Analysis and inference of gene networks from genomic data

Jean-Philippe Vert
Ecole des Mines de Paris
Computational Biology group
`Jean-Philippe.Vert@mines.org`

Institute for Statistical Mathematics, Tokyo, Japan, November 5th, 2004.

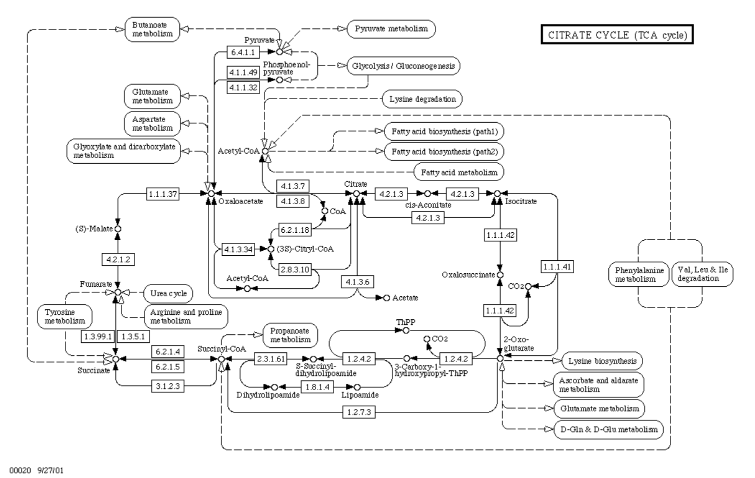
Thanks

- Yoshihiro Yamanishi (Kyoto University)
- Computational biology at the Ecole des Mines

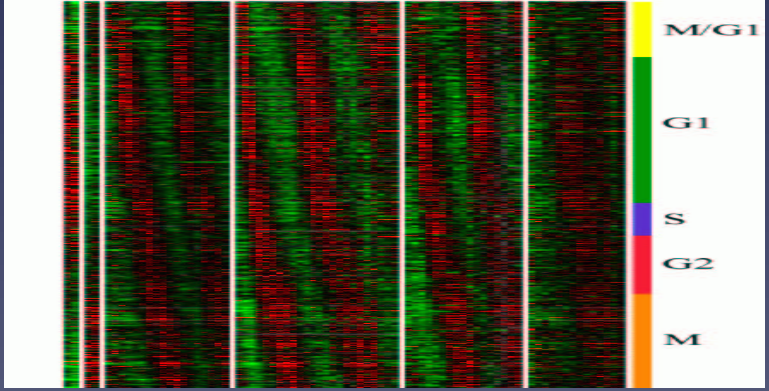
Motivations

- Many heterogeneous data about genes : sequences, expression, evolution, structures, etc...
- More and more data between genes: interactome, pathways, regulation etc...
- Goal: propose a formalism and algorithms to compare these data, and to infer gene networks from high-throughput genomic data.

Example 1:
Comparing gene expression and pathway databases



VS

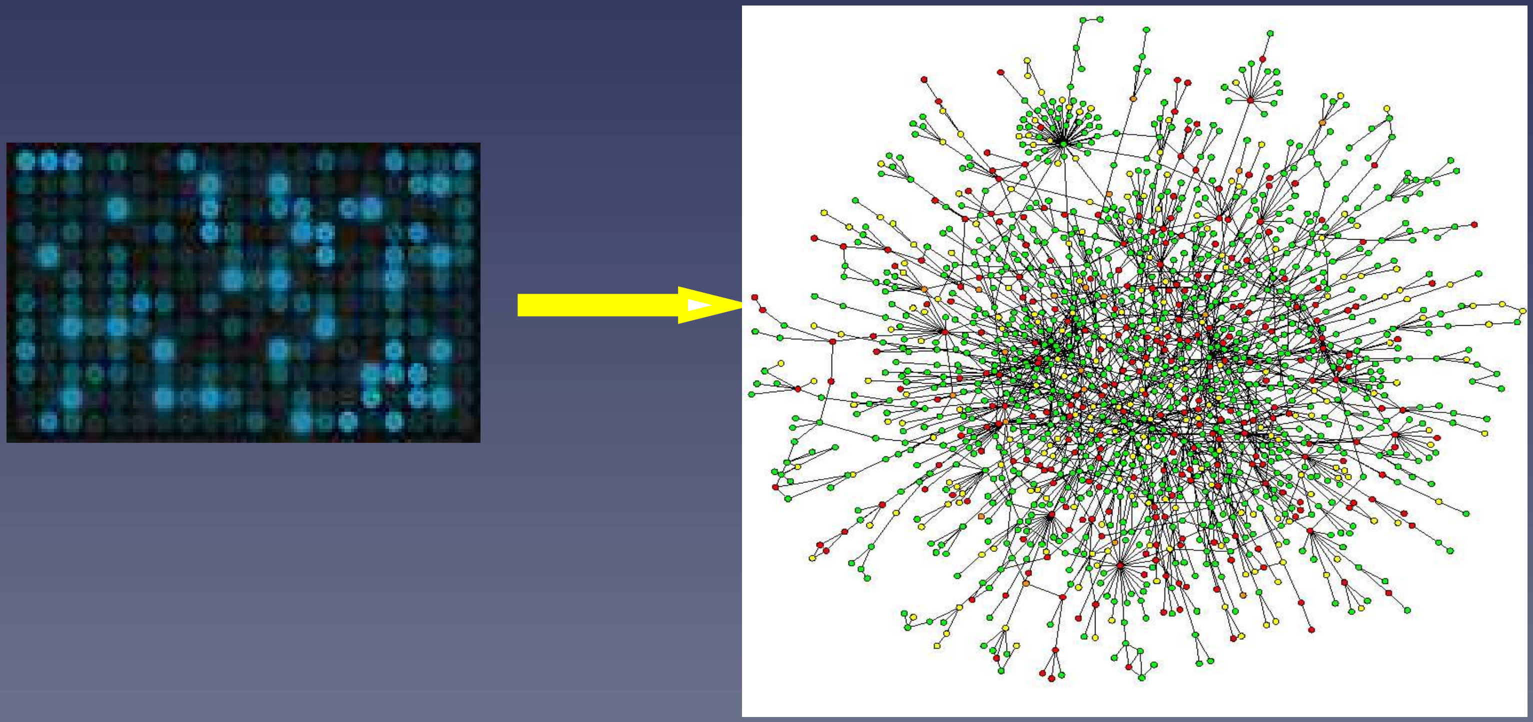


Detect active pathways? Denoise expression data?

Denoise pathway database? Find new pathways?

Are there "correlations"?

Example 2:
Gene network inference



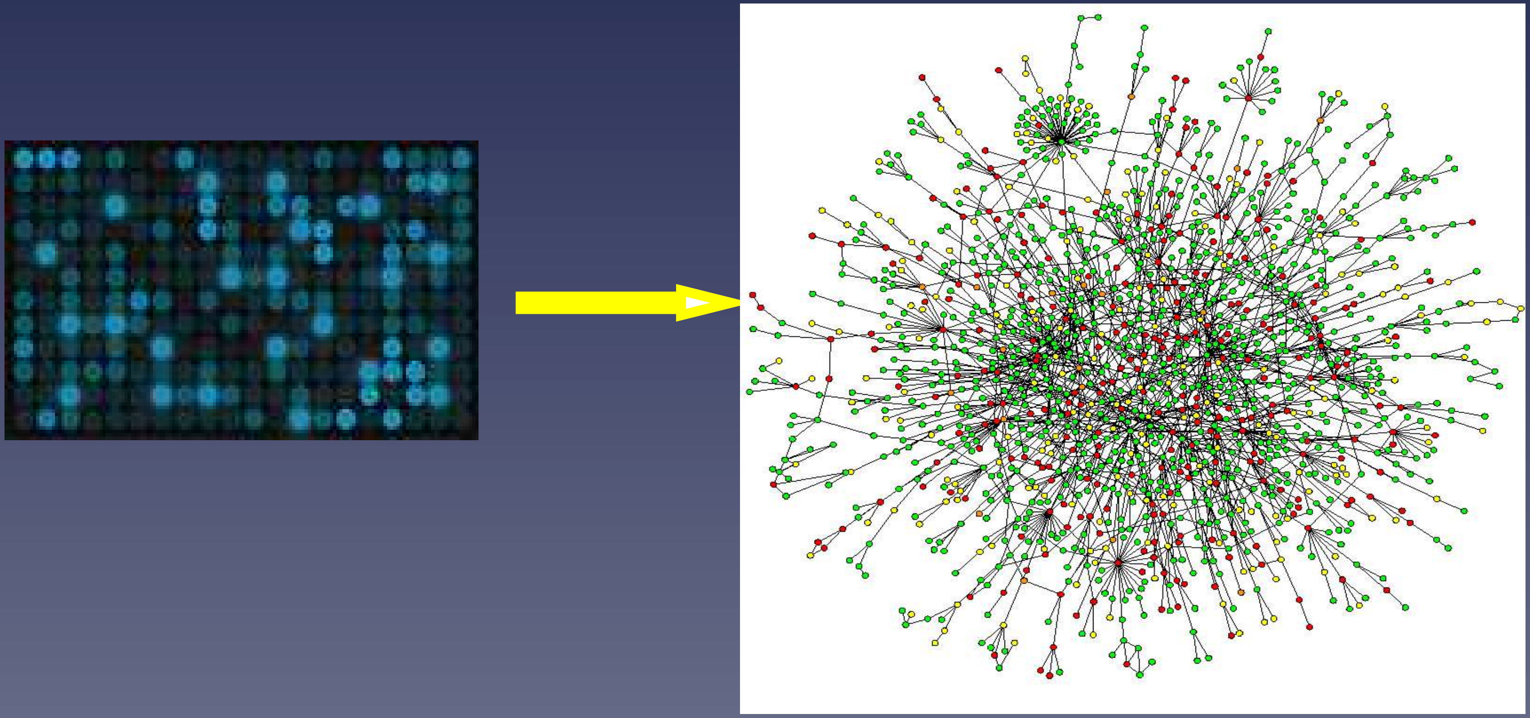
Outline

- A direct approach to network inference
- Supervised network inference
- Extraction of pathway activity
- Learning from several heterogeneous data

Part 1

A direct approach to network inference

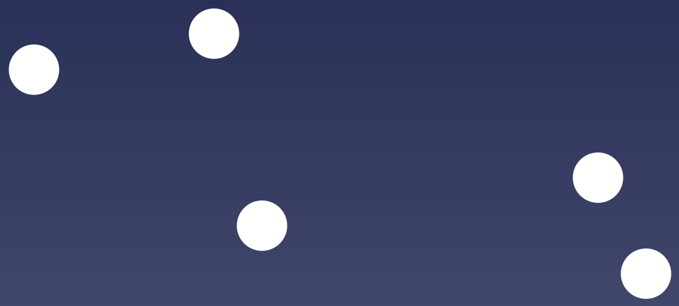
The problem



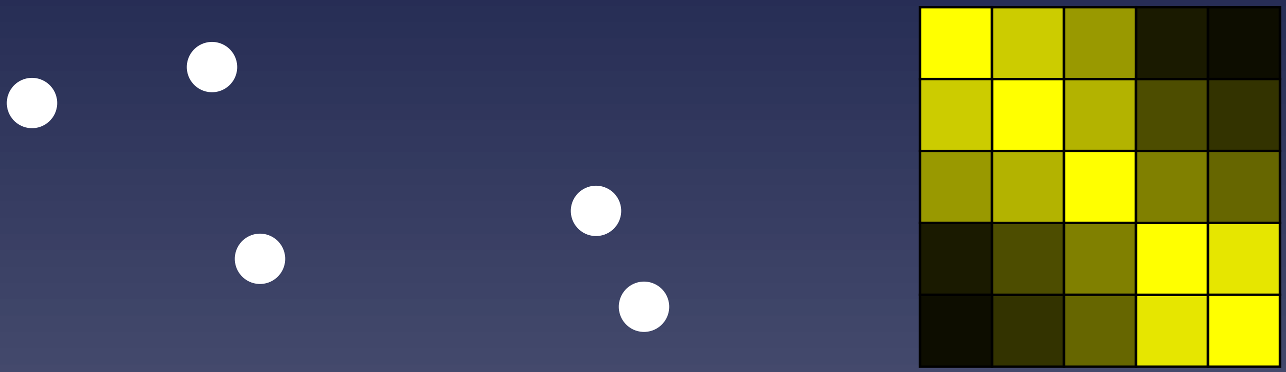
Related approaches

- Bayesian nets for regulatory networks (Friedman et al. 2000)
- Boolean networks (Akutsu, 2000)
- Joint graph method (Marcotte et al, 1999)

