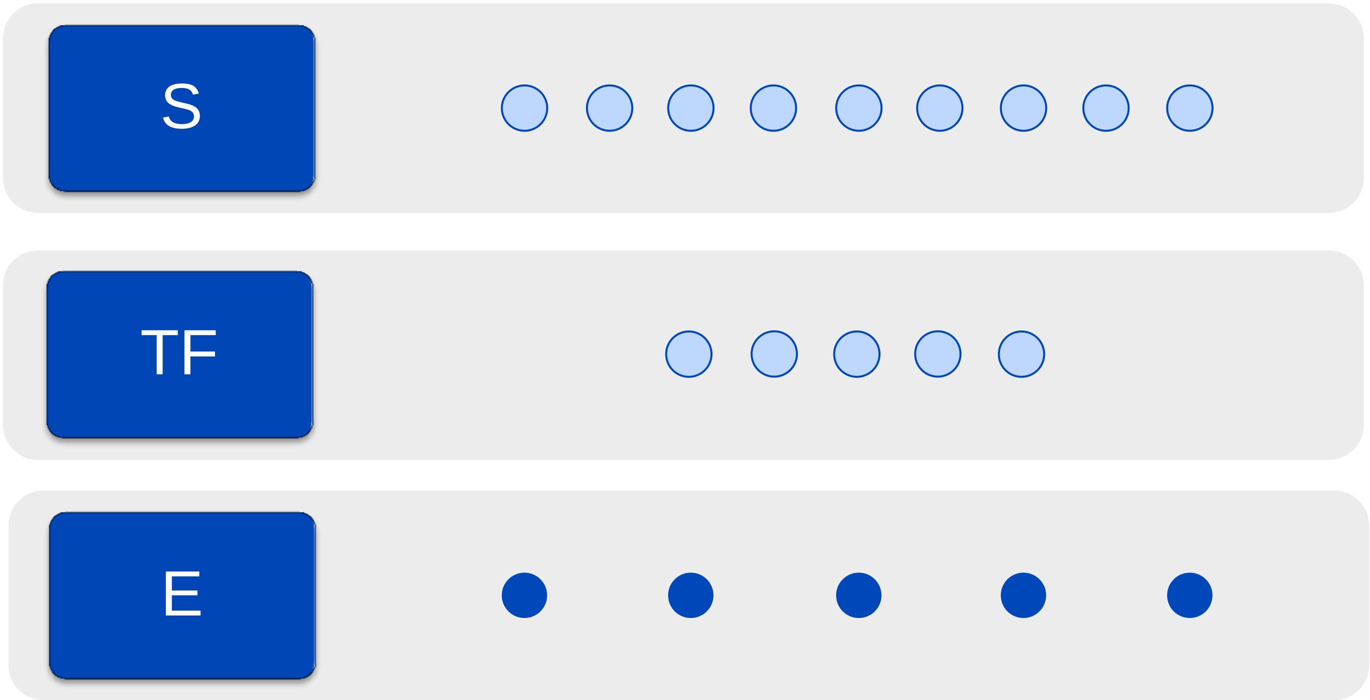
Lets start with some pathway of our interest …

Network Modeling
Restricted Boltzmann Machines (RBM)



… and lists of *interesting* TFs and *interesting* SigMols

Network Modeling
Restricted Boltzmann Machines (RBM)



How to model the topological structure?

Network Modeling
Restricted Boltzmann Machines (RBM)

Graphical Models

Graphical Models can preserve topological structures …

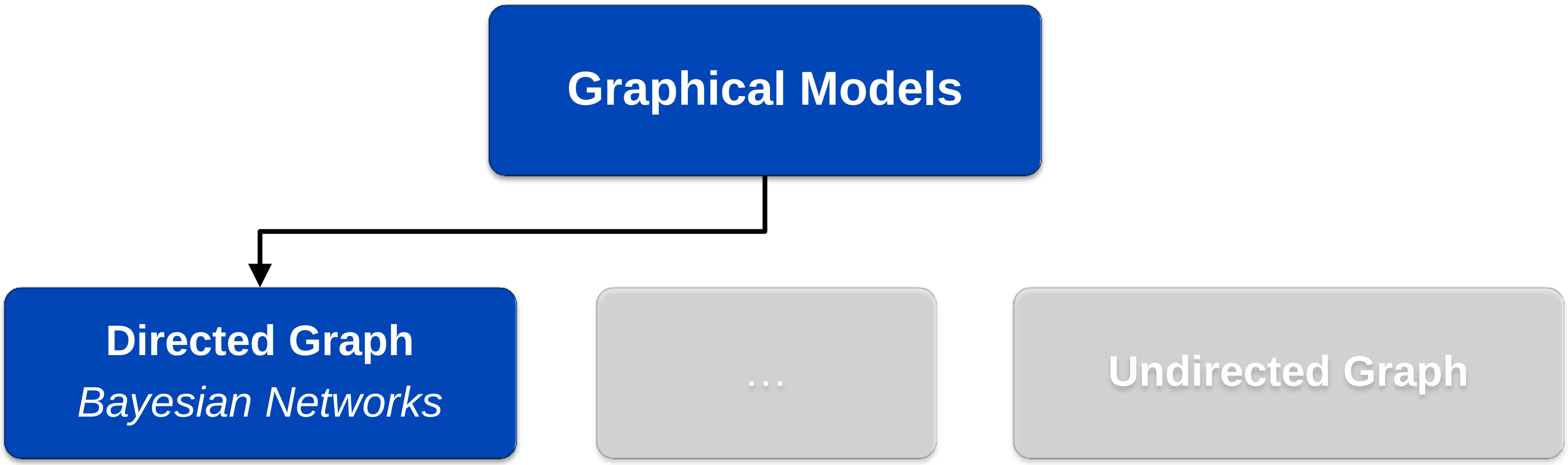
Network Modeling
Restricted Boltzmann Machines (RBM)

Graphical Models

Directed Graph | **…** | **Undirected Graph**

… but there are many types of graphical models

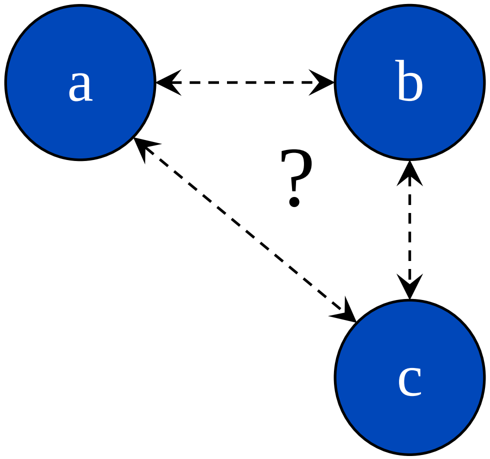
Network Modeling
Restricted Boltzmann Machines (RBM)



The most common type is the **Bayesian Network** (BN) …

Network Modeling
Restricted Boltzmann Machines (RBM)

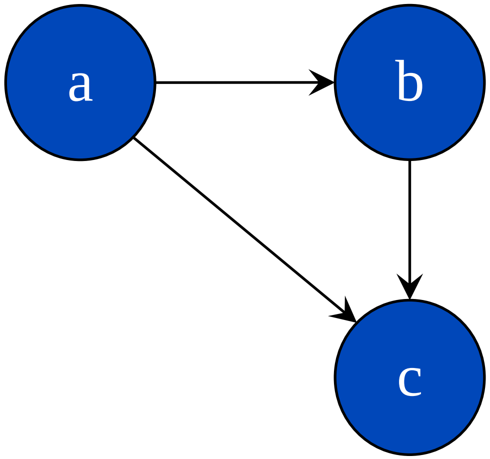
Bayesian Networks



a	b	c	P[a,b,c]
0	0	0	0.1
0	0	1	0.9
0	1	0	0.5
0	1	1	0.5
1	0	0	…
…	…	…	…

Bayesian Networks use **joint probabilities** …

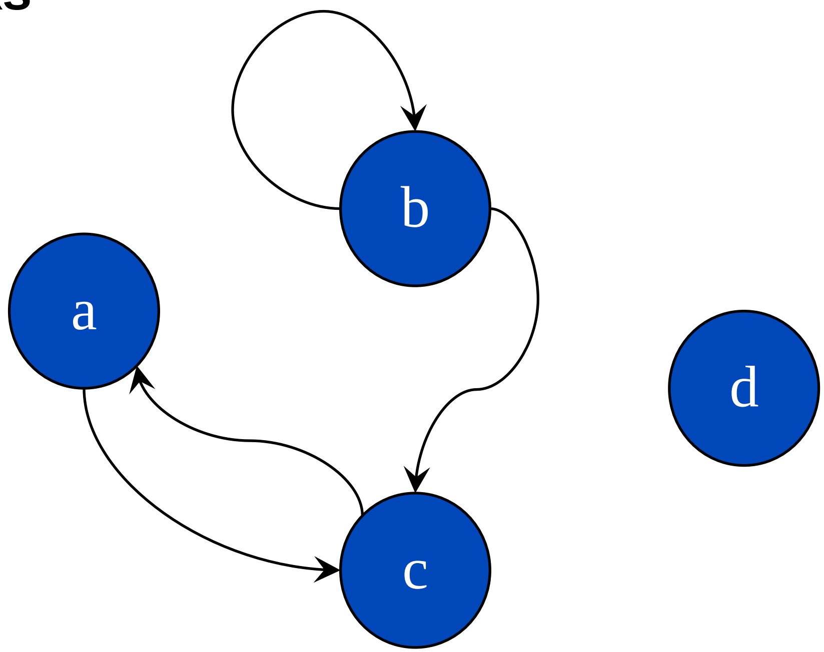
Bayesian Networks



a	b	c	P[a,b,c]
0	0	0	0.1
0	0	1	0.9
0	1	0	0.5
0	1	1	0.5
1	0	0	…
…	…	…	…

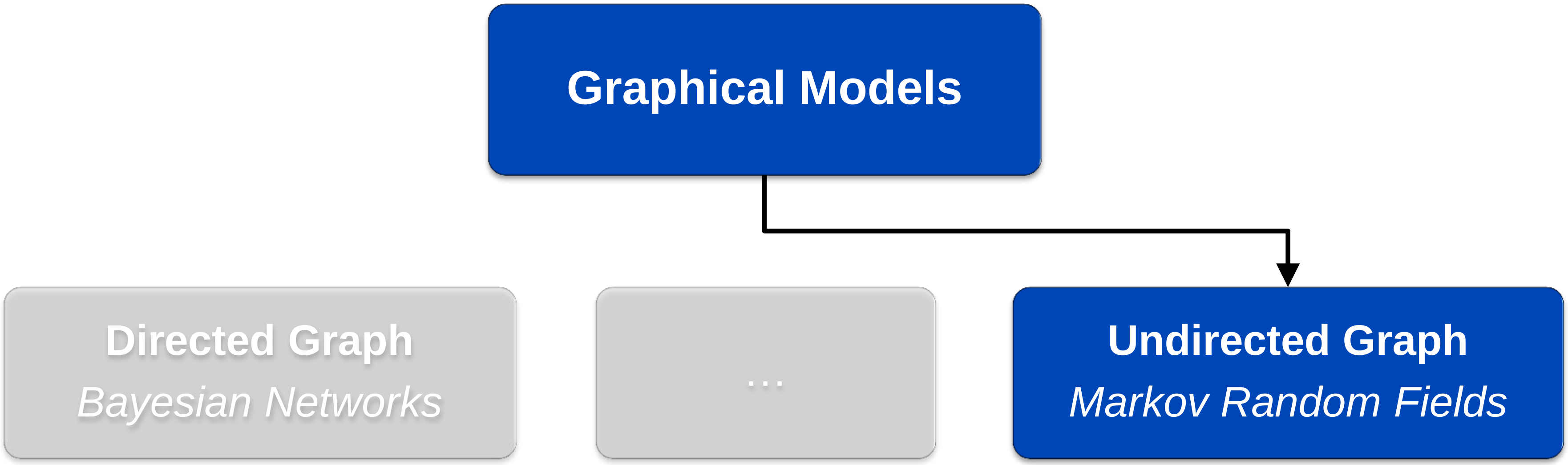
… to represents **conditional dependencies** in an **acyclic graph** …