Network inference : the direct approach



Network inference : the direct approach

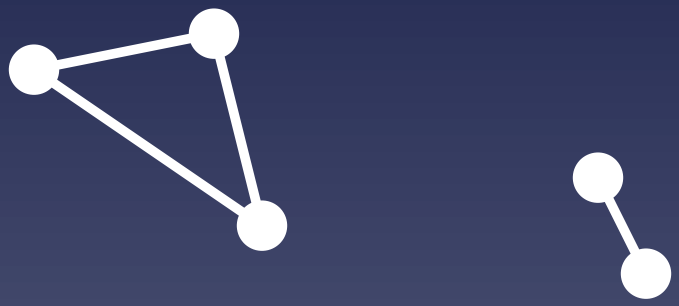


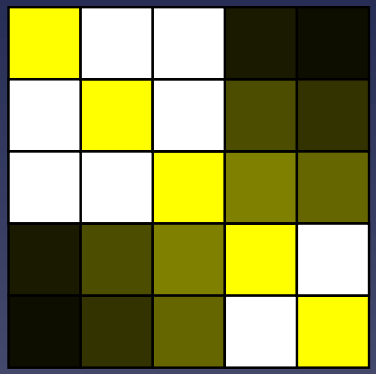
Network inference : the direct approach

Network inference : the direct approach

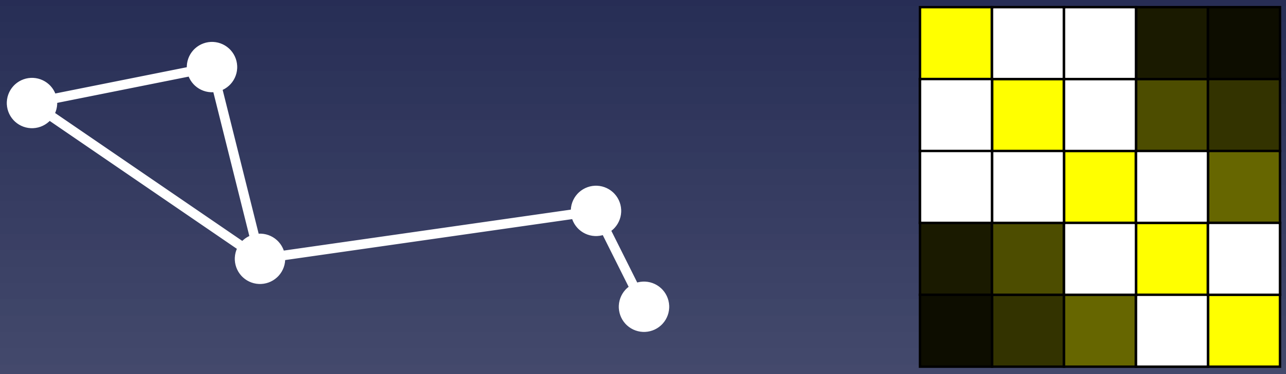
Network inference : the direct approach

Network inference : the direct approach

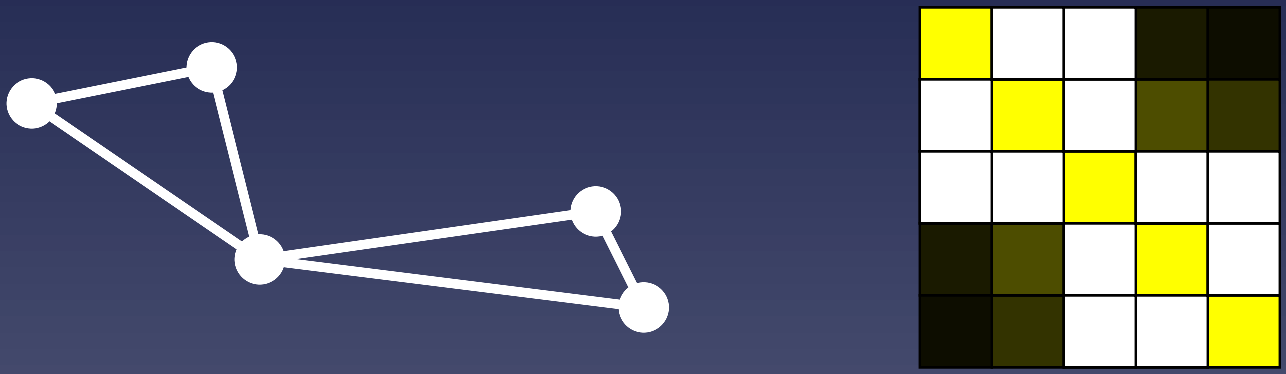




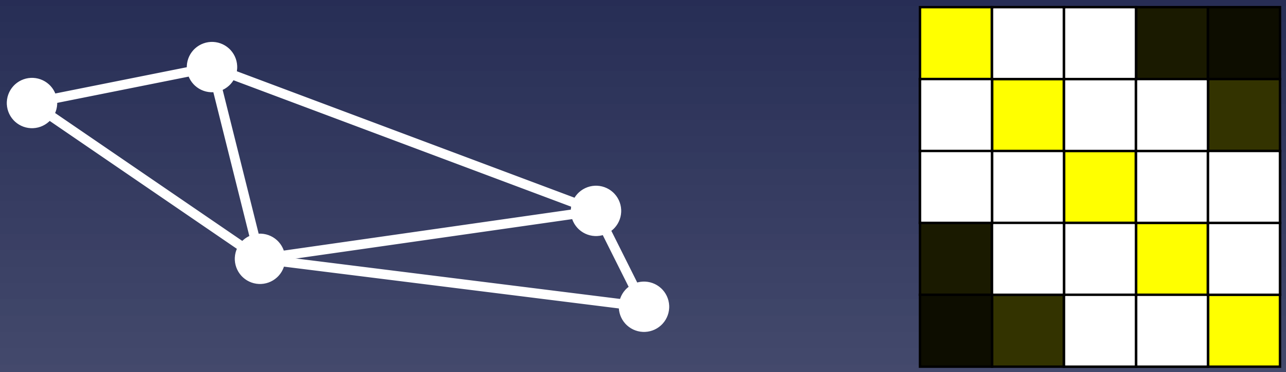
Network inference : the direct approach



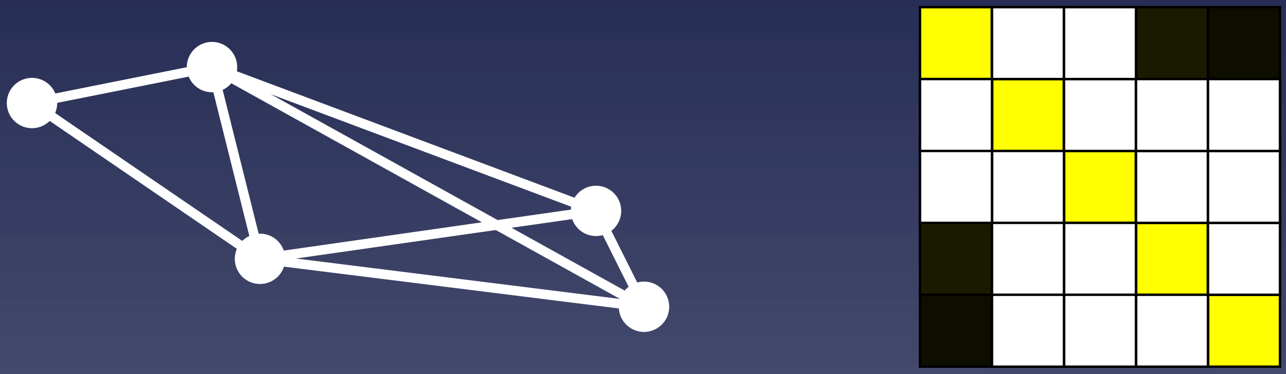
Network inference : the direct approach



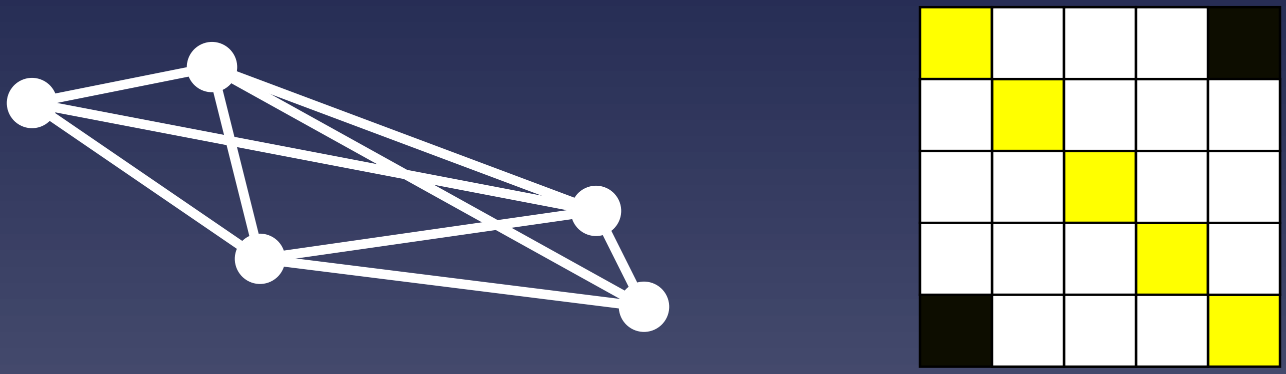
Network inference : the direct approach



Network inference : the direct approach

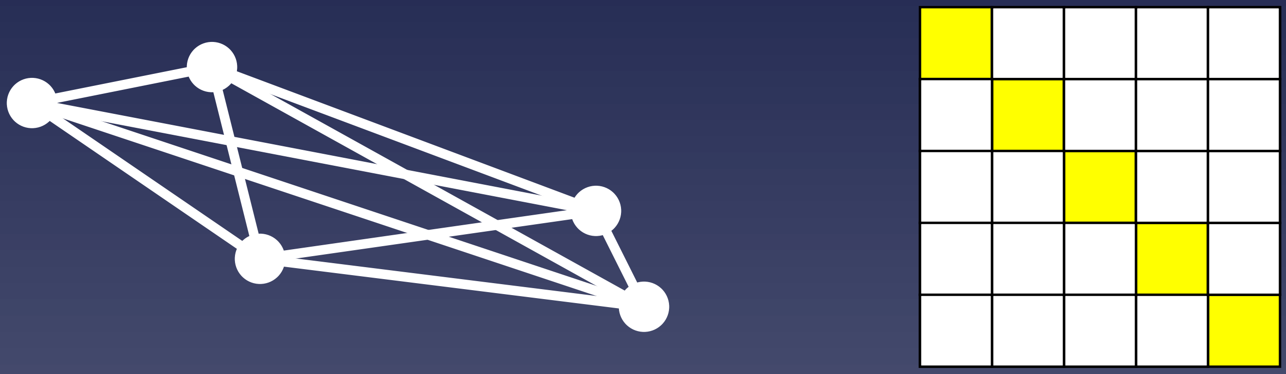


Network inference : the direct approach

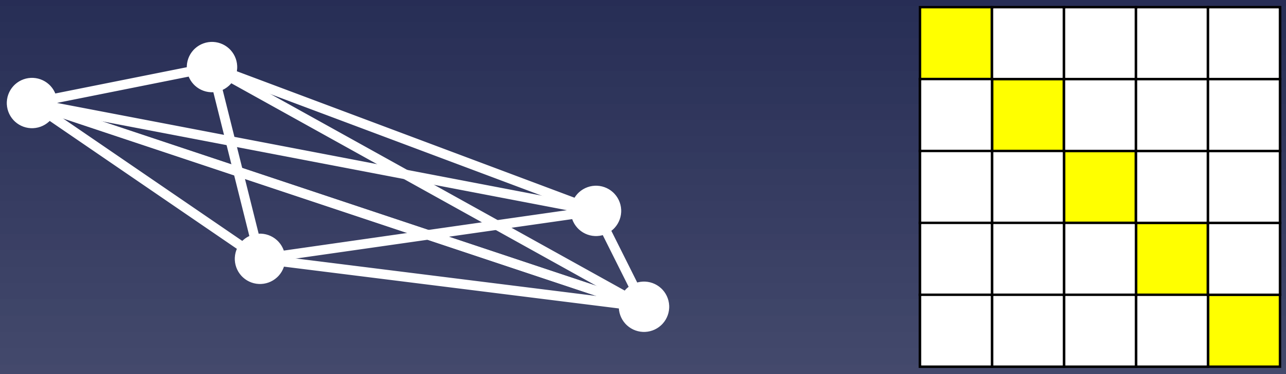


🟨	⬜	⬜	⬜	⬛
⬜	🟨	⬜	⬜	⬜
⬜	⬜	🟨	⬜	⬜
⬜	⬜	⬜	🟨	⬜
⬛	⬜	⬜	⬜	🟨

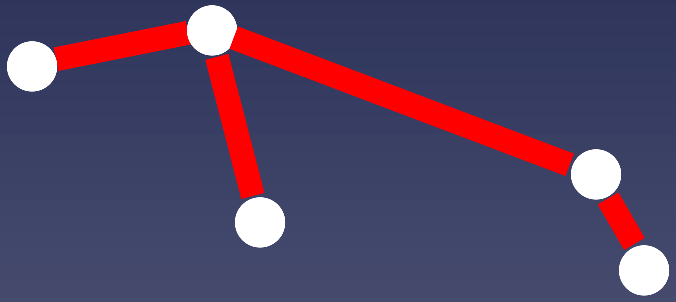
Network inference : the direct approach



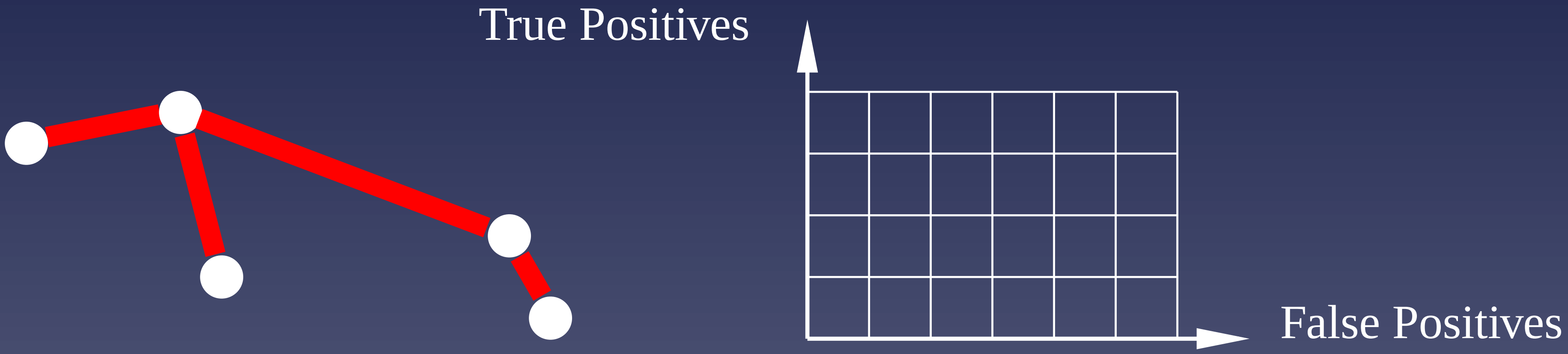
Network inference : the direct approach



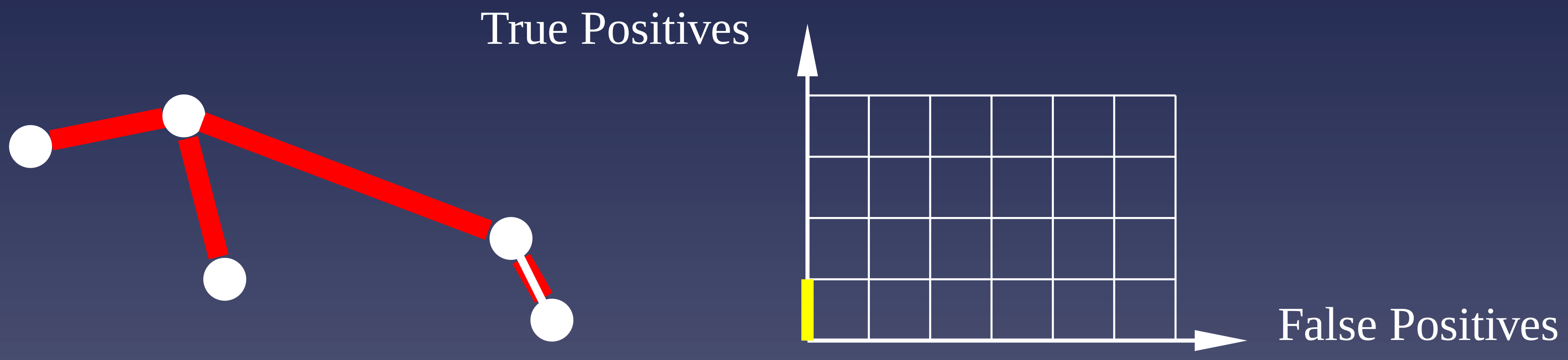
Evaluation of the performance : the ROC curve



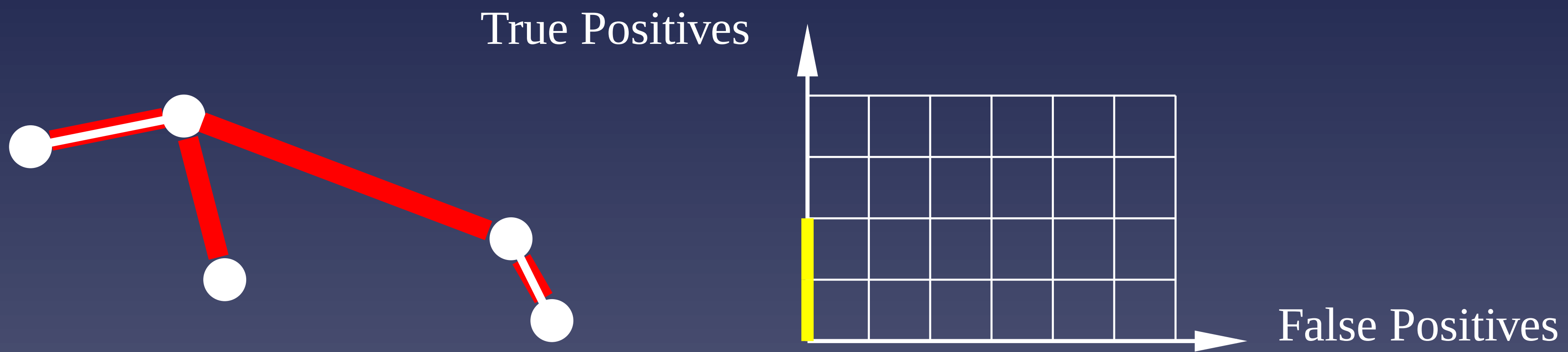
Evaluation of the performance : the ROC curve



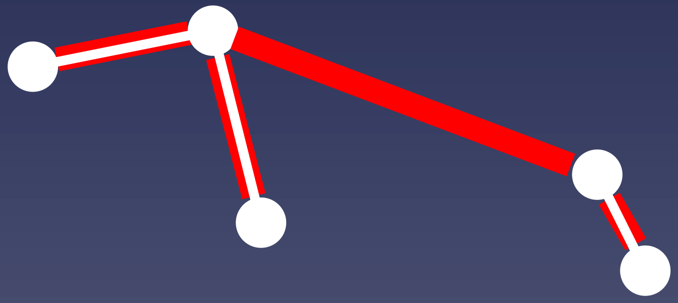
Evaluation of the performance : the ROC curve

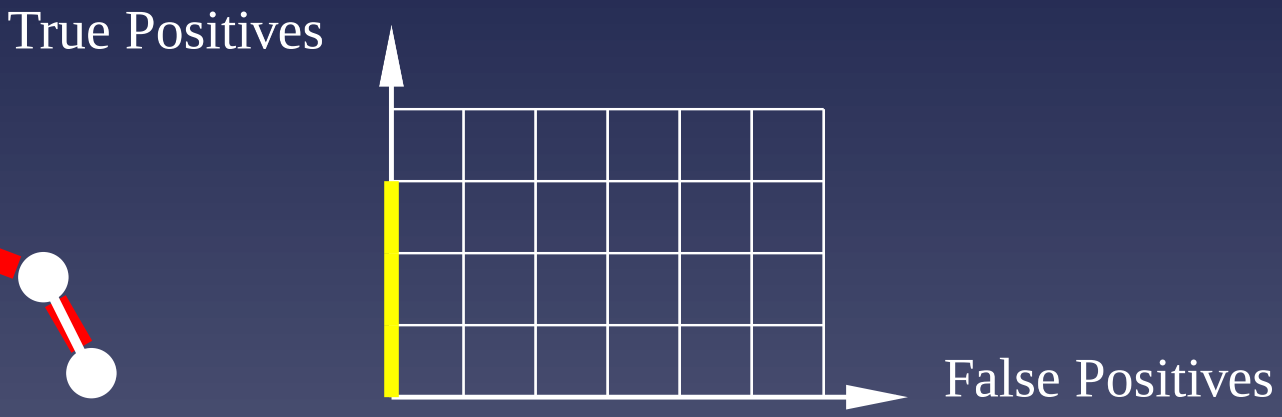


Evaluation of the performance : the ROC curve

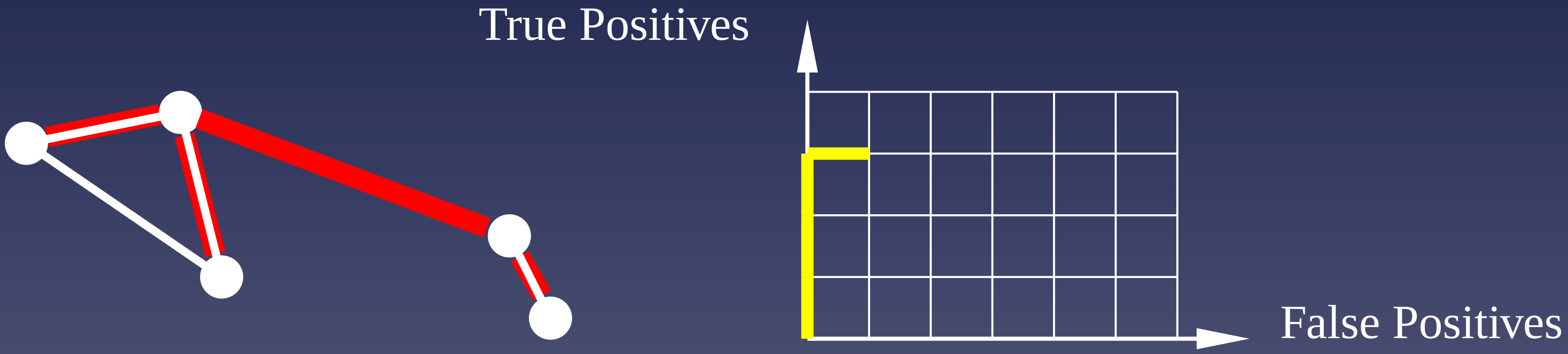


Evaluation of the performance : the ROC curve

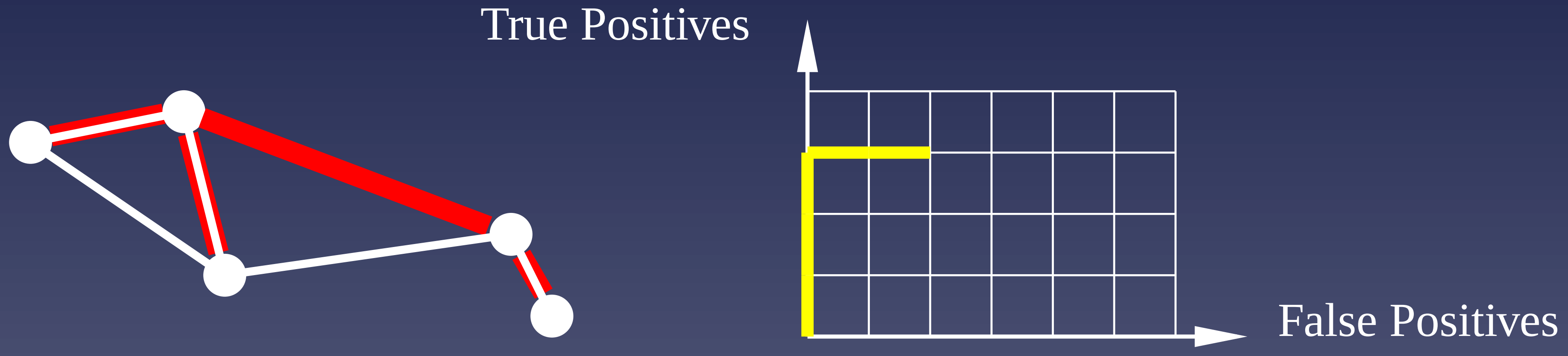




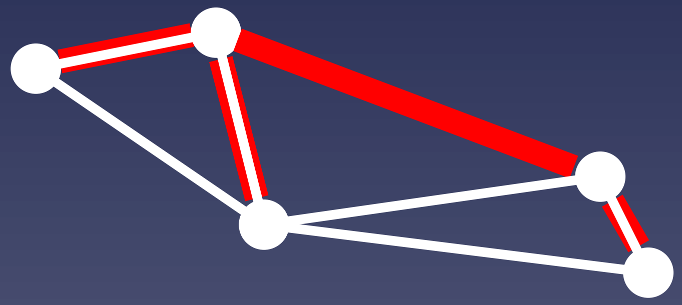
Evaluation of the performance : the ROC curve

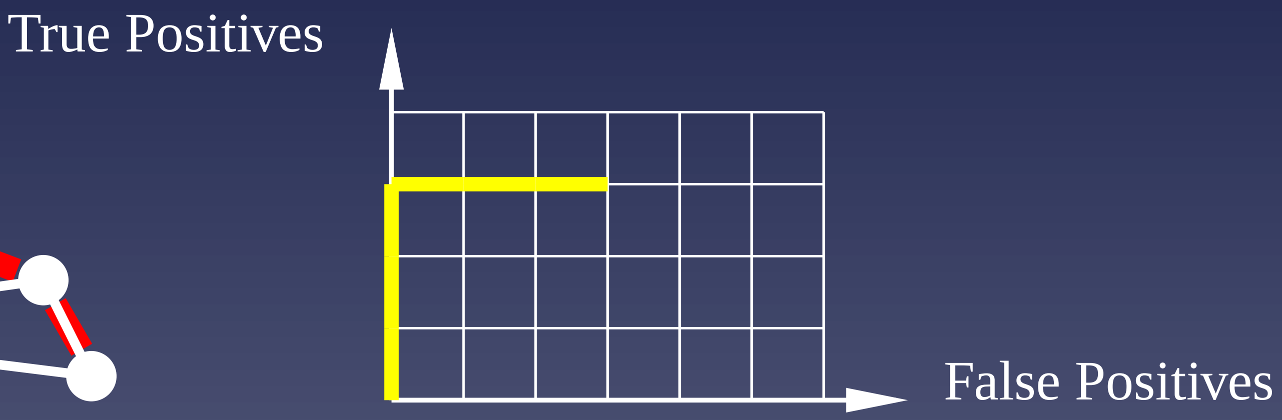


Evaluation of the performance : the ROC curve

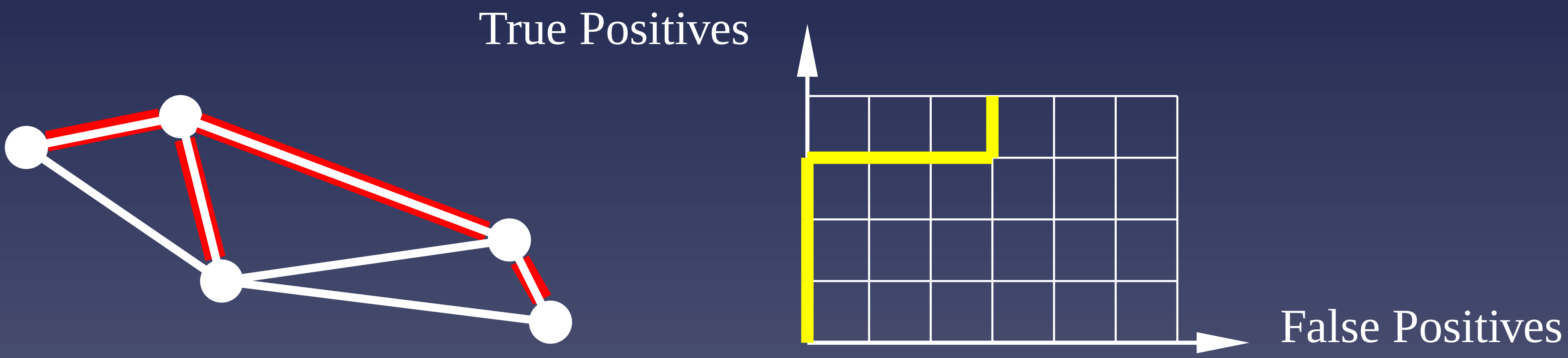


Evaluation of the performance : the ROC curve

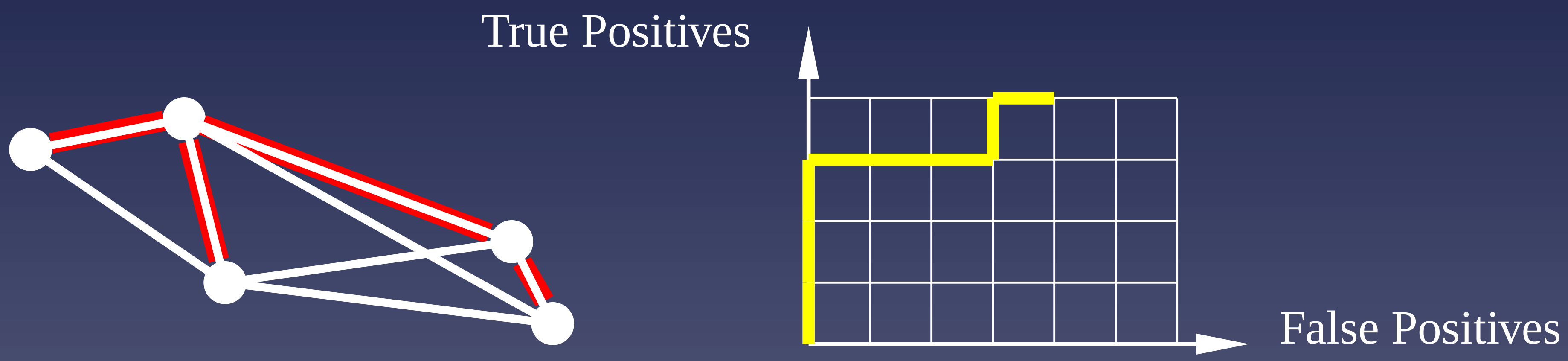




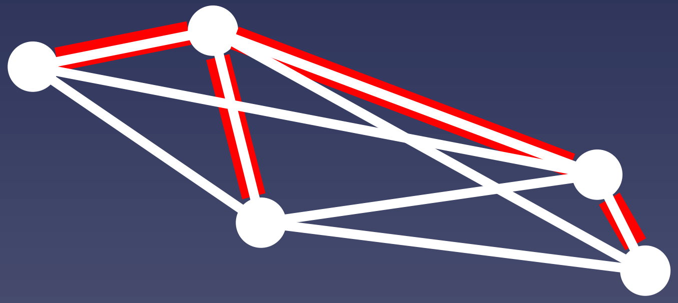
Evaluation of the performance : the ROC curve