Bayesian Networks



… but the regulation mechanism of a cell can be more complicated

Network Modeling
Restricted Boltzmann Machines (RBM)

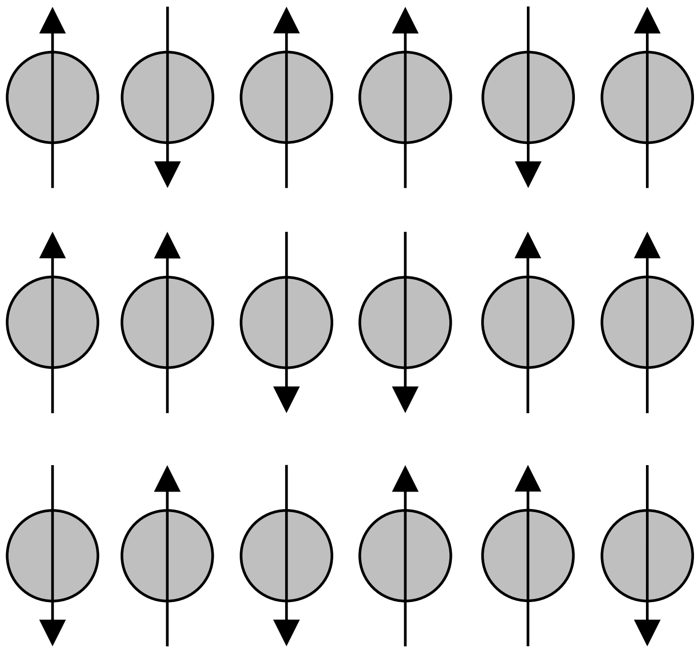


Another type of graphical models are **Markov Random Fields** (MRF)…

Markov Random Fields

Motivation (Ising Model)

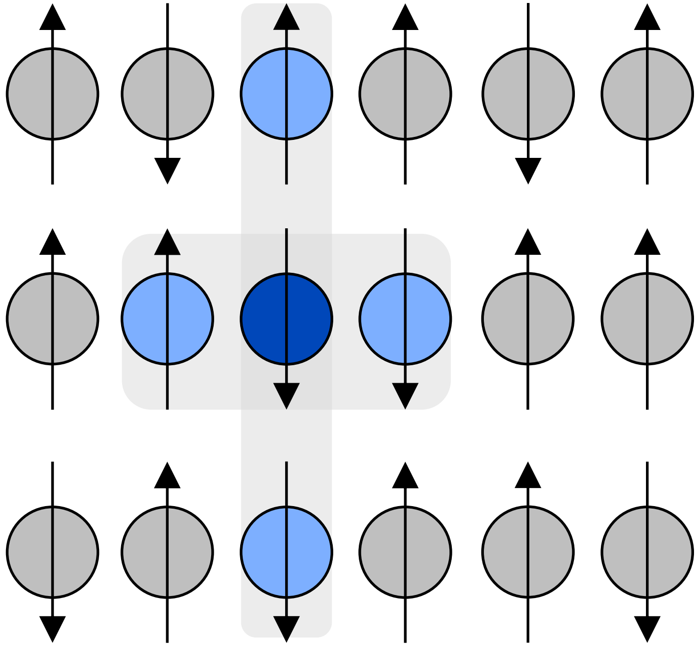
A set of magnetic dipoles (*spins*) is arranged in a graph (lattice) where neighbors are coupled with a given strengt



... which emerged with the **Ising Model** from statistical Physics …

Network Modeling
Restricted Boltzmann Machines (RBM)

Markov Random Fields



Motivation (Ising Model)

A set of magnetic dipoles (*spins*) is arranged in a graph (lattice) where neighbors are coupled with a given strengt

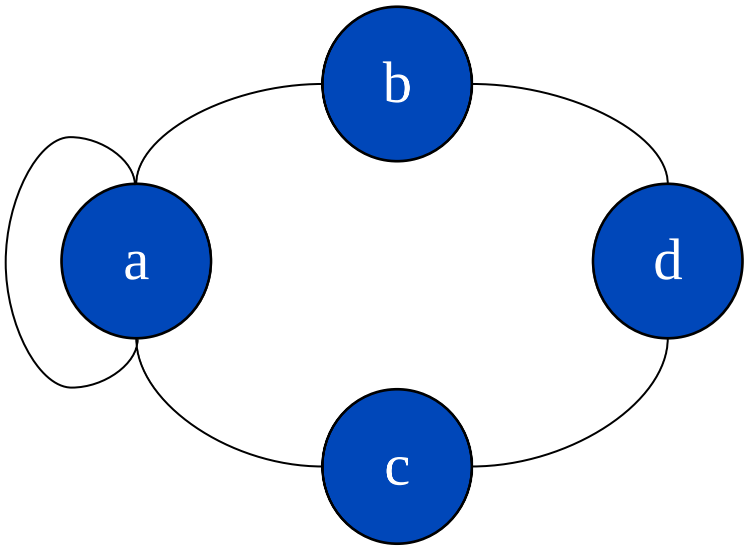
... which uses **local energies** to calculate new states …

Network Modeling
Restricted Boltzmann Machines (RBM)

Markov Random Fields

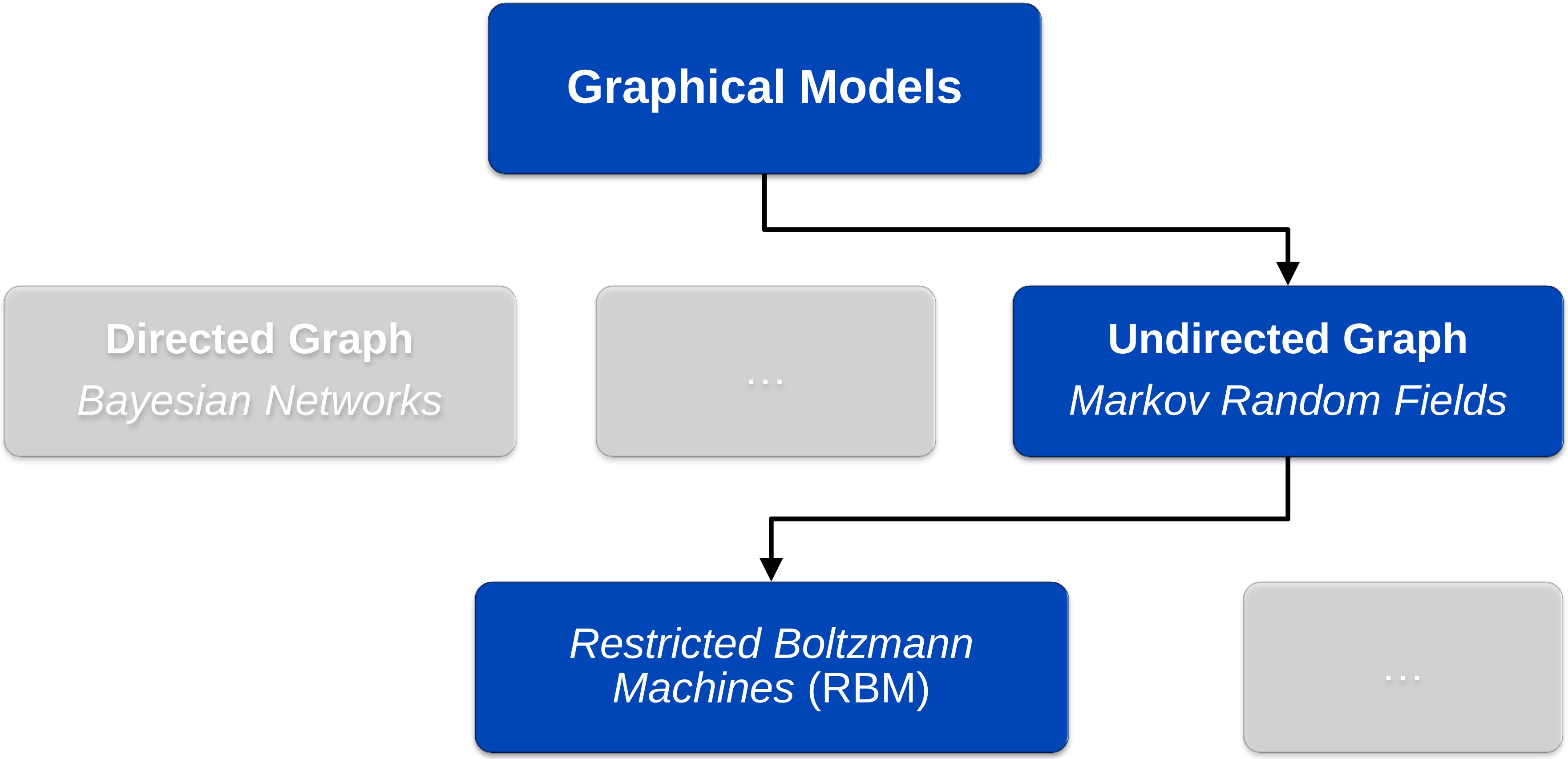
Drawback

By allowing cyclic dependencies the computational costs explode



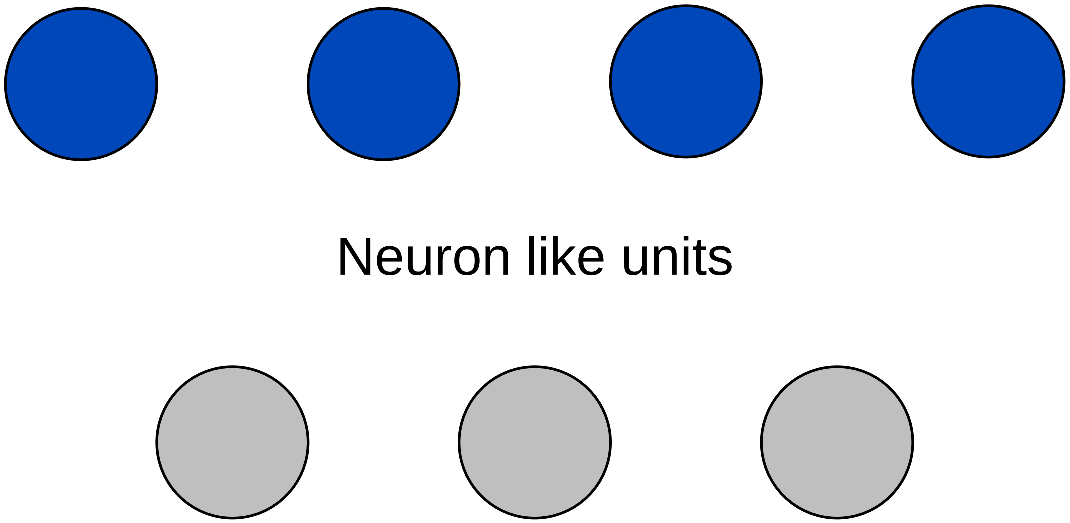
… the drawback are **high computational costs** …