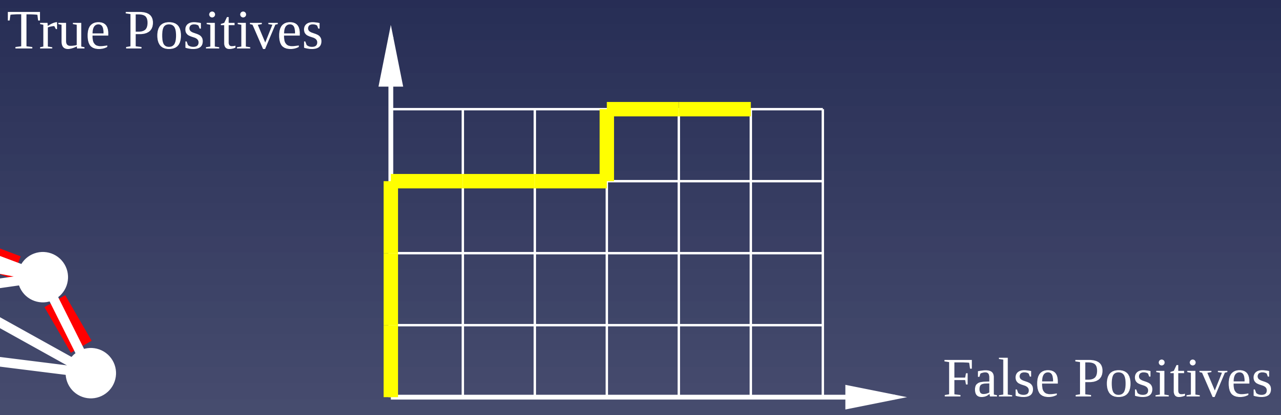


Evaluation of the performance : the ROC curve

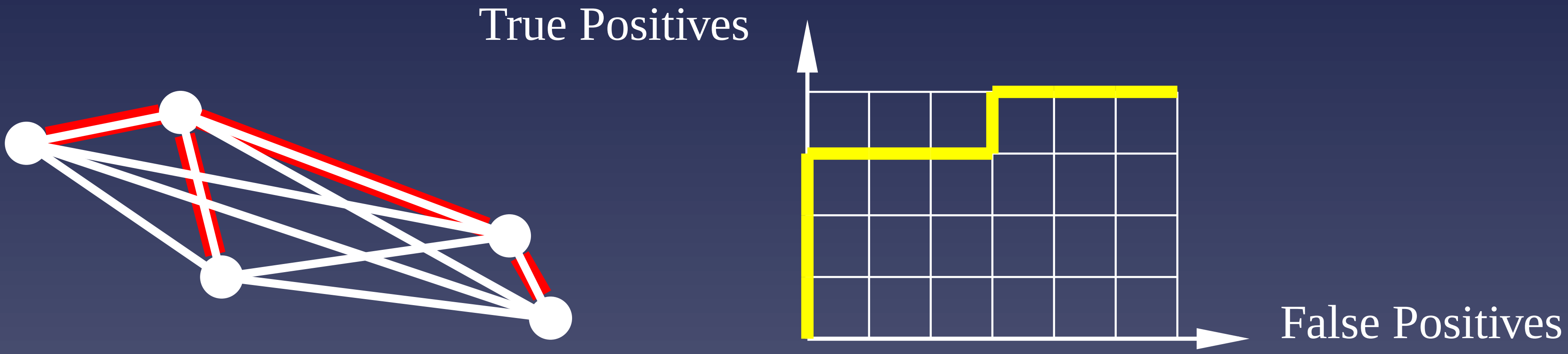


Evaluation of the performance : the ROC curve



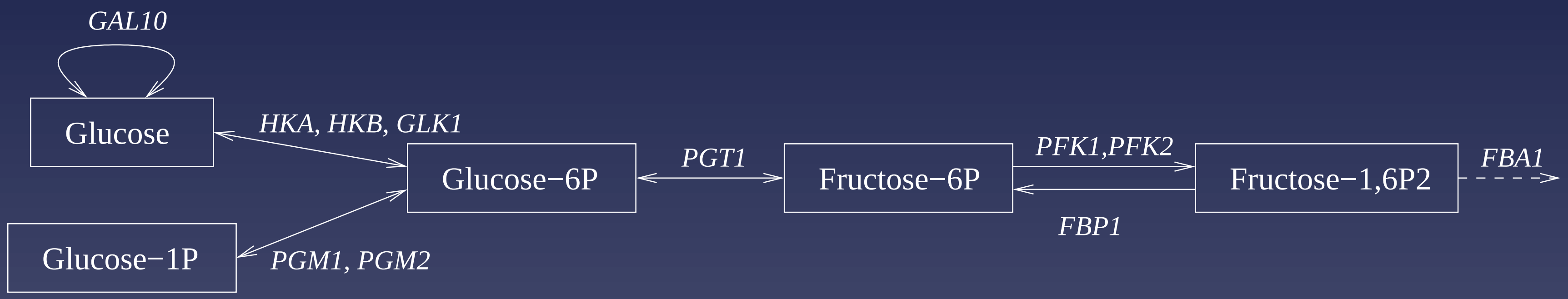


Evaluation of the performance : the ROC curve



$$ROC = 21/24 = 87,5\%$$

Application: the metabolic gene network



GAL10

Glucose

HKA, HKB, GLK1

Glucose−6P

PGT1

Fructose−6P

PFK1,PFK2

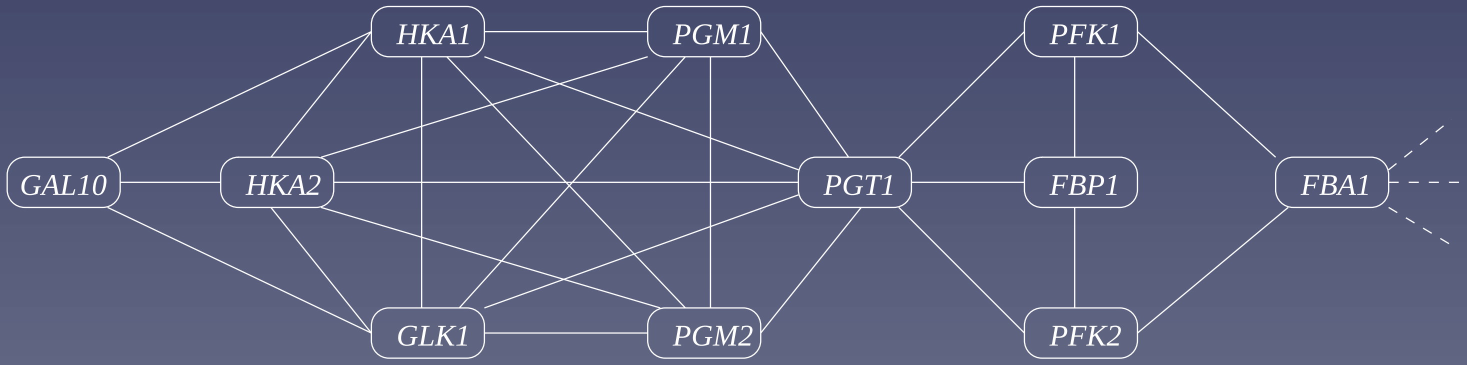
Fructose−1,6P2

FBA1

Glucose−1P

PGM1, PGM2

FBP1



HKA1

PGM1

PFK1

GAL10

HKA2

PGT1

FBP1

FBA1

GLK1

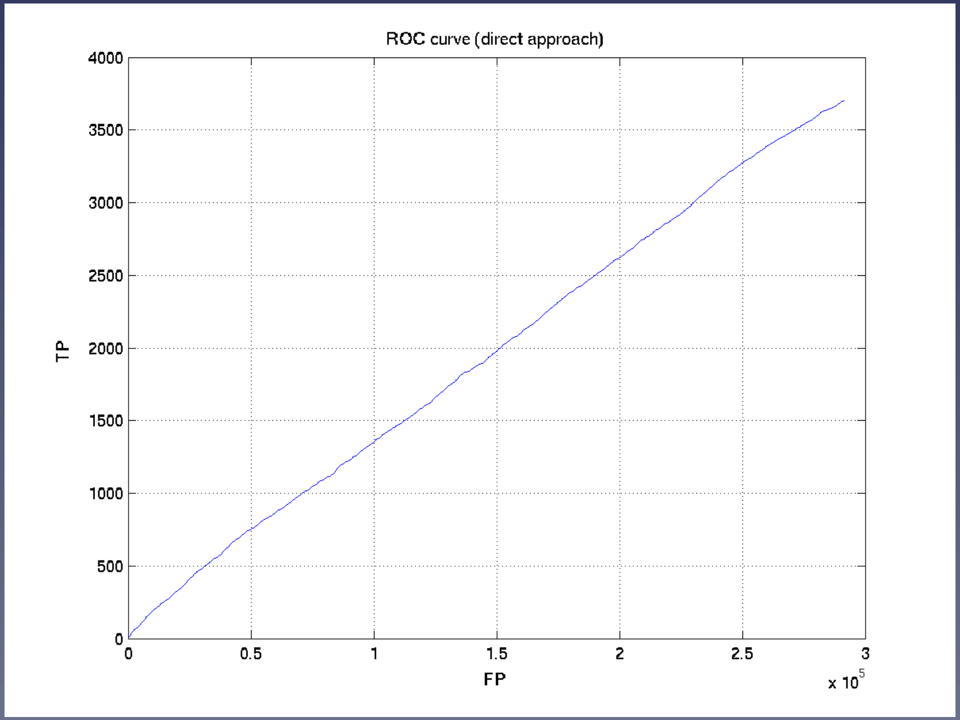
PGM2

PFK2

Link two genes when they can catalyze two successive reactions

Evaluation of the direct approach

The metabolic network of the yeast involves 769 genes. Each gene is represented by 157 expression measurements. (ROC=0.52)



Shortcuts of the direct approach

- What similarity measure between profiles should be use?

Shortcuts of the direct approach

- What similarity measure between profiles should be use?
- Which network are we expecting to recover?

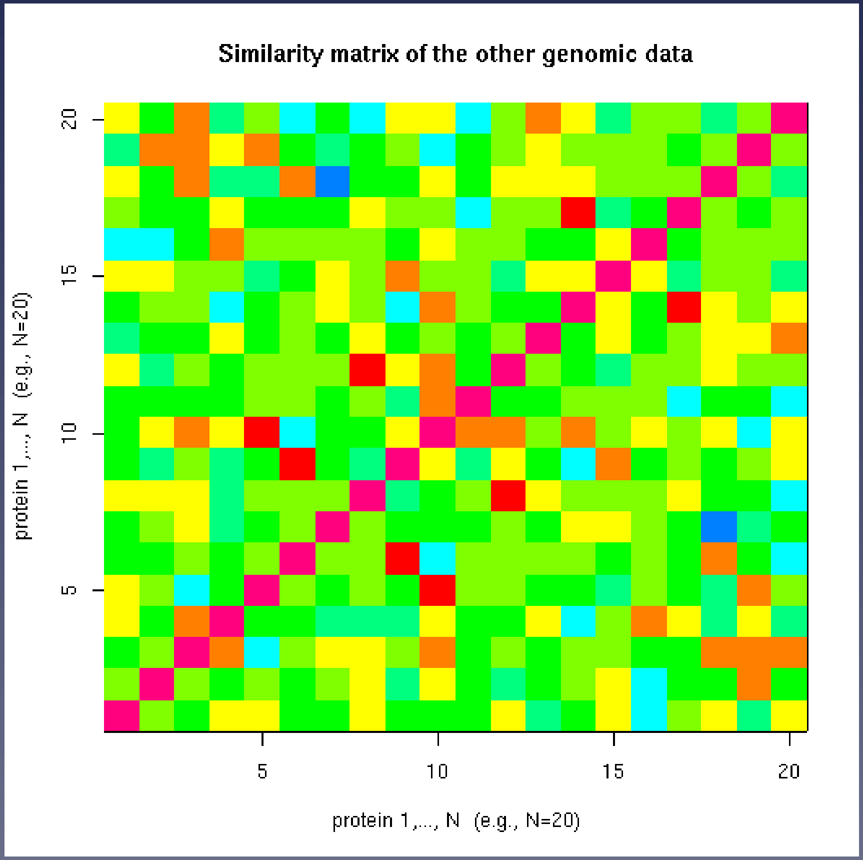
Shortcuts of the direct approach

- What similarity measure between profiles should be use?
- Which network are we expecting to recover?
- How to use prior knowledge about the network to be recovered?