Network Modeling
Restricted Boltzmann Machines (RBM)



… which can be avoided by using **Restricted Boltzmann Machines**

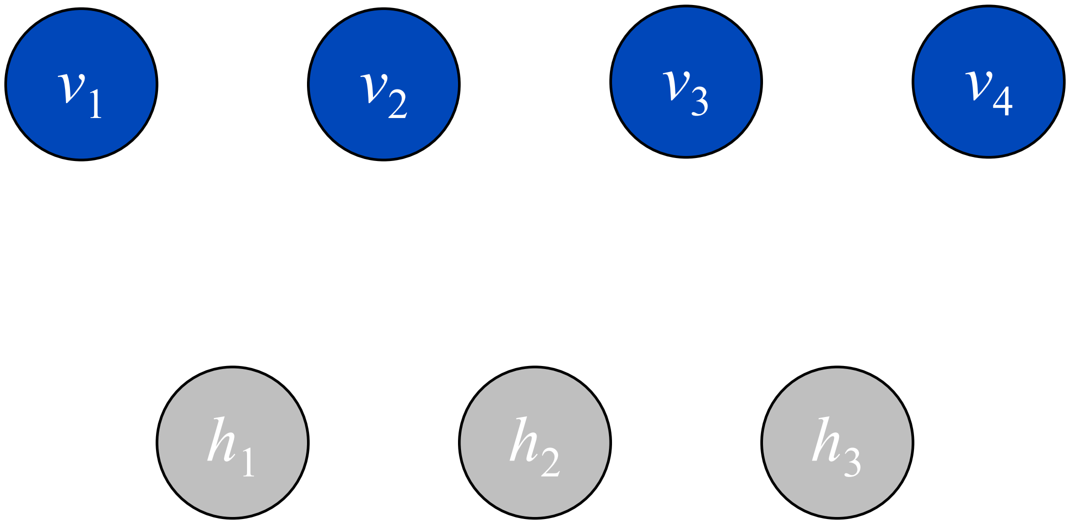
Restricted Boltzmann Machines



RBMs are Artificial Neuronal Networks …

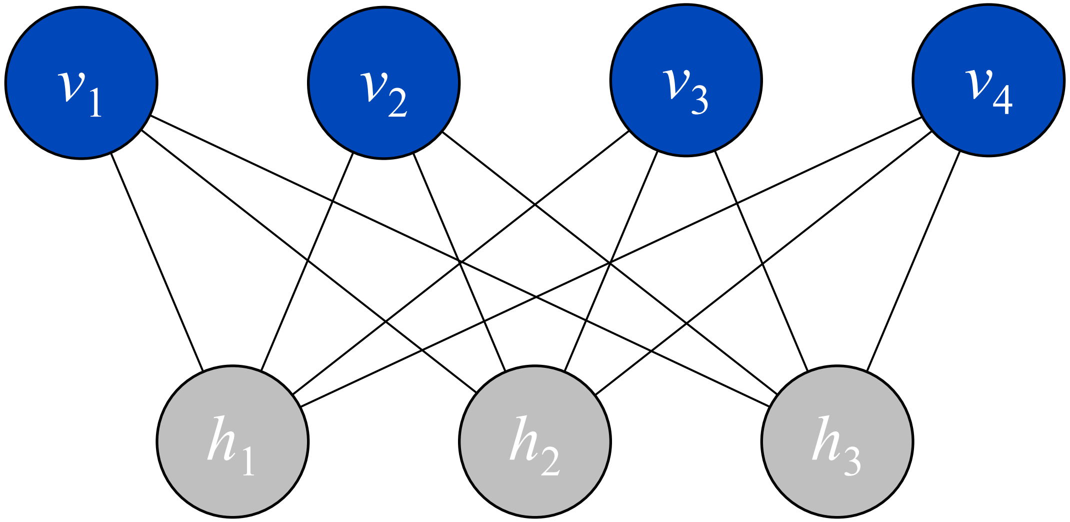
Network Modeling
Restricted Boltzmann Machines (RBM)

Restricted Boltzmann Machines



… with two layers: **visible units** (*v*) and **hidden units** (*h*)

Restricted Boltzmann Machines



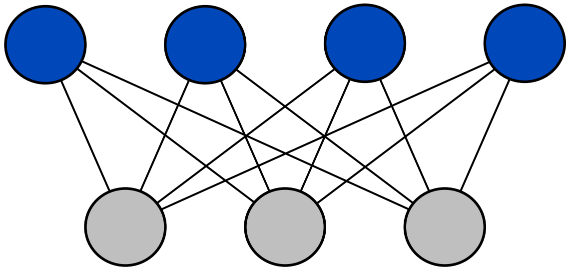
Visible units are **strictly** connected with hidden units

Restricted Boltzmann Machines

V := set of visible units

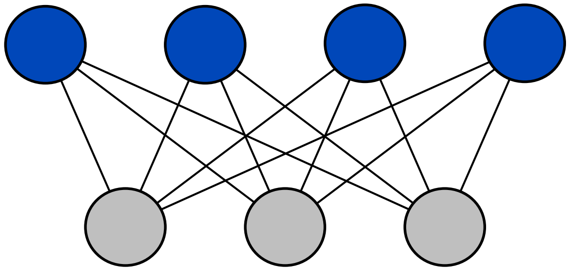
x_v := value of unit v, $\forall v \in V$

$x_v \in R$, $\forall v \in V$



In our model the visible units have **continuous values** …

Restricted Boltzmann Machines



V := set of visible units
x_v := value of unit v, $\forall v \in V$
$x_v \in R$, $\forall v \in V$

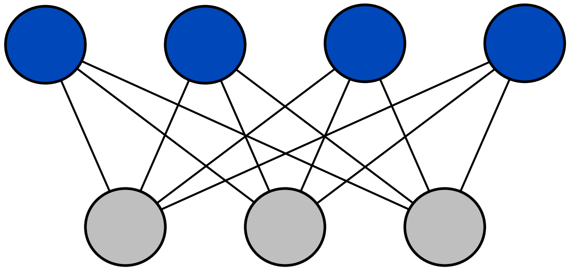
H := set of hidden units
x_h := value of unit h, $\forall h \in H$
$x_h \in \{0, 1\}$, $\forall h \in H$

… and the hidden units **binary values**

Network Modeling
Restricted Boltzmann Machines (RBM)

Restricted Boltzmann Machines

$$x_v \sim N\left(b_v + \sum_h w_{vh}\, x_h, \sigma_v\right), \forall v \in V$$



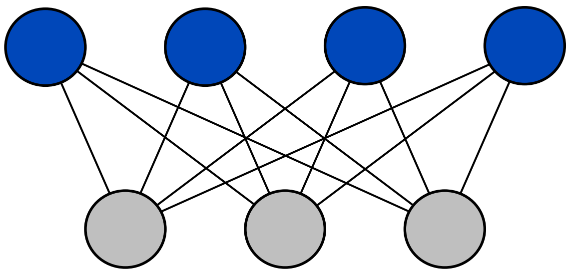
σ_v ≔ std. dev. of unit v
b_v ≔ bias of unit v
w_{vh} ≔ weight of edge (v, h)

Visible units are modeled with **gaussians** to encode **data** …

Restricted Boltzmann Machines

$$x_v \sim N(b_v + \sum_h w_{vh}\, x_h, \sigma_v), \forall v \in V$$

σ_v := std. dev. of unit v
b_v := bias of unit v
w_{vh} := weight of edge (v, h)



$$x_h \sim \text{sigmoid}\left(b_h + \sum_v w_{vh} \frac{x_v}{\sigma_v}\right), \forall h \in H$$

b_h := bias of unit h
w_{vh} := weight of edge (v, h)

… and hidden units with **simoids** to encode **dependencies**

Learning in Restricted Boltzmann Machines

Task: Find dependencies in data
↔ Find configuration of parameters with maximum likelihood (to data)

The challenge is to find the configuration of the parameters …

Learning in Restricted Boltzmann Machines

Task: Find dependencies in data
↔ Find configuration of parameters with maximum likelihood (to data)

In RBMs configurations of parameters have probabilities,
that can be defined by local energies

Local Energy

1

$$E_v := -\sum_h w_{vh} \frac{x_v}{\sigma_v} x_h + \frac{(x_v - b_v)^2}{2{\sigma_v}^2}$$

2

$$E_h := -\sum_v w_{vh} \frac{x_v}{\sigma_v} x_h + x_h b_h$$

Like in the Ising model the units states correspond to **local energies** …

Learning in Restricted Boltzmann Machines

Task: Find dependencies in data
↔ Find configuration of parameters with maximum likelihood (to data)
↔ Minimize global energy (to data)

Global Energy

$$E := \sum_v E_v + \sum_h E_h = -\sum_v \sum_h w_{vh} \frac{x_v}{\sigma_v} x_h + \sum_v \frac{(x_v - b_v)^2}{2{\sigma_v}^2} + \sum_h w_{vh} \frac{x_v}{\sigma_v} x_h$$

… which sum to a **global energy**, which is our objective function

Network Modeling
Restricted Boltzmann Machines (RBM)

Learning in Restricted Boltzmann Machines

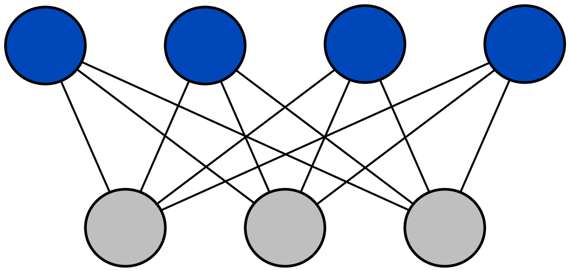
Task: Find dependencies in data
↔ Find configuration of parameters with maximum likelihood (to data)
↔ Minimize global energy (to data)
↔ Perform stochastic gradient descent on σ_v, b_v, b_h, w_{vh} (to data)