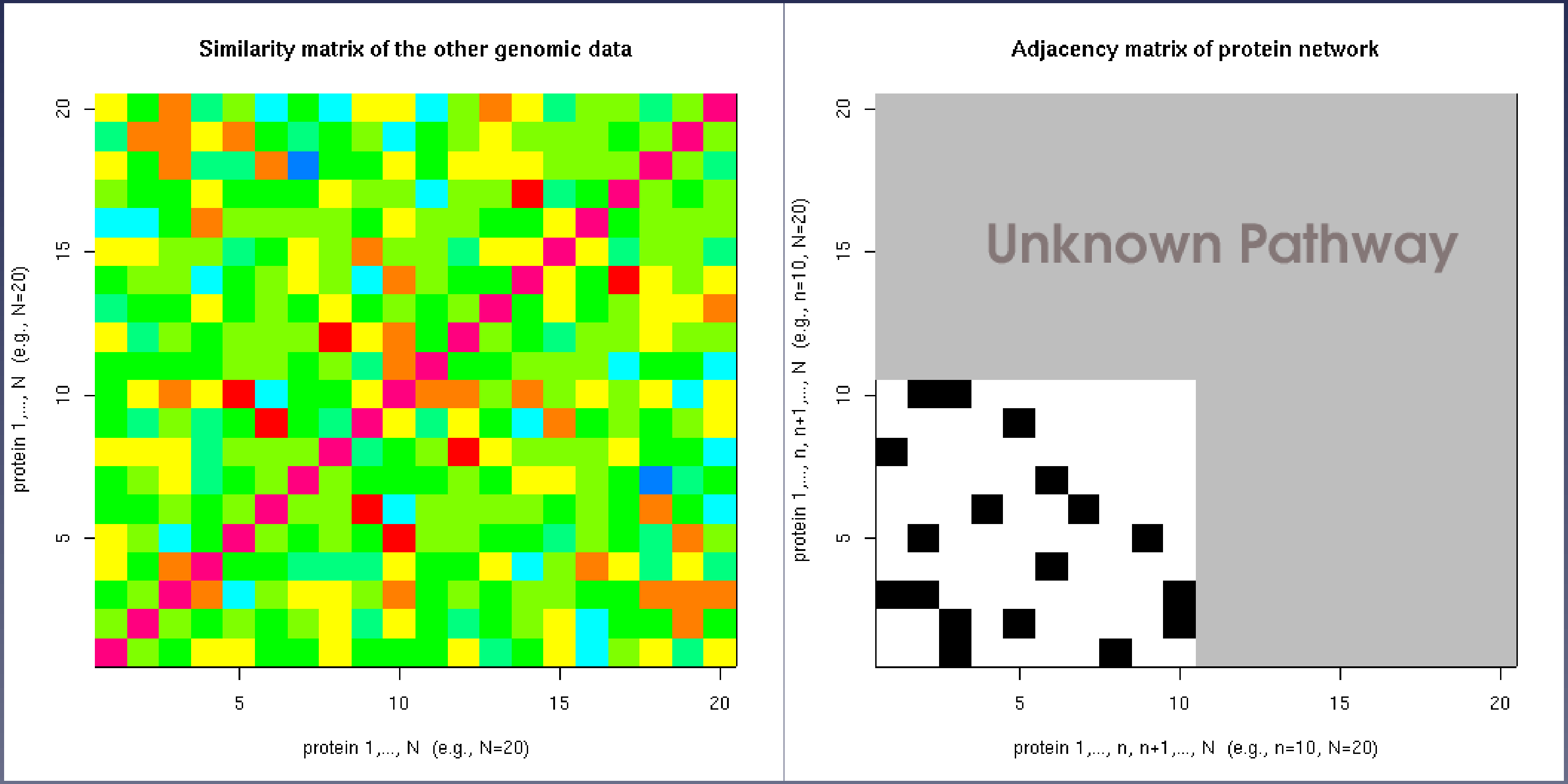
Part 2

Supervised network inference

The supervised gene inference problem



The supervised gene inference problem



The idea in a nutshell

- Use the known network to “learn” a more relevant measure of similarity

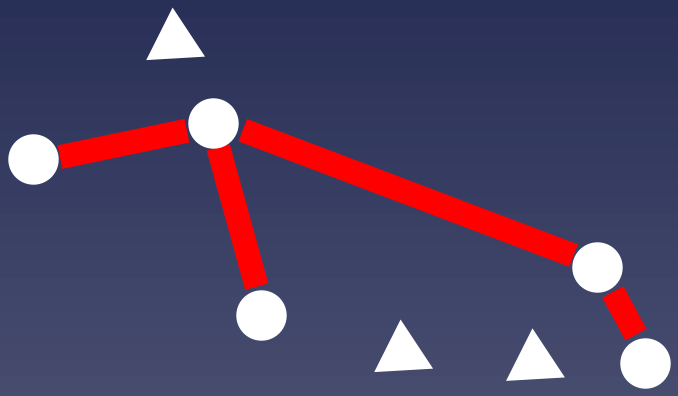
The idea in a nutshell

- Use the known network to “learn” a more relevant measure of similarity
- For example, map the genes expression profiles to a different space, where the natural distance better fits the known network

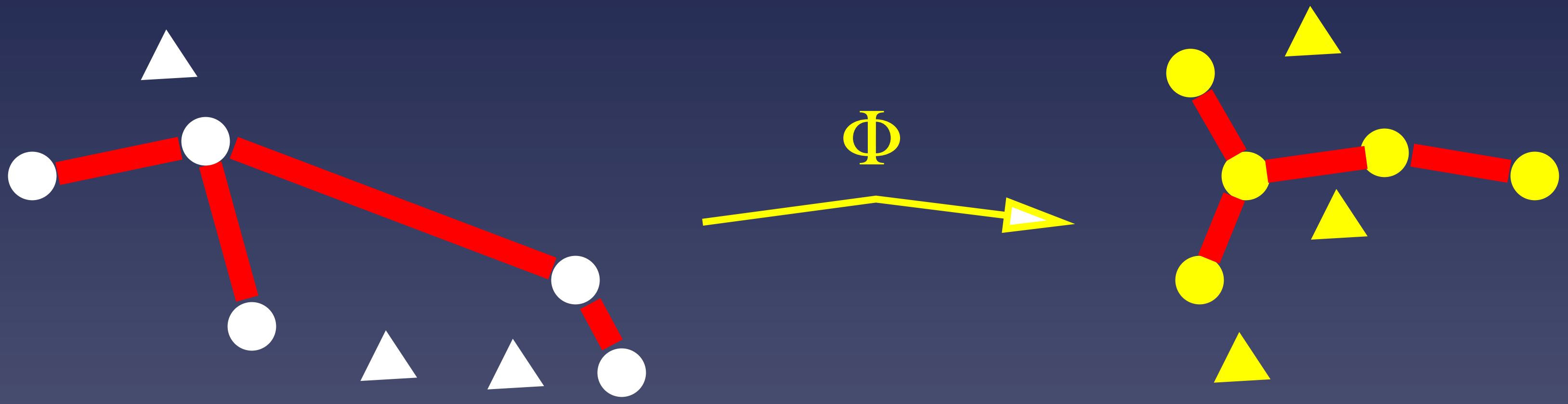
The idea in a nutshell

- Use the known network to “learn” a more relevant measure of similarity
- For example, map the genes expression profiles to a different space, where the natural distance better fits the known network
- Then apply the direct strategy in the second space

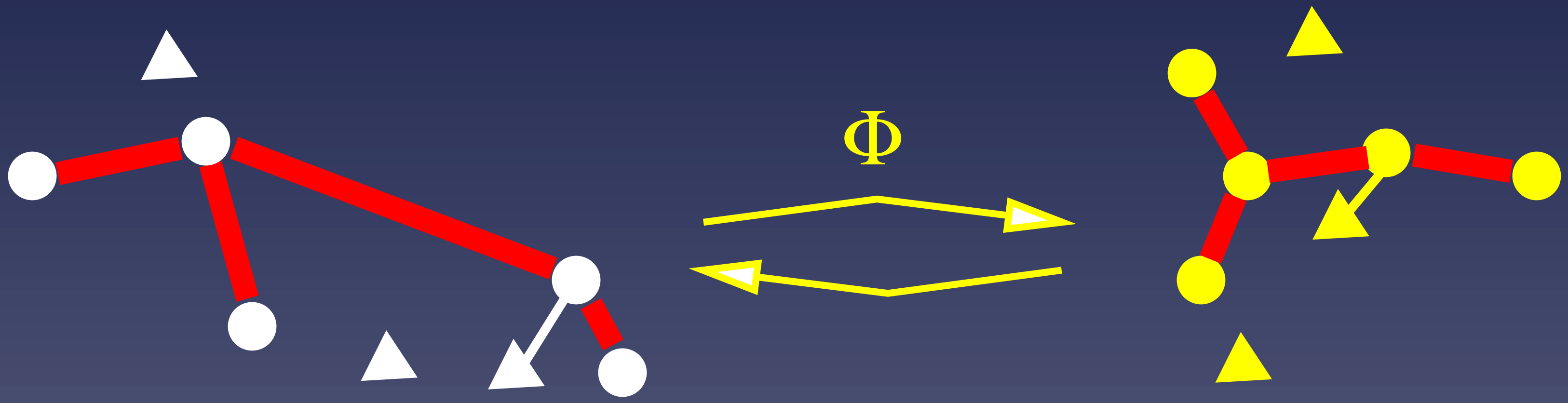
Illustration



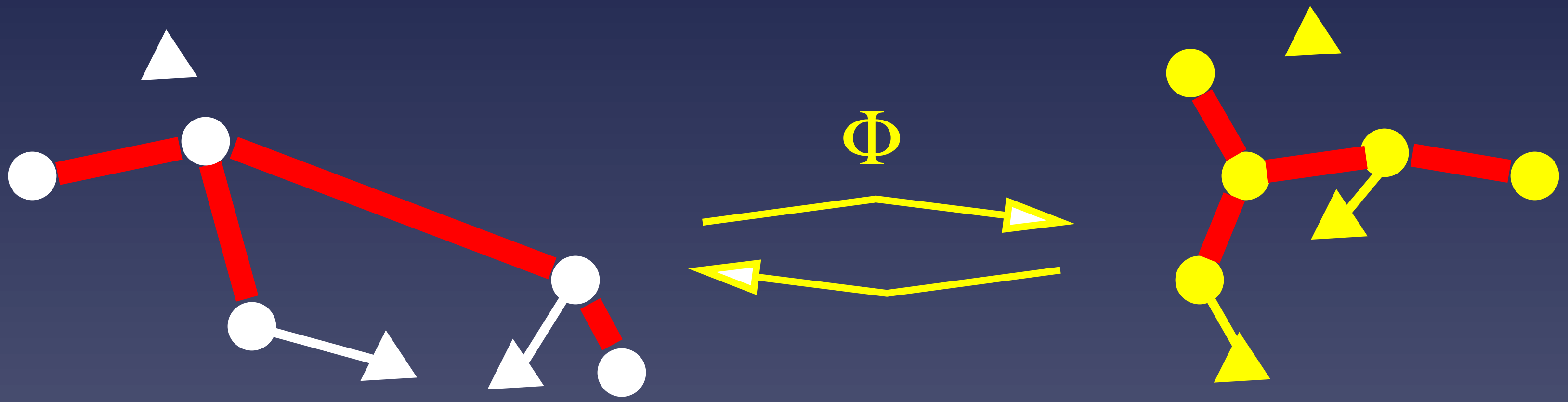
Illustration



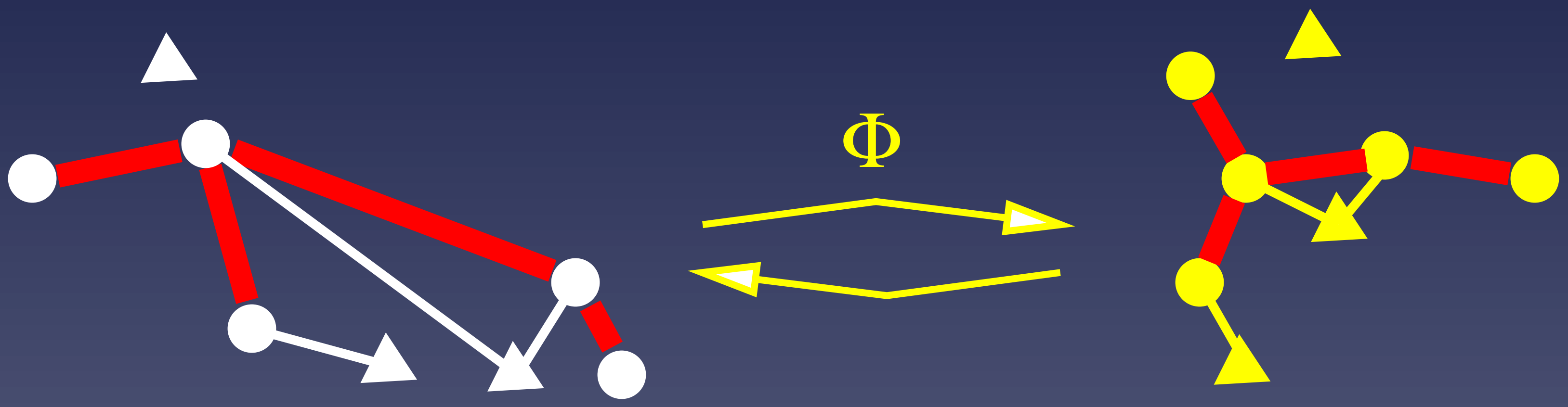
Illustration



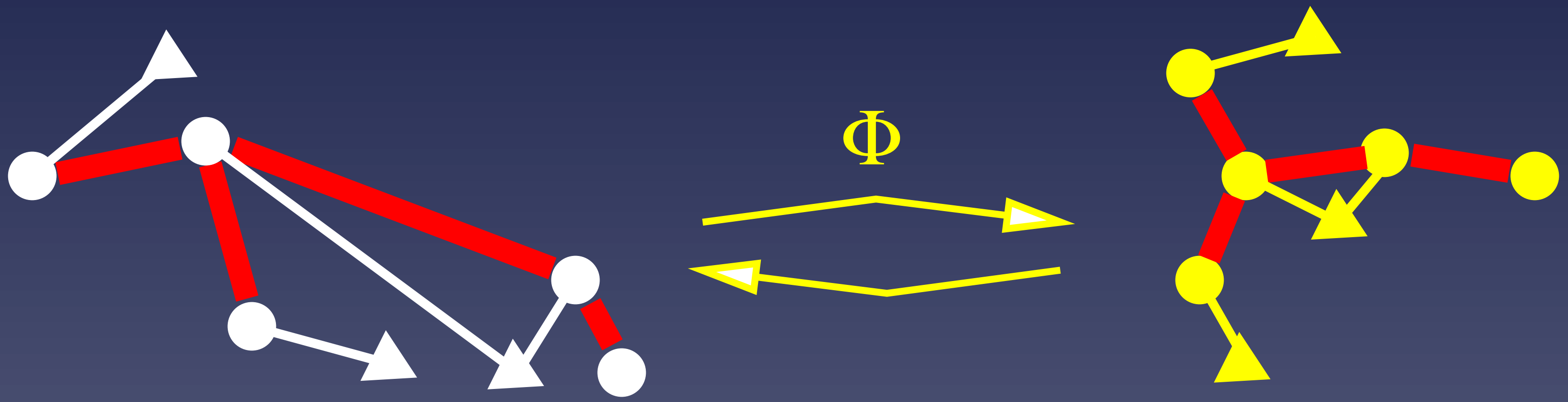
Illustration



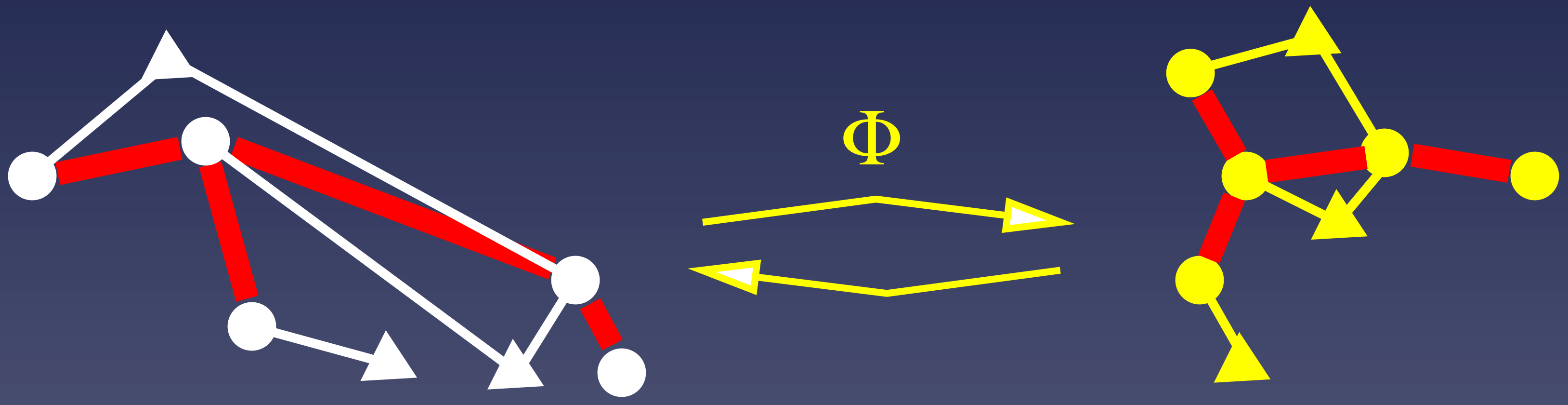
Illustration



Illustration



Illustration



Learning the mapping Φ

- Let $x \in \mathbb{R}^p$ be an expression profile

Learning the mapping Φ

- Let $x \in \mathbb{R}^p$ be an expression profile
- Let us consider linear mappings:

$$\Phi(x) = (f_1(x), \ldots, f_d(x))' \in \mathbb{R}^d$$

made of linear features $f_i(x) = w_i^\top x$

Learning the mapping Φ

- Let $x \in \mathbb{R}^p$ be an expression profile
- Let us consider linear mappings:

$$\Phi(x) = (f_1(x), \dots, f_d(x))' \in \mathbb{R}^d$$

made of linear features $f_i(x) = w_i^\top x$

- A feature $f : \mathbb{R}^p \to \mathbb{R}$ is “good” if connected genes in the known network have similar value.

"Good" features

- A "good" feature $f(x) = w^\top x$ should minimize:

$$R(f) = \frac{\sum_{i \sim j} \left(f(x_i) - f(x_j)\right)^2}{\sum_{i=1}^n f(x_i)^2},$$

- Regularisation: for statistical reasons, it is safer to minimize:

$$\min_{f(x)=w^\top x} \frac{\sum_{i \sim j} \left(f(x_i) - f(x_j)\right)^2 + \lambda ||w||^2}{\sum_{i=1}^n f(x_i)^2},$$

Influence of λ

- $\lambda \rightarrow +\infty$: PCA
  - ⋆ Useful for noisy, high-dimensional data.
  - ⋆ Used in spectral clustering. The graph does not play any role (unsupervised)
- $\lambda \rightarrow 0$: second smallest eigenvector of the graph
  - ⋆ Useful to embed the graph in a Euclidean space (used in graph partitioning)
  - ⋆ Sensitive to noise. Mapping of points outside of the graph unstable (overfitting)

Extracting successive features

- Successive features to form Φ can be obtained by:

$$w_i = \mathop{\arg\min}_{w \perp \{w_1, \ldots, w_{i-1}\}, \hat{\text{var}}(f_w)=1} \left\{ \sum_{i \sim j} \left(f_w(x_i) - f_w(x_j)\right)^2 + \lambda ||w||^2 \right\}.$$

Extracting successive features

- Successive features to form Φ can be obtained by:

$$w_i = \underset{w \perp \{w_1, \ldots, w_{i-1}\}, \hat{\mathrm{var}}(f_w)=1}{\arg\min} \left\{ \sum_{i \sim j} (f_w(x_i) - f_w(x_j))^2 + \lambda ||w||^2 \right\}.$$

- Generalizes Principal Component Analysis (PCA)

Extension to non-linear features

Let us now only suppose that $\mathcal{X}$ is a set endowed with a symmetric positive definite kernel $k : \mathcal{X}^2 \to \mathbb{R}$, i.e.,

$$\sum_{i=1}^{n} \sum_{j=1}^{n} c_i c_j k(x_i, x_j) \geq 0$$

for any $n \geq 0, (x_1, \ldots, x_n) \in \mathcal{X}$ and $(a_1, \ldots, a_n) \in \mathbb{R}$
Examples:

- $k(x, y) = \exp(-||x - y||^2/(2\sigma^2))$ for $\mathcal{X} = \mathbb{R}^d$
- string and tree kernels (Watkins 99, Haussler 99, Saigo et al. 04), phylogenetic tree kernel (Vert 02), Fisher kernel (Jaakkola et al 00), ...

Features and RKHS

- A p.d. kernel defines a Hilbert space of functions $f : \mathcal{X} \rightarrow \mathbb{R}$ obtained by completing the span of $\{k(x, \cdot), x \in \mathcal{X}\}$

- The norm of a function $f(x) = \sum_{i=1}^{n} c_i k(x_i, x)$ is:

$$||f||_k^2 = \sum_{i,j=1}^{n} c_i c_j k(x_i, x_j).$$

- This functional space $\mathcal{H}_k$ is called the reproducing kernel Hilbert space (RKHS).

Example : Kernel PCA

- For $\mathcal{X} = \mathbb{R}^d$, let $k(x, y) = x \cdot y$ (linear kernel). Then the hilbert space of functions $\mathcal{H}_k$ is the set of linear functions $f_w(x) = w \cdot x$ with norm:

$$||f||_k^2 = ||w||^2$$

- PCA can therefore be reformulated as:

$$f_{i=} \operatorname*{arg\,min}_{f \perp \{f_1, \dots, f_{i-1}\}, \hat{\text{var}}(f)=1} ||f||_k^2.$$

Graph-driven feature extraction in RKHS

- For a general set $\mathcal{X}$ endowed with a p.d. kernel k we therefore have the following graph-driven feature extractor:

$$f_i = \operatorname*{arg\,min}_{f \perp \{f_1,\ldots,f_{i-1}\}, \hat{\mathrm{var}}(f)=1} \left\{ \sum_{i \sim j} \left(f(x_i) - f(x_j) \right)^2 + \lambda ||f||_k^2 \right\}.$$

- The values at the minima (the spectrum) quantifies how much the graph fits the data

Solving the problem

- By the representer theorem, f_i can be expanded as:

$$f_i(x) = \sum_{j=1}^{n} \alpha_{i,j} k(x_i, x).$$

- This shows that

$$\begin{aligned} < f_i, f_j >_k &= \alpha_i^\top K \alpha_j \\ ||f_i||_k^2 &= \alpha_i^\top K \alpha_i \end{aligned} \tag{1}$$

Solving the problem (cont.)

- The problem can then be rewritten:

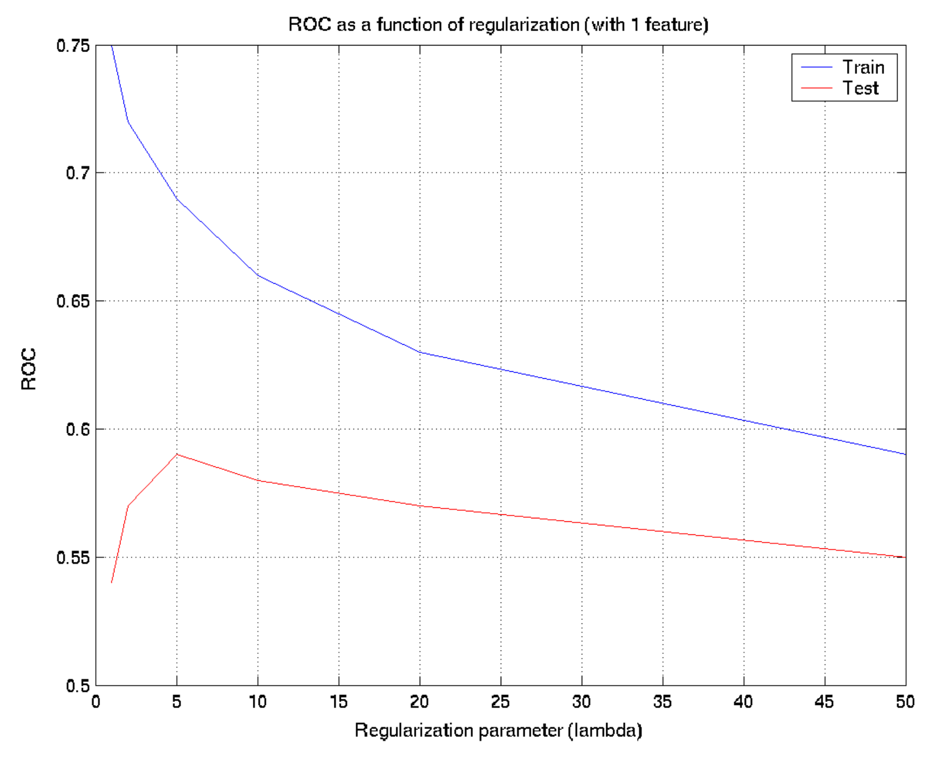
$$\alpha_i = \underset{\alpha \in \mathbb{R}^n, \alpha K_V \alpha_1 = \ldots = \alpha K_V \alpha_{i-1} = 0}{\arg\min} \left\{ \frac{\alpha^\top K_V L K_V \alpha + \lambda \alpha^\top K_V \alpha}{\alpha^\top K_V^2 \alpha} \right\}$$

  where K_V is the centered $n \times n$ Gram matrix and L is the Laplacian of the graph

- It is equivalent to solving the generalized eigenvalue problem:

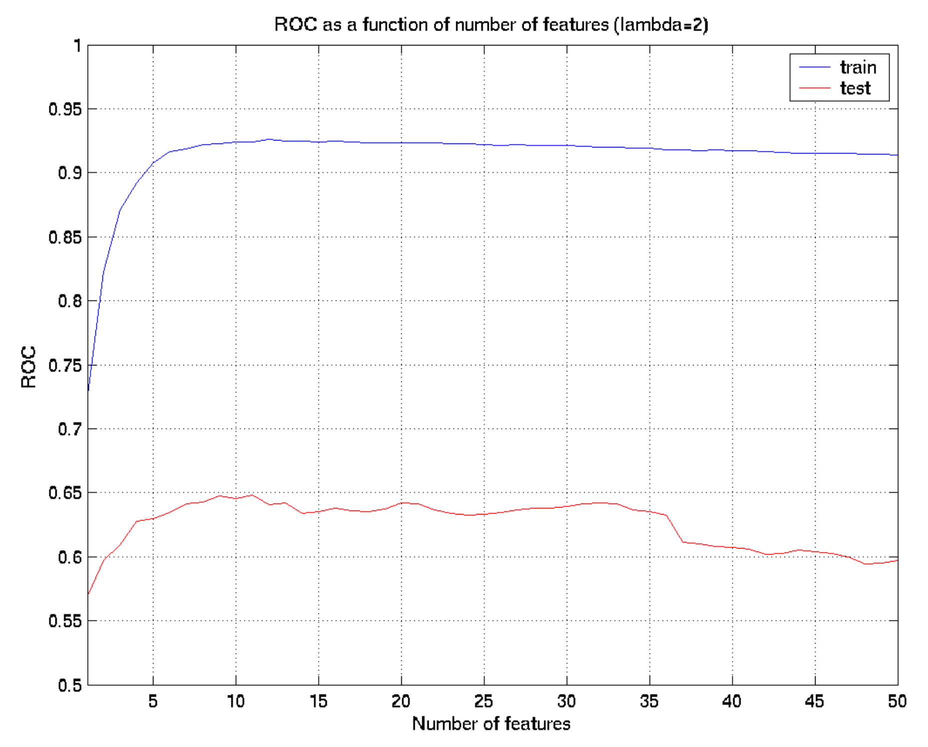
$$(LK_V + \lambda I)\alpha = \mu K_V \alpha.$$

Evaluation of the supervised approach: effect of λ



Metabolic network, 10-fold cross-validation, 1 feature

Evaluation of the supervised approach: number of features ($\lambda = 2$)



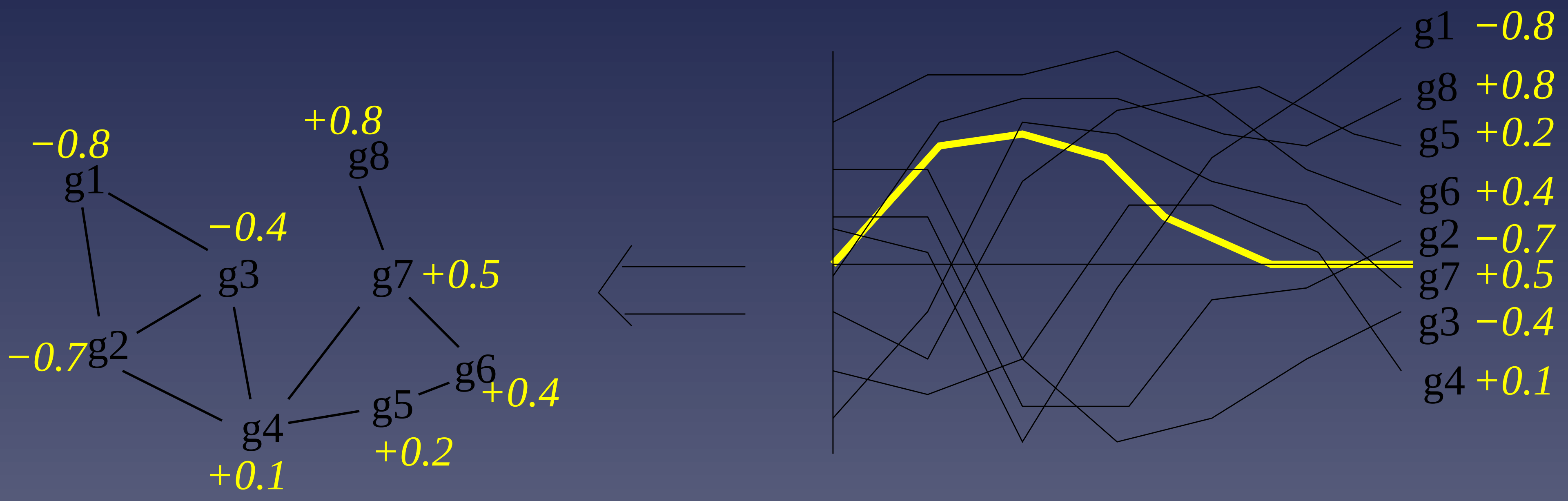
Part 3

Extraction of pathway activity

The idea

- The previous approach is a way to extract features from gene expression data: $f(x) = w^\top x$.
- These features are smooth on the graph: connected nodes tend to have similar values
- This is way to detect "correlations" between gene expression data and metabolic network : typical activity patterns of typical pathways