The optimization can be done using stochastic **gradient descent** …

Learning in Restricted Boltzmann Machines

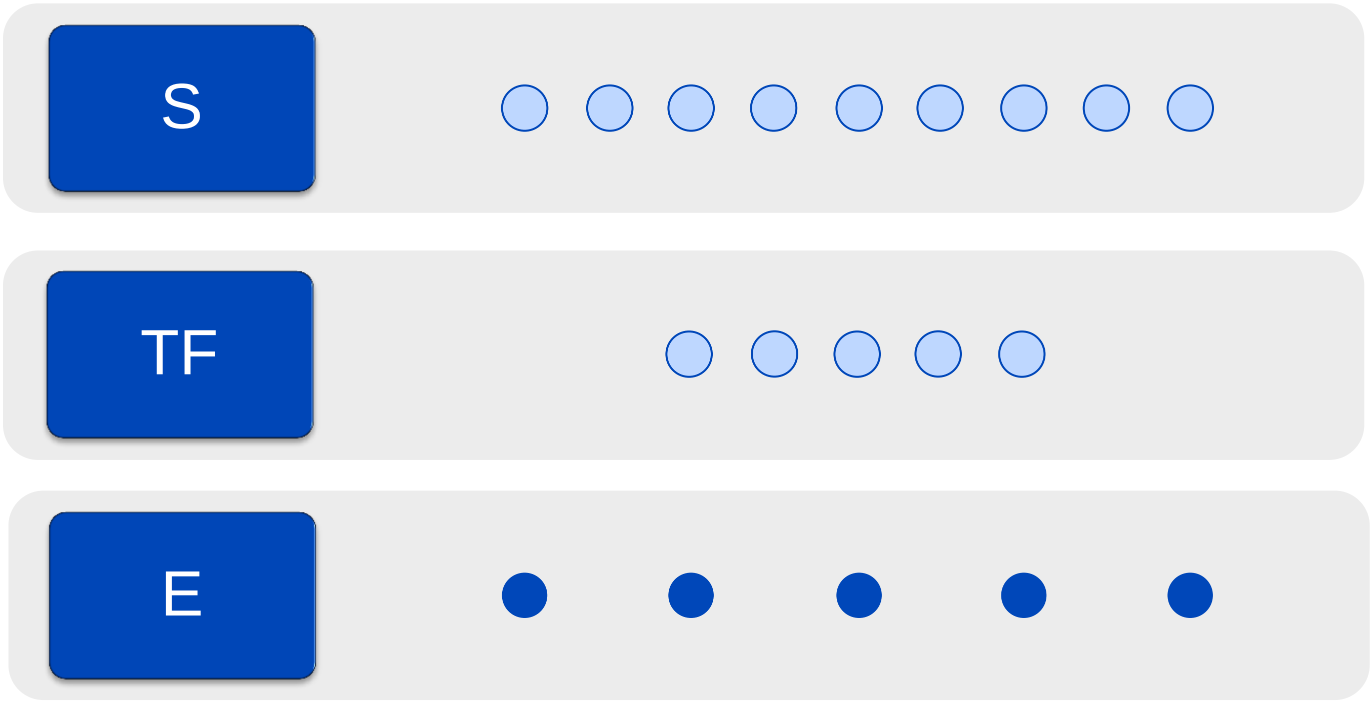
Task: Find dependencies in data
↔ Find configuration of parameters with maximum likelihood (to data)
↔ Minimize global energy (to data)
↔ Perform stochastic gradient descent on σ_v, b_v, b_h, w_{vh} (to data)

Gradient Descent on RBMs
The bipartite graph structure allows *constrastive divergency* learning, using *Gibbs-sampling*



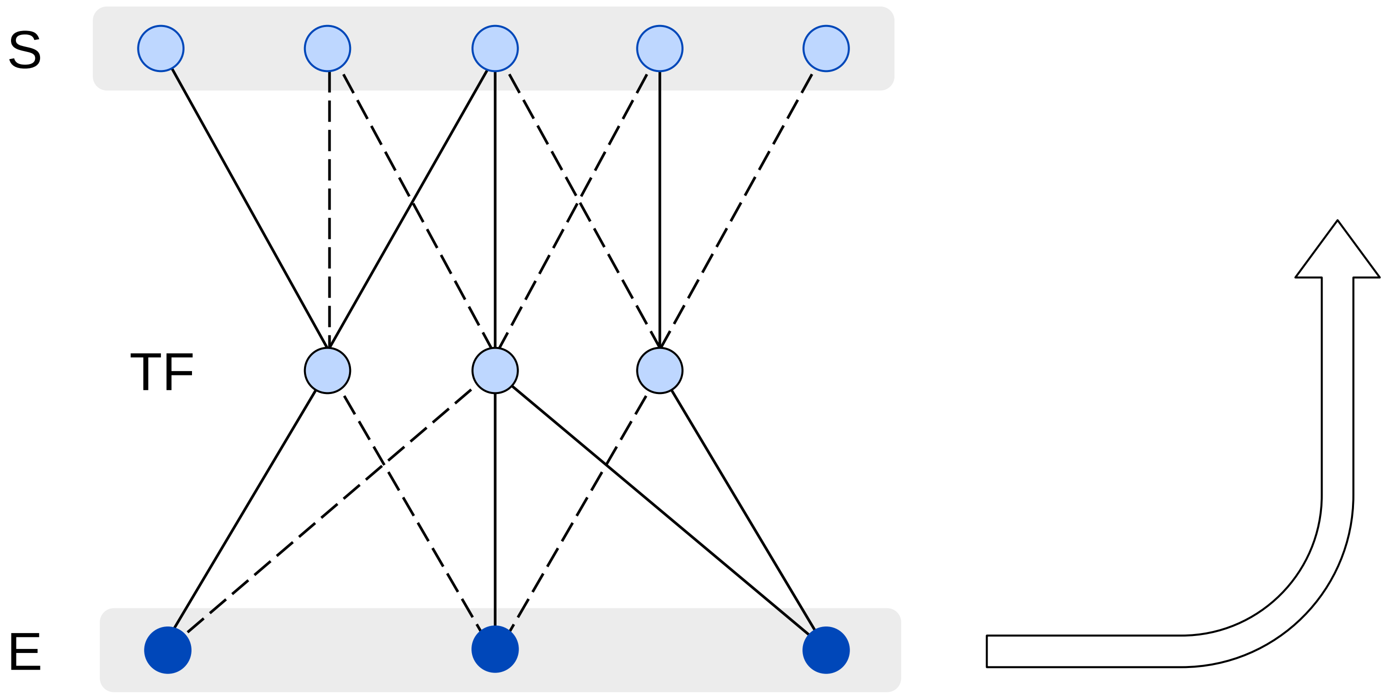
… which has an efficient learning algorithmus

Network Modeling
Restricted Boltzmann Machines (RBM)



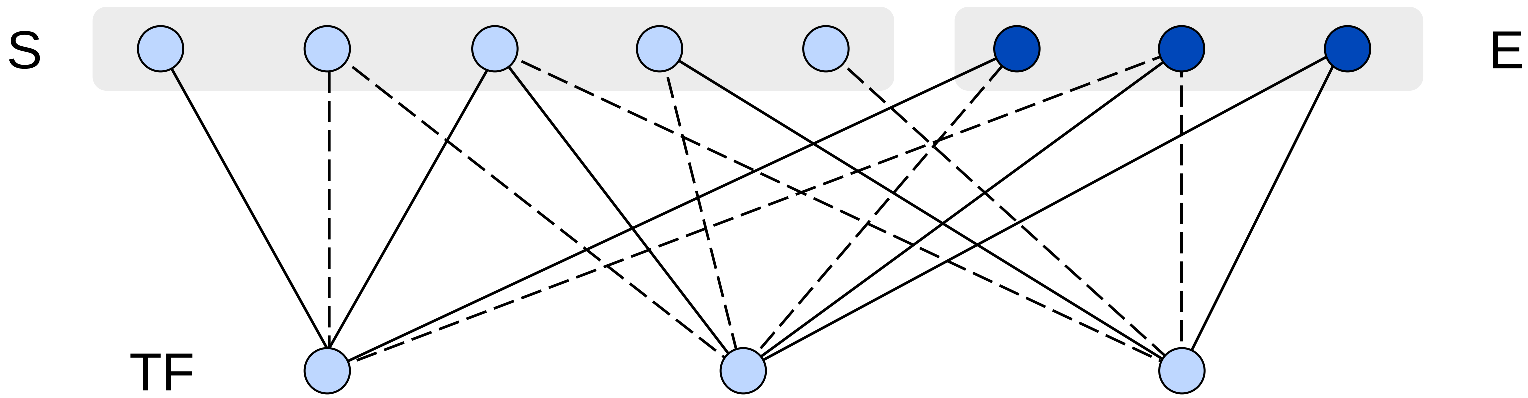
How to model our initial structure as an RBM?

Network Modeling
Restricted Boltzmann Machines (RBM)



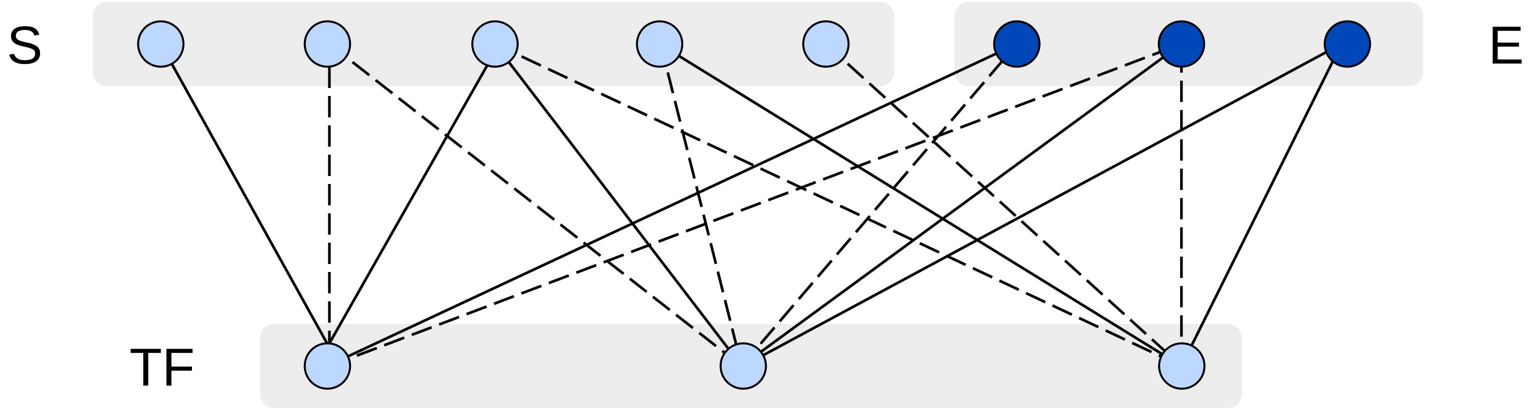
We define S and E as **visible Layer** …

Network Modeling
Restricted Boltzmann Machines (RBM)



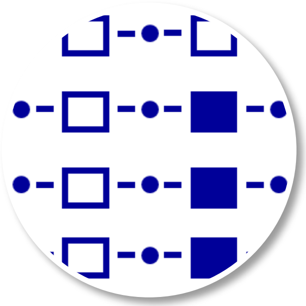
We define S and E as **visible Layer** …

Network Modeling
Restricted Boltzmann Machines (RBM)

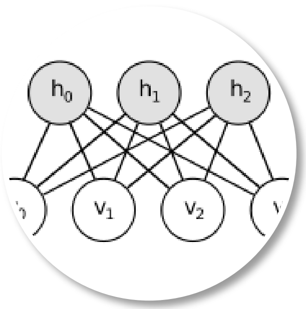


… and TF as **hidden Layer**

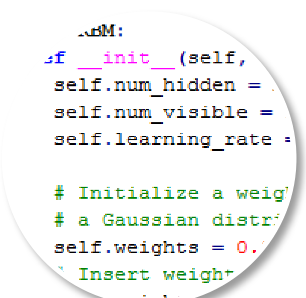
Agenda



Biological Problem
Analysing the regulation of metabolism



Network Modeling
Restricted Boltzmann Machines (RBM)



Implementation & Results
python::metapath

Results

Validation of the results

- Information about the true regulation
- Information about the descriptive power of the data

Results

Validation of the results

- Information about the true regulation
- Information about the descriptive power of the data

Without this infomation validation can only be done, using simulated data!

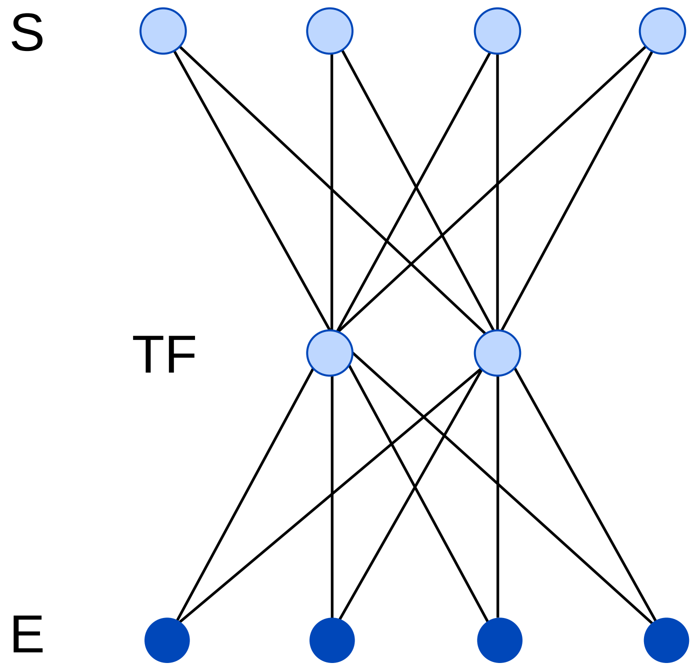
Results

Simulation 1

First of all we need to understand how the modell handles **dependencies** and **noise**

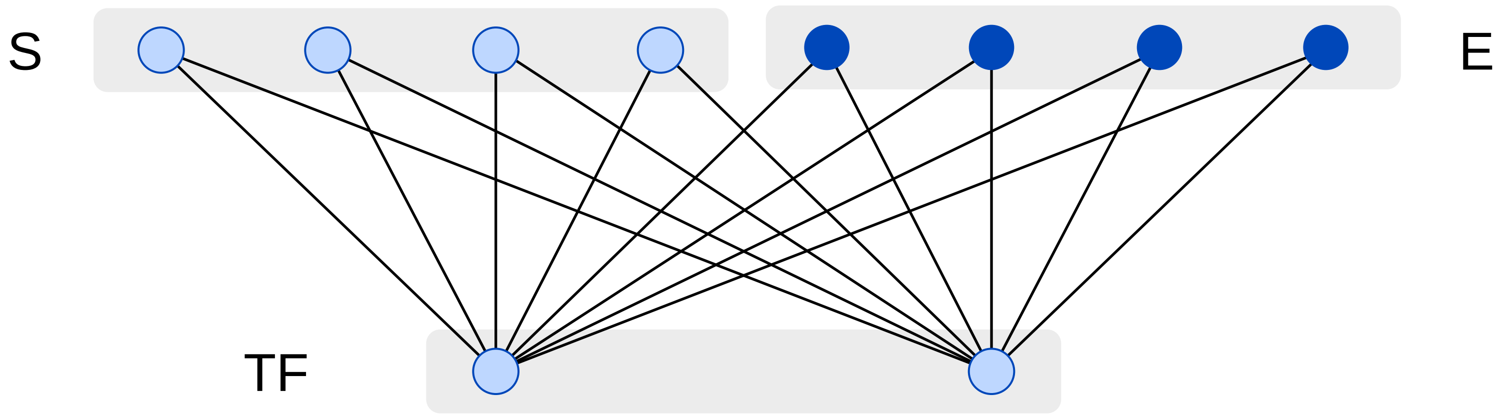
To demonstrate this we create very simple data with a simple structure

Simulation 1



What can we expect from this model?

Simulation 1

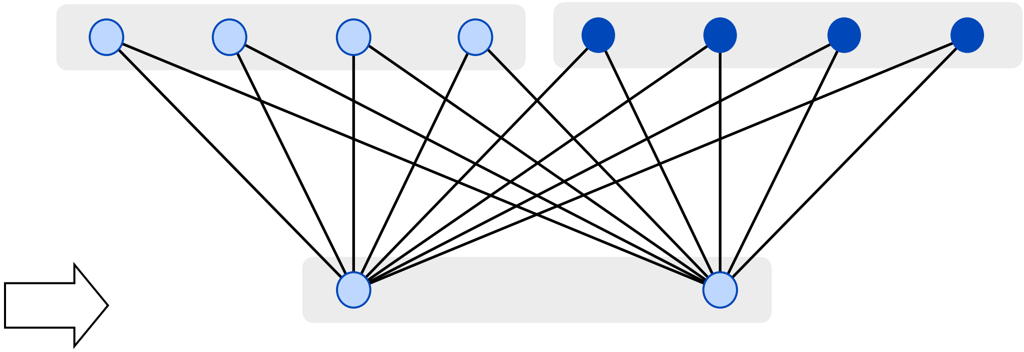


… as RBM we get 8 visible and 2 hidden units, fully connected

Simulation 1

Data

S	E
1,0,0,1	1,0,0,0
1,0,0,1	1,1,0,0
1,0,0,1	1,0,1,0
1,0,0,1	1,0,0,1
1,0,1,1	0,0,0,0
1,0,1,1	0,1,0,0
1,0,1,1	0,0,1,0
1,0,1,1	0,0,0,1

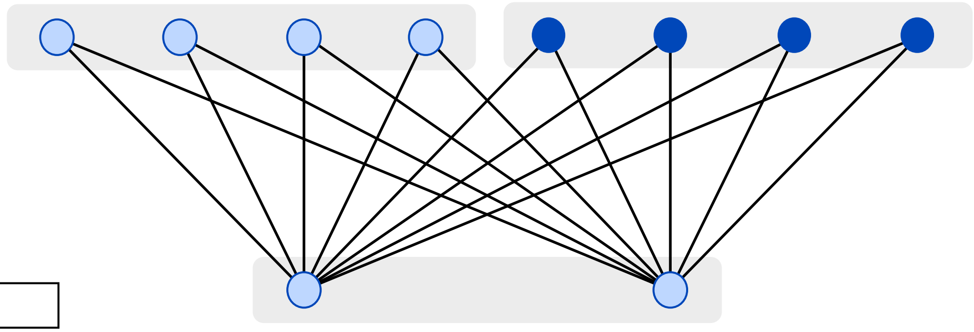


Let‘s feed the machine with **samples** …

Simulation 1

Weight matrix

	TF_1	TF_2
S_1	0,3	0,8
S_2	0,5	0,6
S_3	1,0	0,1
S_4	0,3	0,8
E_1	0,8	0,0
E_2	0,1	0,0
E_3	0,1	0,0
E_4	0,2	0,0

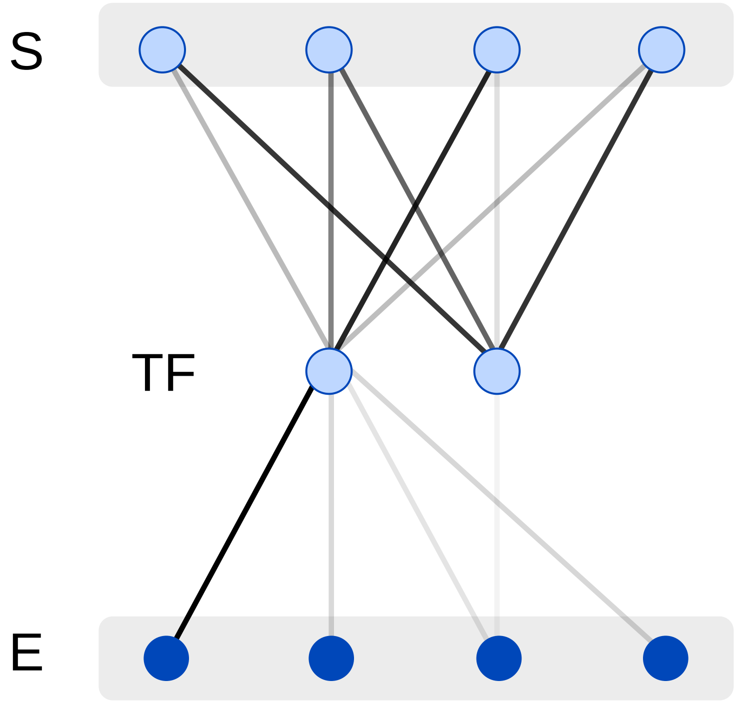


.. to get the calculated parameters (especially the **weight matrix**)