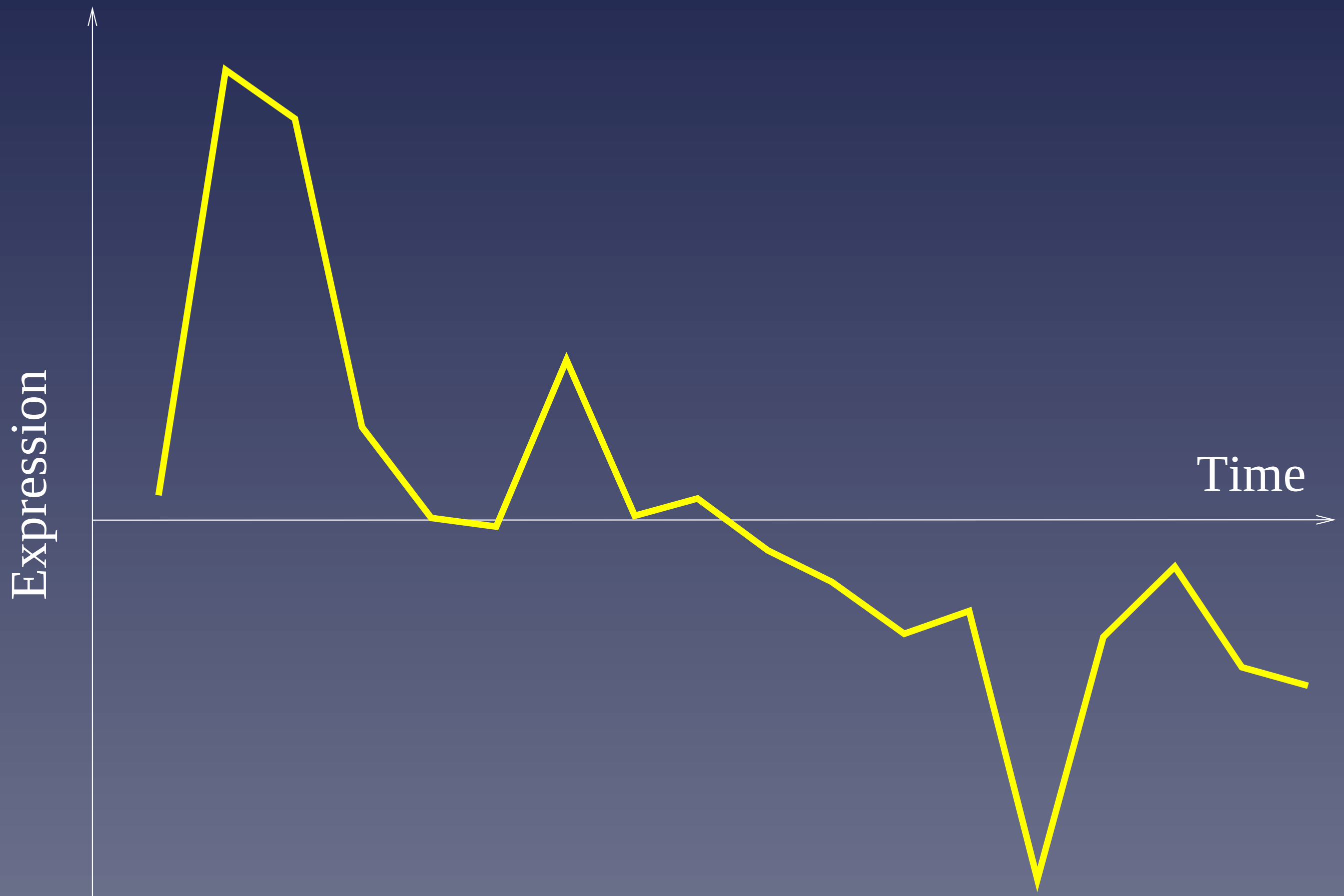
Illustration



Experiment

- Gene network: two genes are linked if the catalyze successive reactions in the KEGG database (669 yeast genes)
- Expression profiles: 18 time series measures for the 6,000 genes of yeast, during two cell cycles

First pattern of expression

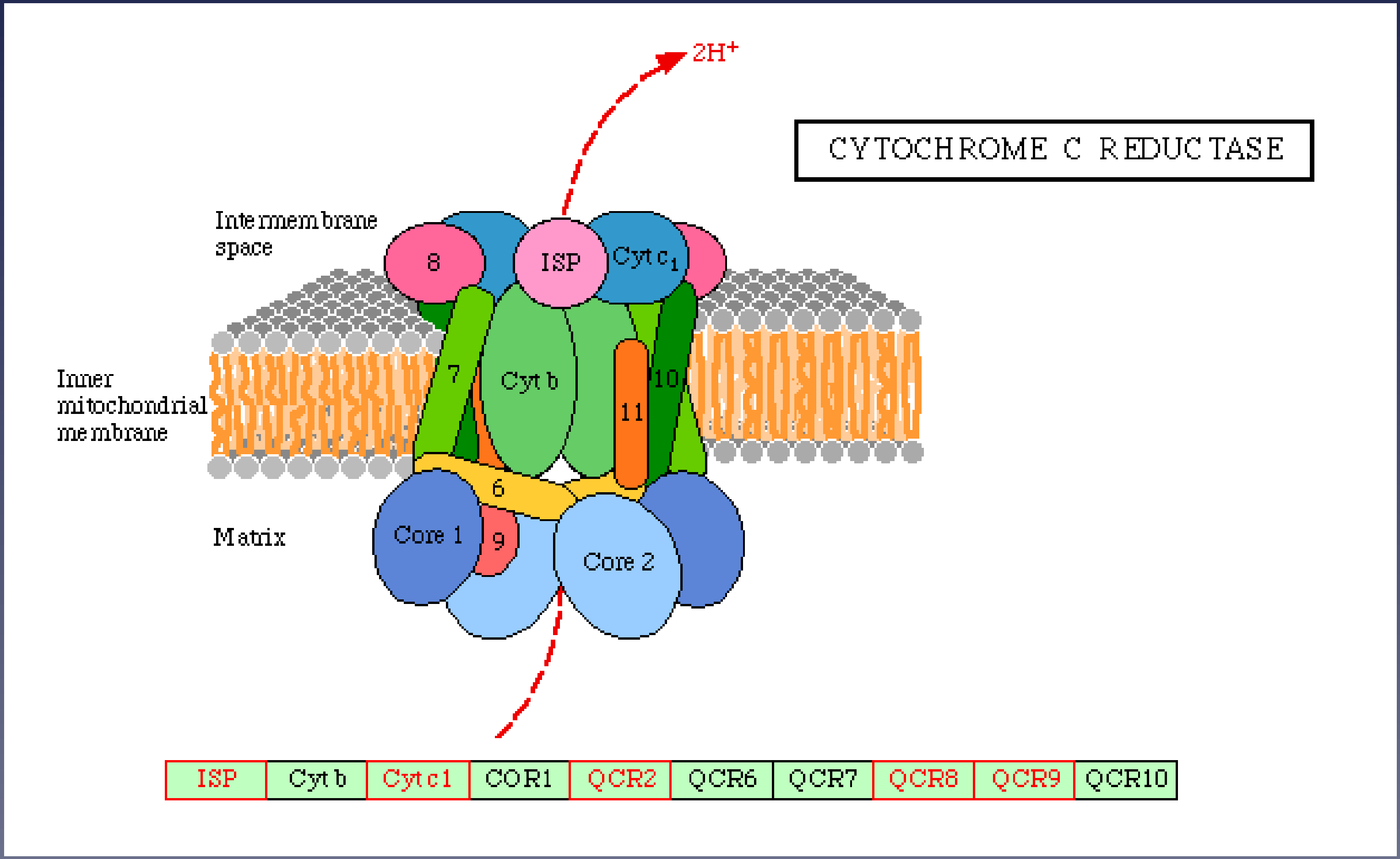


Related metabolic pathways

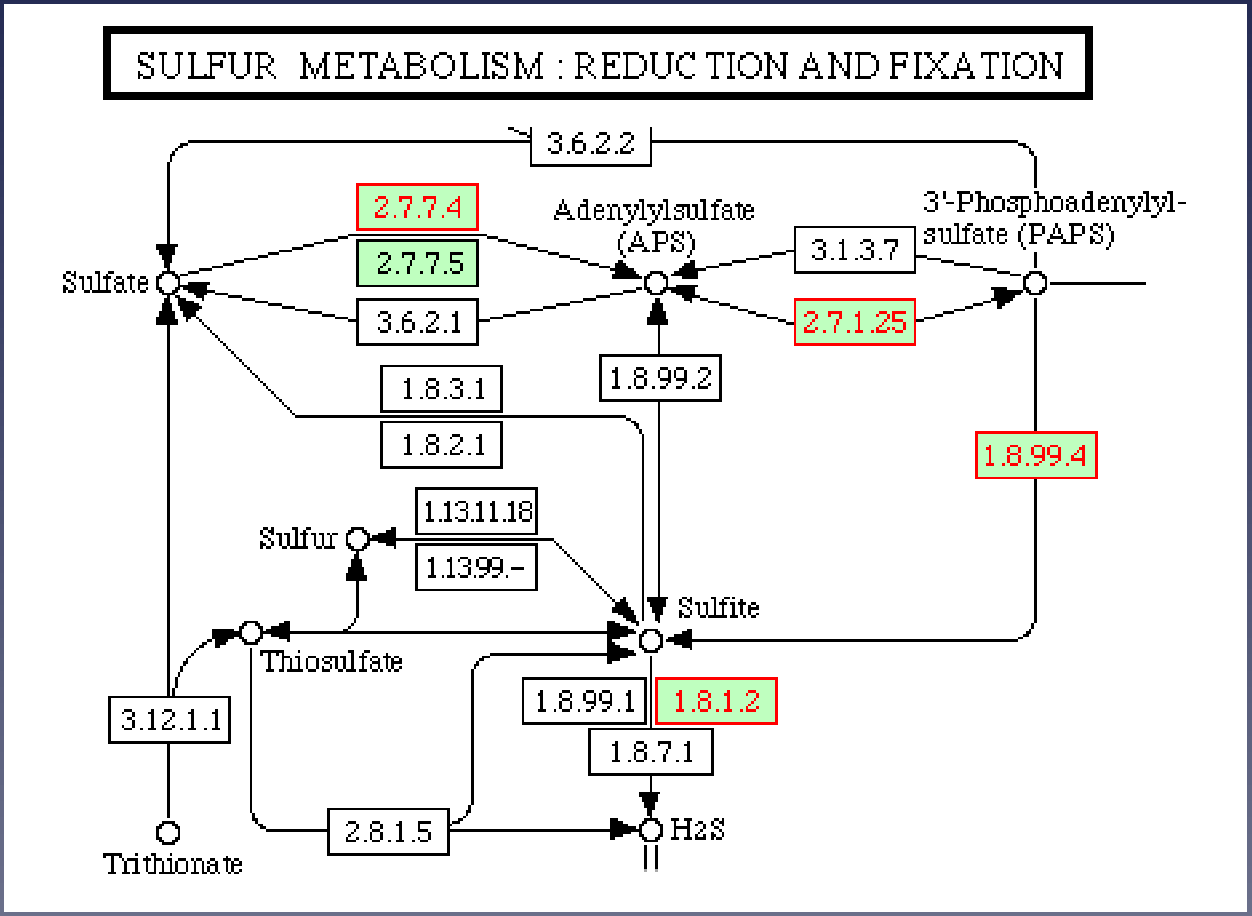
50 genes with highest $s_2 - s_1$ belong to:

- Oxidative phosphorylation (10 genes)
- Citrate cycle (7)
- Purine metabolism (6)
- Glycerolipid metabolism (6)
- Sulfur metabolism (5), etc...