Simulation 1

Weight matrix

	TF_1	TF_2
S_1	0,3	0,8
S_2	0,5	0,6
S_3	1,0	0,1
S_4	0,3	0,8
E_1	0,8	0,0
E_2	0,1	0,0
E_3	0,1	0,0
E_4	0,2	0,0

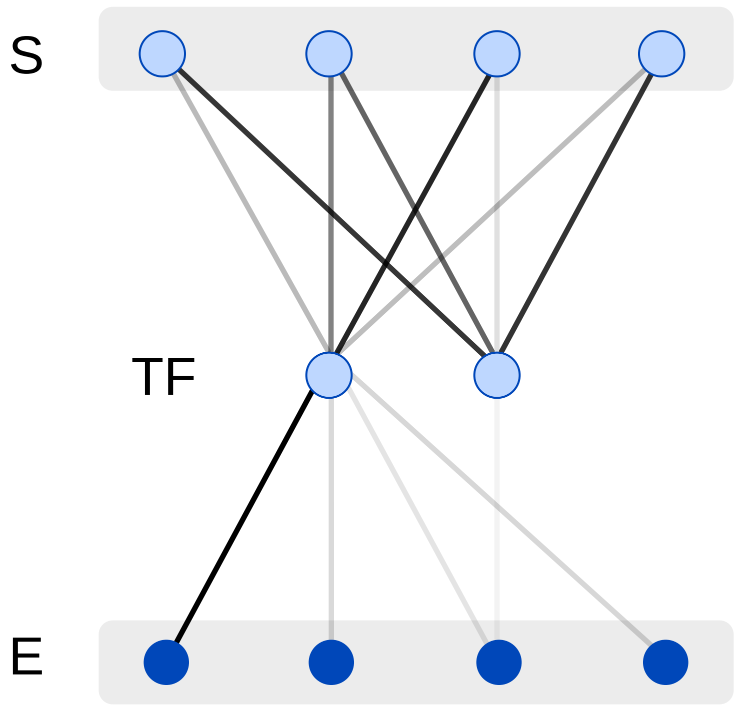


The weights are visualized by the **intensity** of the edges

Simulation 1

Learning samples

S	E
1,0,0,1	1,0,0,0
1,0,0,1	1,1,0,0
1,0,0,1	1,0,1,0
1,0,0,1	1,0,0,1
1,0,1,1	0,0,0,0
1,0,1,1	0,1,0,0
1,0,1,1	0,0,1,0
1,0,1,1	0,0,0,1

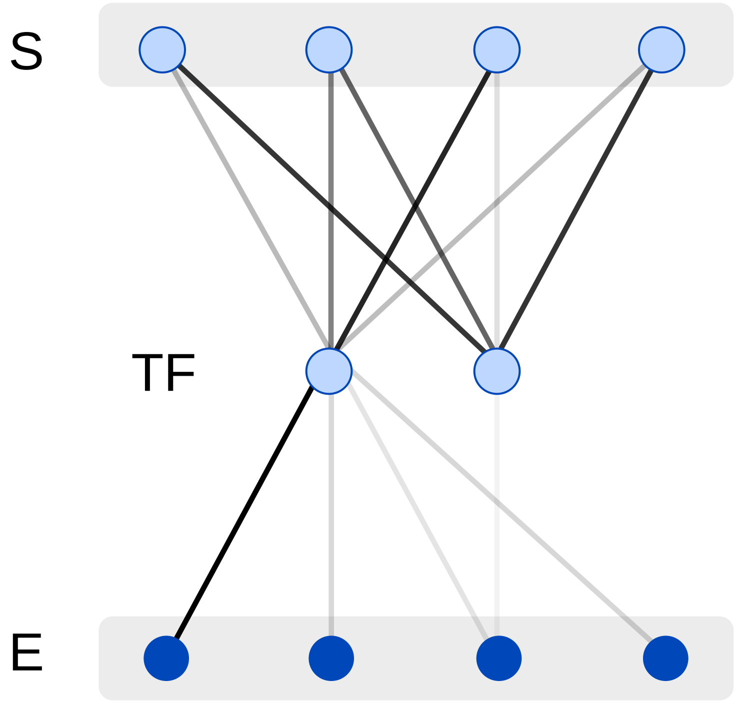


Now we can compare the results with the samples

Simulation 1

Learning samples

S	E
1,0,**0**,1	**1**,0,0,0
1,0,**0**,1	**1**,1,0,0
1,0,**0**,1	**1**,0,1,0
1,0,**0**,1	**1**,0,0,1
1,0,**1**,1	**0**,0,0,0
1,0,**1**,1	**0**,1,0,0
1,0,**1**,1	**0**,0,1,0
1,0,**1**,1	**0**,0,0,1

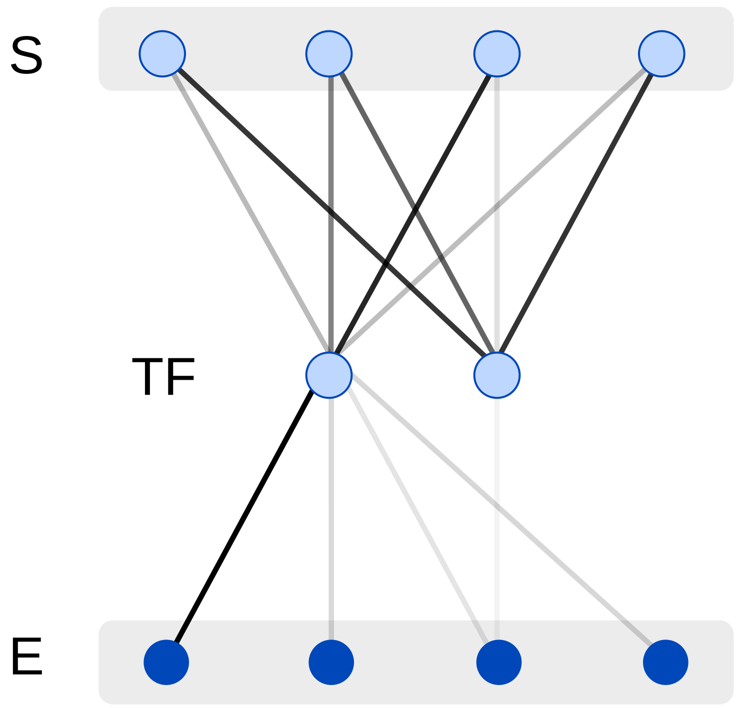


There's a strong dependency between S_3 an E_1

Simulation 1

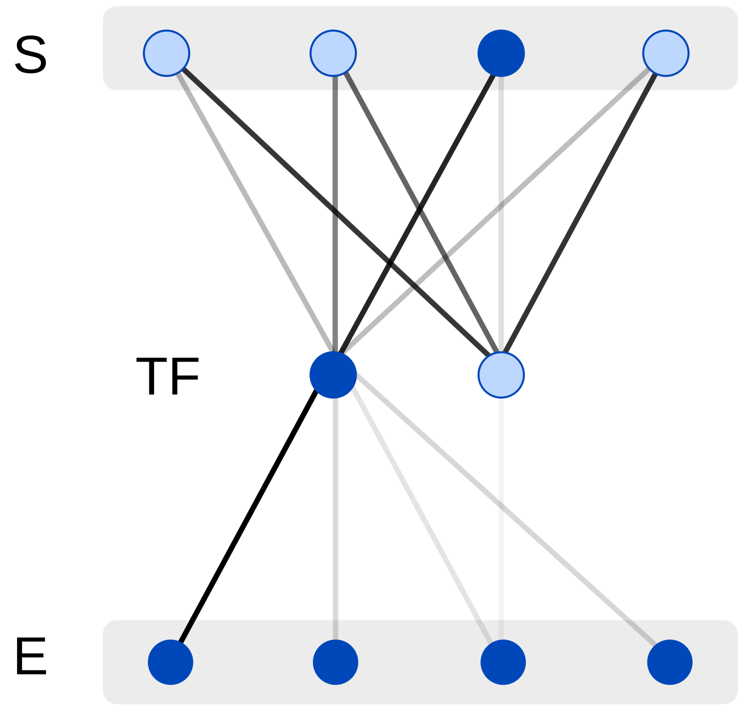
Learning samples

S	E
1,0,0,1	1,0,0,0
1,0,0,1	1,1,0,0
1,0,0,1	1,0,1,0
1,0,0,1	1,0,0,1
1,0,1,1	0,0,0,0
1,0,1,1	0,1,0,0
1,0,1,1	0,0,1,0
1,0,1,1	0,0,0,1



S_1, S_2 and S_4 do almost not affect the metabolism …

Simulation 1

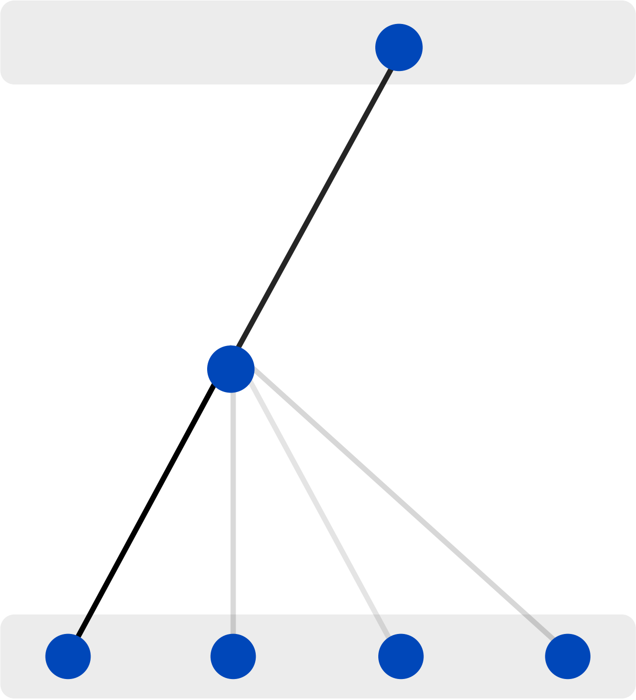


… so we can forget them and get S_1, TF_1 for our regulation model

Simulation 1

Weight matrix

	TF_1	TF_2
S_1	0,3	0,8
S_2	0,5	0,6
S_3	1,0	0,1
S_4	0,3	0,8
E_1	0,8	0,0
E_2	0,1	0,0
E_3	0,1	0,0
E_4	0,2	0,0

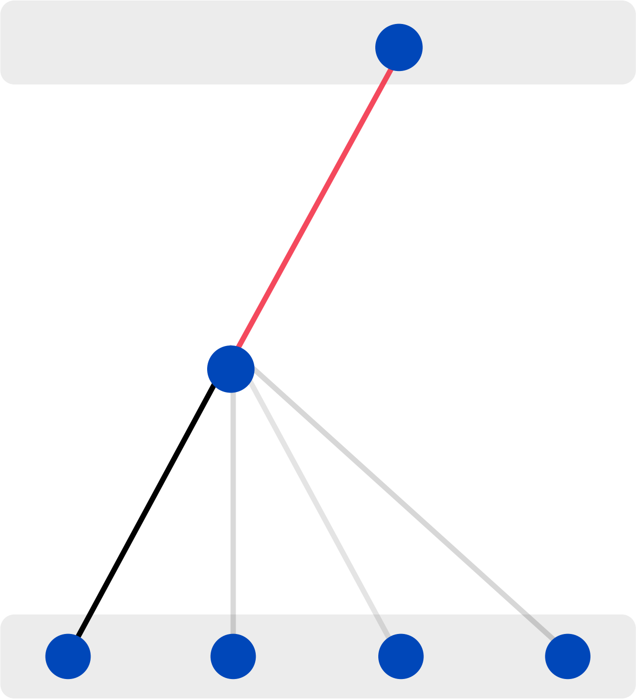


We can also take a look at the causal mechanism …

Simulation 1

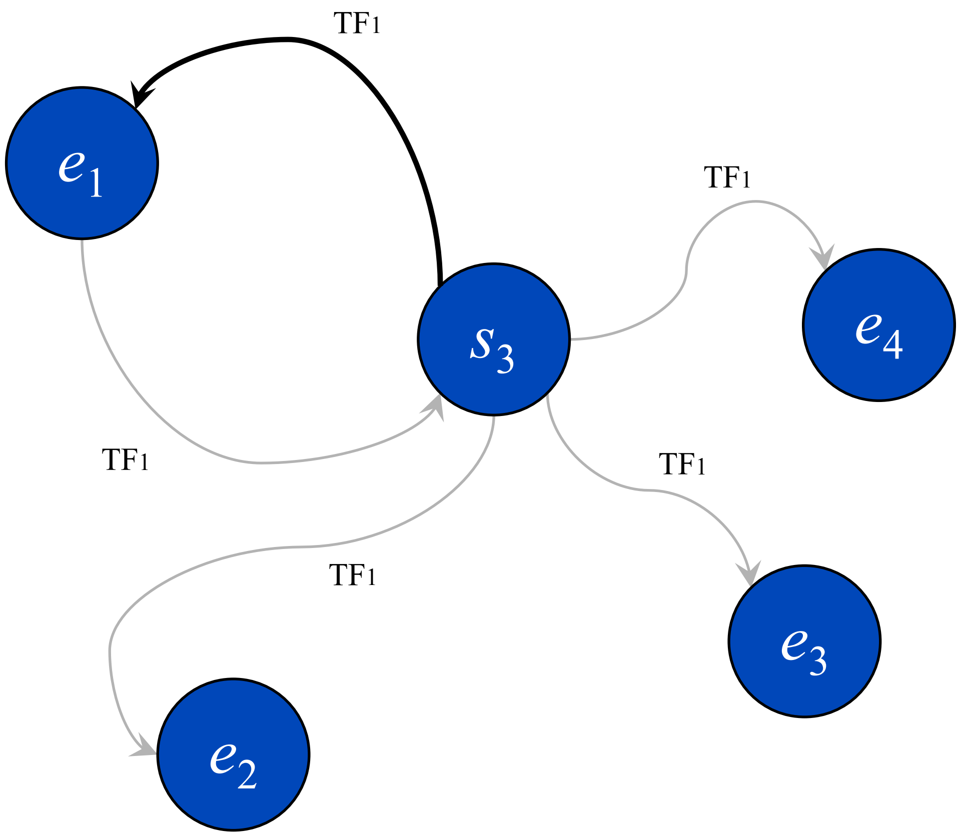
Weight matrix

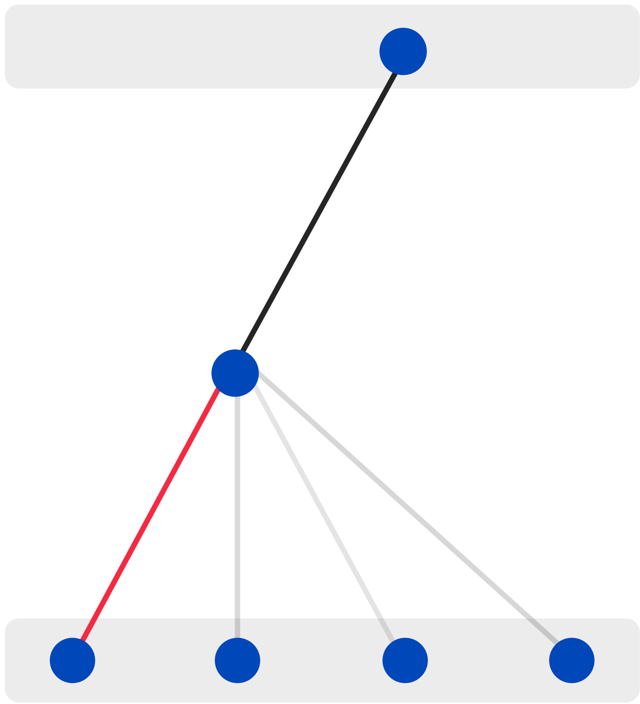
	TF_1	TF_2
S_1	0,3	0,8
S_2	0,5	0,6
S_3	1,0	0,1
S_4	0,3	0,8
E_1	0,8	0,0
E_2	0,1	0,0
E_3	0,1	0,0
E_4	0,2	0,0



The edge (S_3, TF_1) dominates TF_1 …

Simulation 1





Also E_1 seems to have an effect on S_3 (fewer than S_3 on E_1)

Results

Comparing to Bayesian Networks

For this purpose we simulate data in three steps

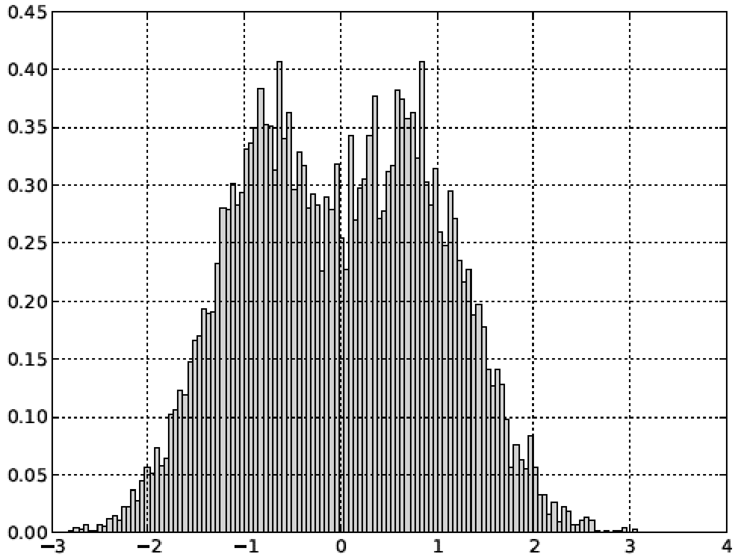
Of course we want to compare the method with Bayesian Networks

Results

Comparing to Bayesian Networks

Step 1

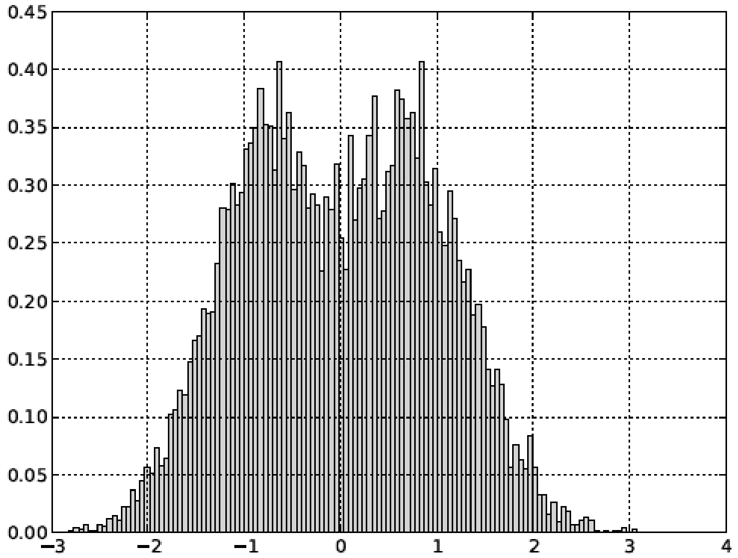
Choose number of Genes (E+S) and create random bimodal distributed data



Of course we want to compare the method with Bayesian Networks

Results

Comparing to Bayesian Networks



Step 1

Choose number of Genes (E+S) and create random bimodal distributed data

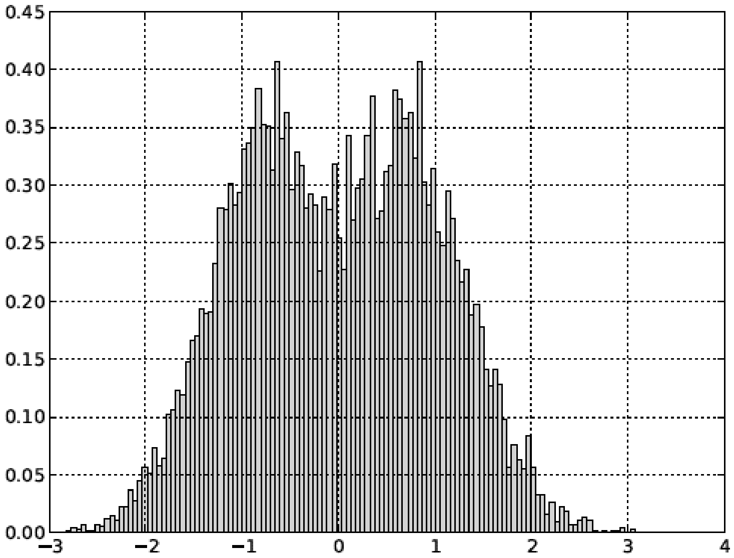
Step 2

Manipulate data in a fixed order

Of course we want to compare the method with Bayesian Networks

Results

Comparing to Bayesian Networks



Step 1

Choose number of Genes (E+S) and create random bimodal distributed data

Step 2

Manipulate data in a fixed order

Step 3

Add noise to manipulated data and normalize data

Of course we want to compare the method with Bayesian Networks

Results

Comparing to Bayesian Networks

Idea

- ‚melt down‘ the bimodal distribution from very sharp to very noisy
- Try to find the original causal structure with BN and RBM
- Measure Accuracy by counting the right and wrong dependencies

Of course we want to compare the method with Bayesian Networks

Results

Simulation 2

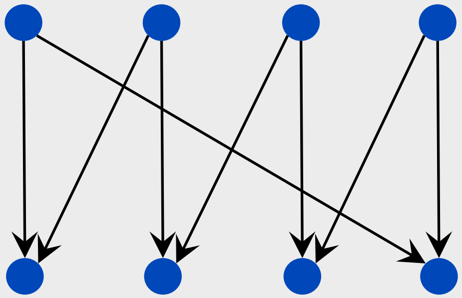
Step 1: Number of visible nodes 8 (4E, 4S)

Create intergradient datasets from sharp to noisy bimodal distribution

$\sigma_1 = 0.0, \sigma_1 = 0.3, \sigma_3 = 0.9, \sigma_4 = 1.2, \sigma_4 = 1.5$

Step 2 + 3: Data Manipulation + add noise

$$e_1 = 0.5s_1 + 0.5s_2 + N(\mu = 0, \sigma)$$
$$e_2 = 0.5s_2 + 0.5s_3 + N(\mu = 0, \sigma)$$
$$e_3 = 0.5s_3 + 0.5s_4 + N(\mu = 0, \sigma)$$
$$e_4 = 0.5s_4 + 0.5s_1 + N(\mu = 0, \sigma)$$

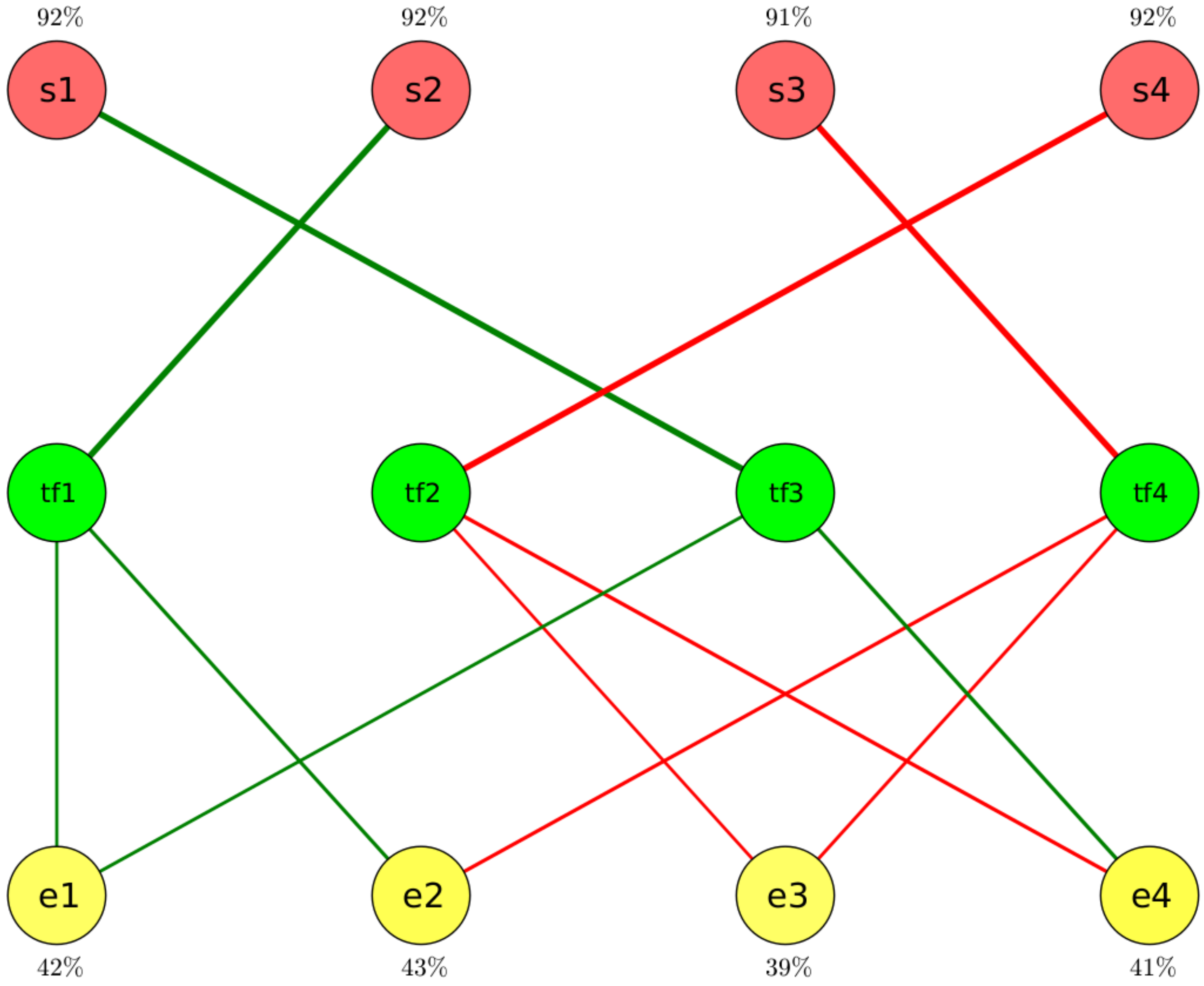


Of course we want to compare the method with Bayesian Networks

Results

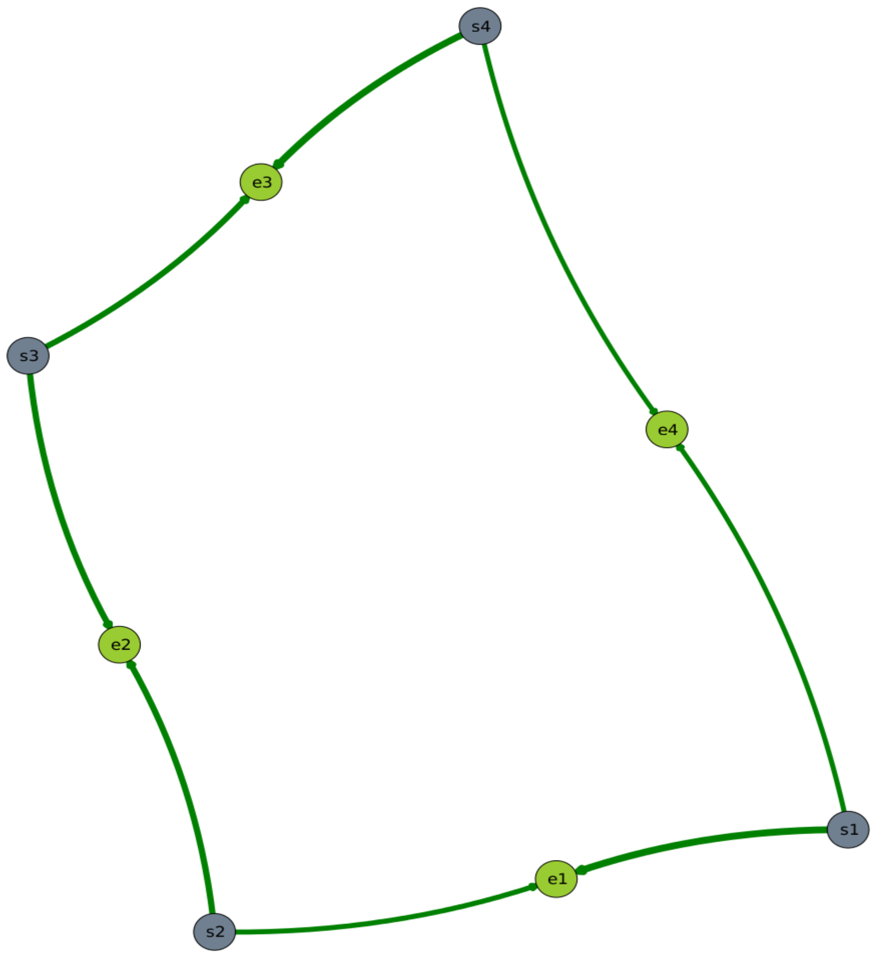
Simulation 2

RBM
Model
($\sigma = 0.0$)



Results

Simulation 2



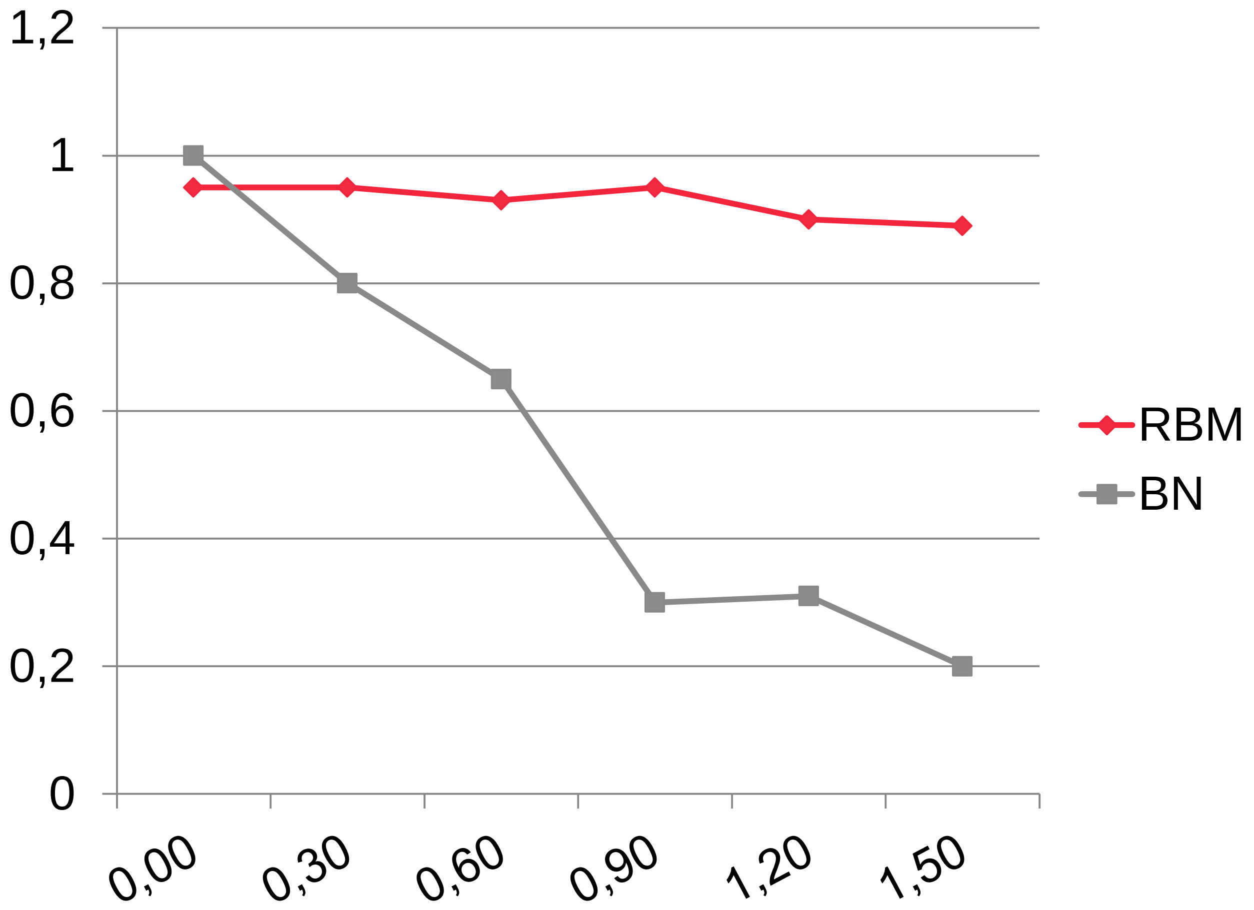
Causal Mechanism ($\sigma = 0.0$)

Results

Simulation 2

Comparison

BN / RBM



Conclusion

- RBMs are more **stable against noise** compared to BNs.

  It has to be assumed that RBMs have high predictive power regarding the regulation mechanisms of cells

- The drawback are **high computational costs**

  Since RBMs are getting more popular (Face recognition / Voice recognition, Image transformation). Many new improvements in facing the computational costs have been made.

Acknowledgement

eilsLABS

PD Dr. Rainer König

Prof. Dr Roland Eils

Network Modeling Group