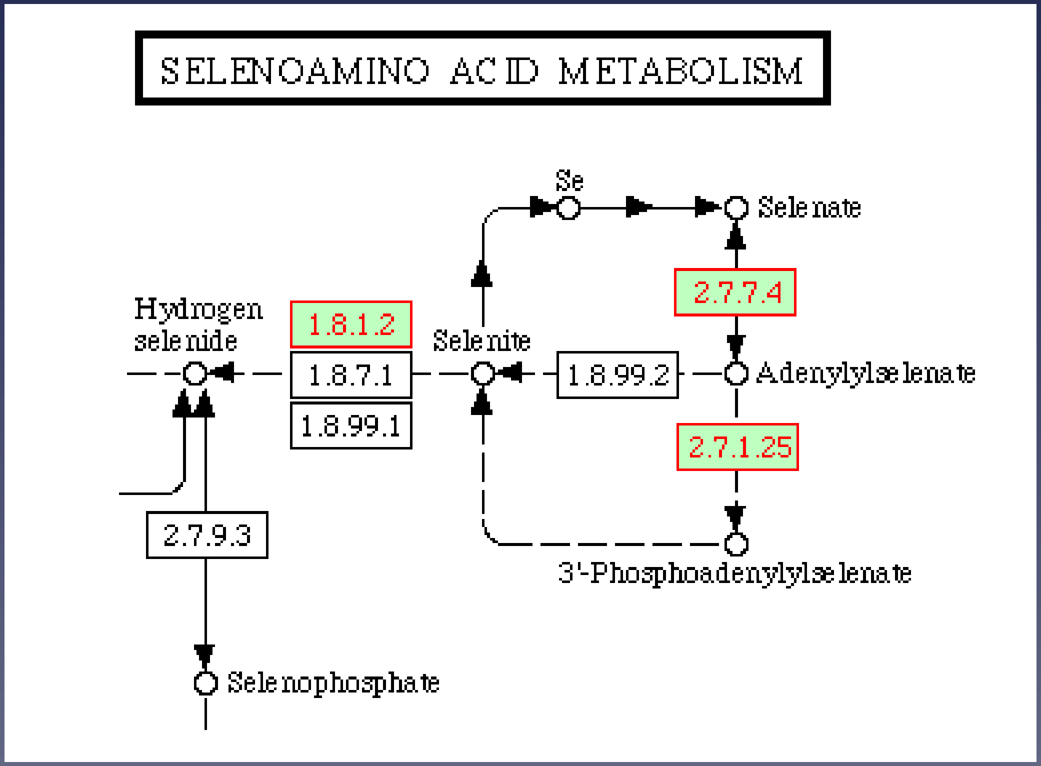
Related genes



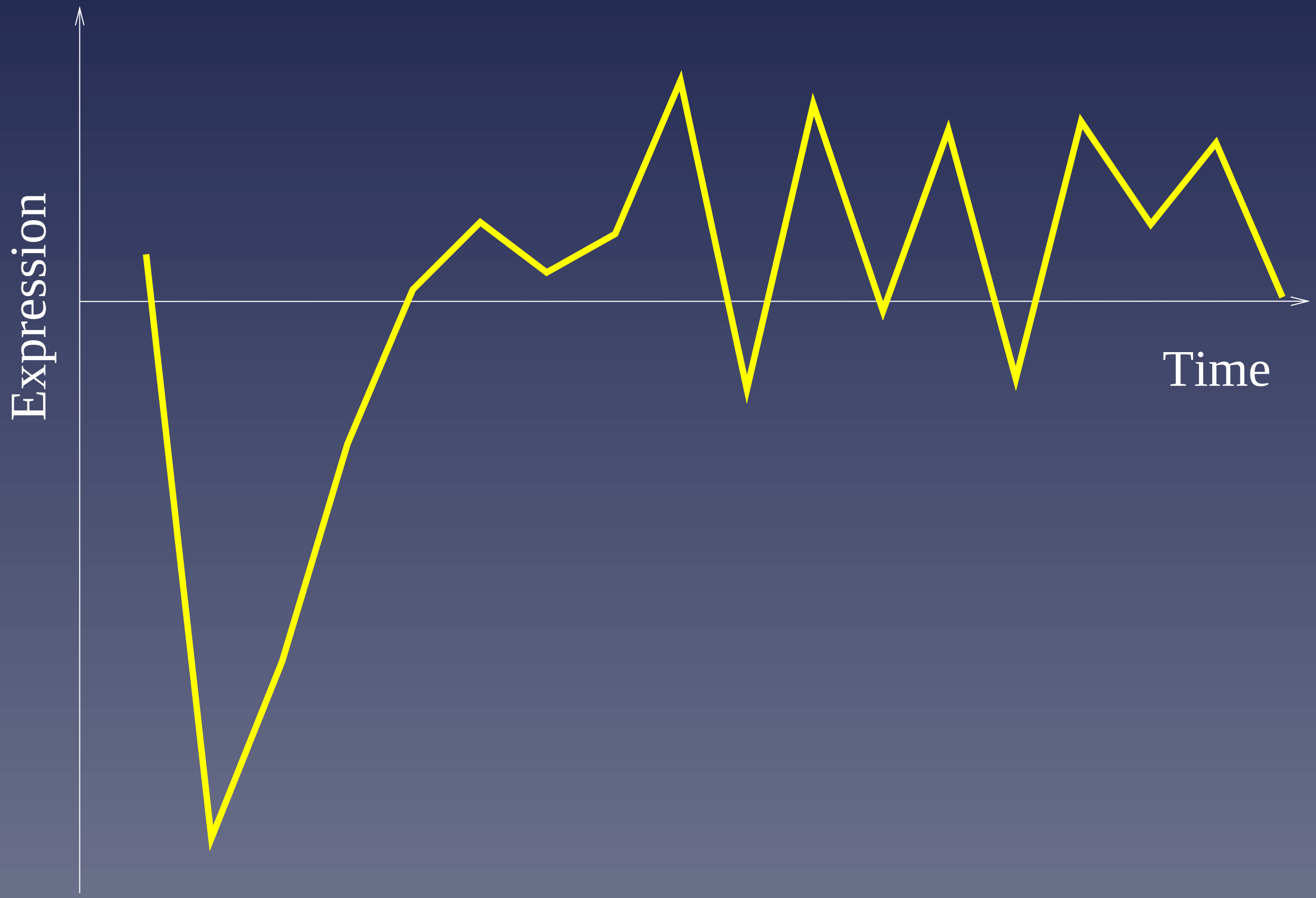
Related genes



Related genes



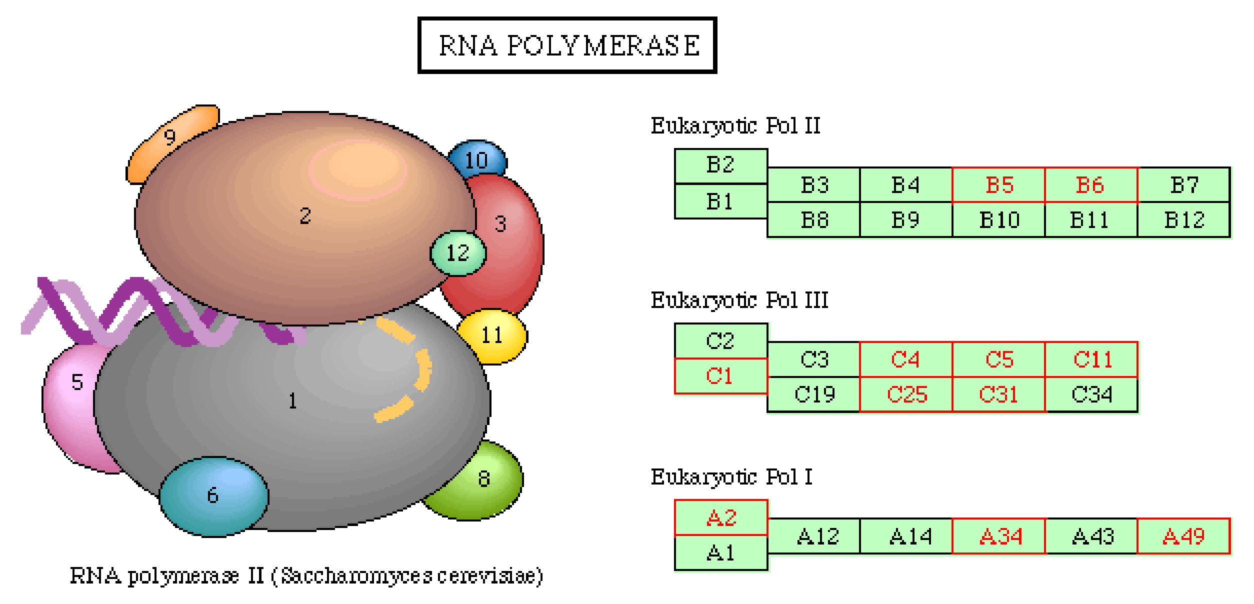
Opposite pattern



Related genes

- RNA polymerase (11 genes)
- Pyrimidine metabolism (10)
- Aminoacyl-tRNA biosynthesis (7)
- Urea cycle and metabolism of amino groups (3)
- Oxidative phosphorlation (3)
- ATP synthesis(3) , etc...

Related genes



RNA POLYMERASE

RNA polymerase II (Saccharomyces cerevisiae)

Eukaryotic Pol II

B2	B3	B4	B5	B6	B7
B1	B8	B9	B10	B11	B12

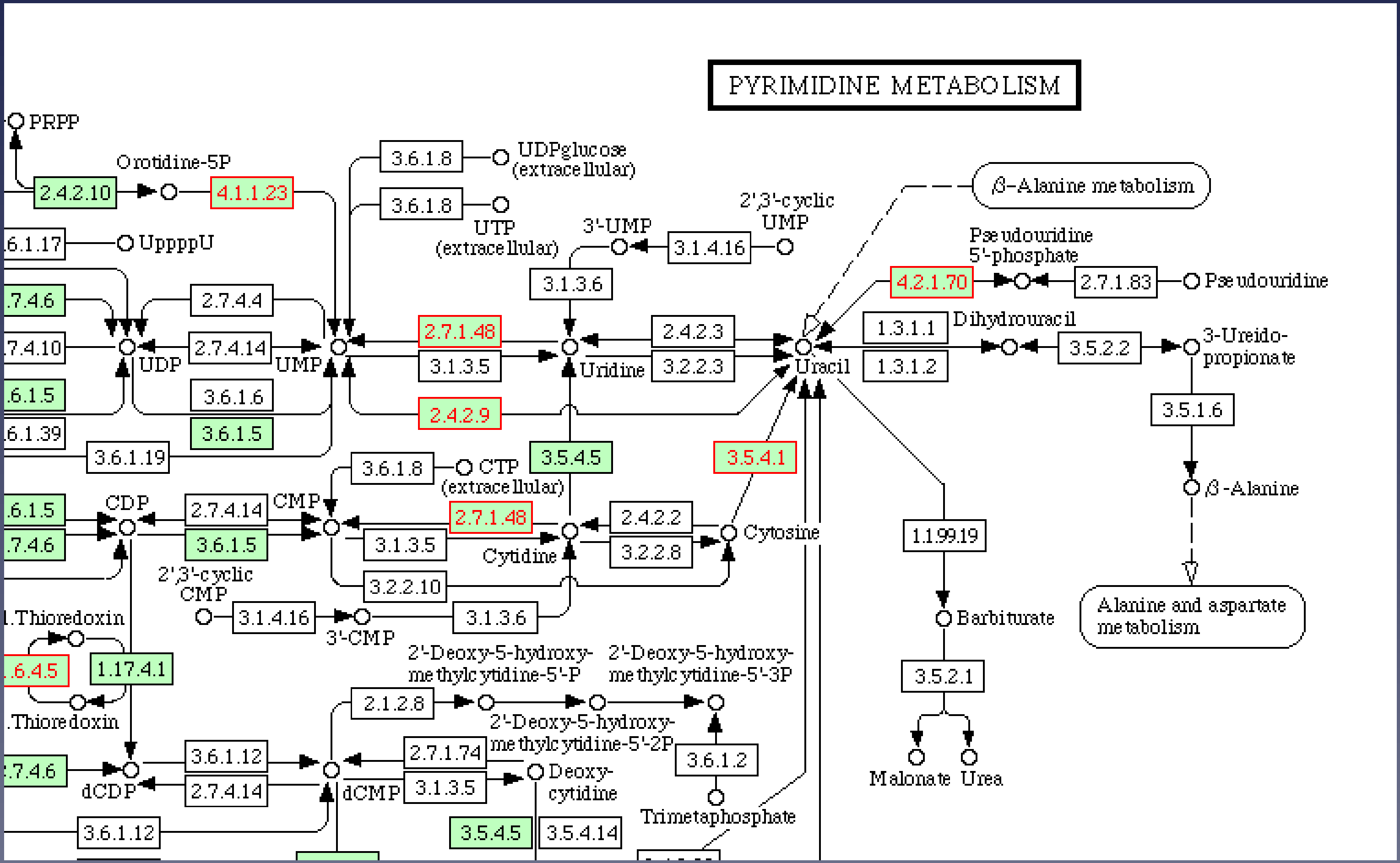
Eukaryotic Pol III

C2	C3	C4	C5	C11
C1	C19	C25	C31	C34

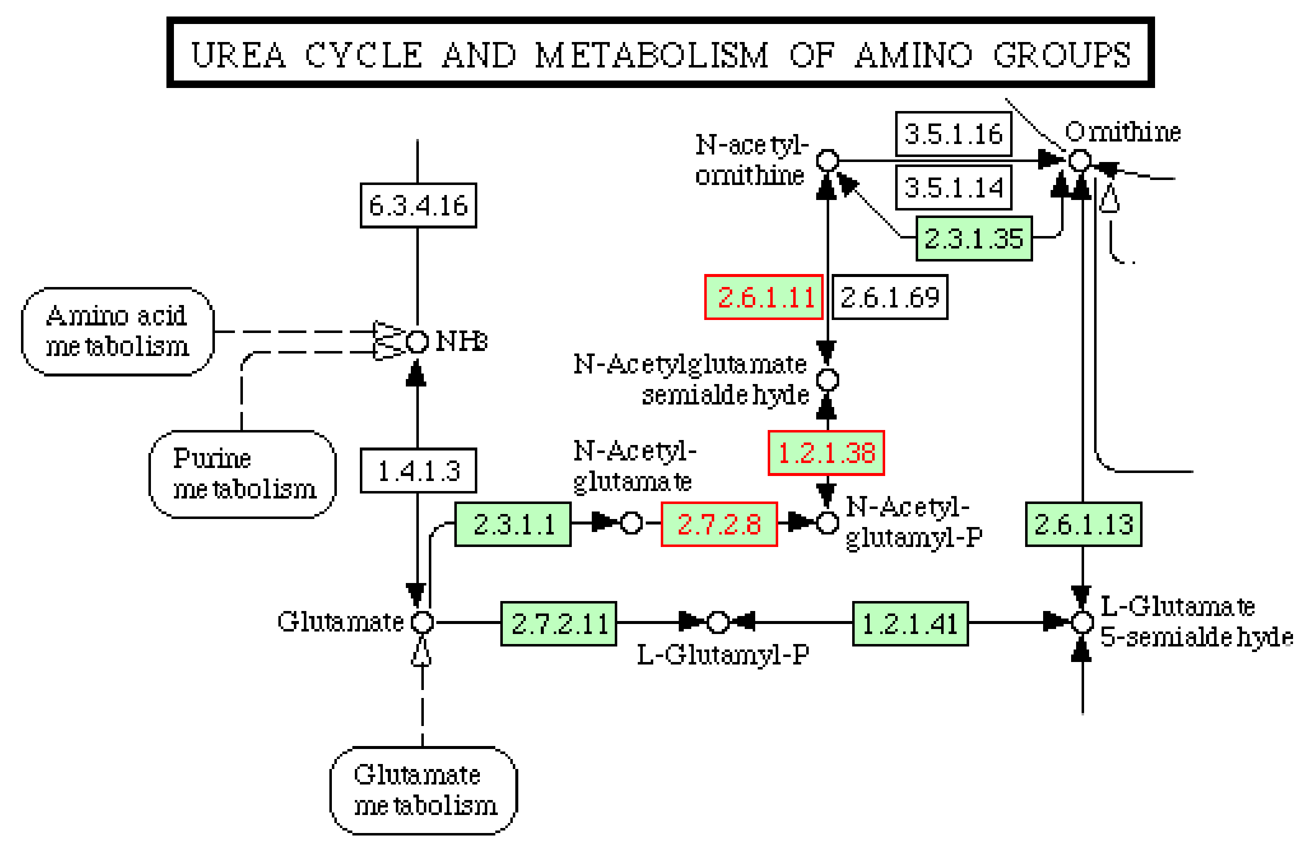
Eukaryotic Pol I

A2	A12	A14	A34	A43	A49
A1					

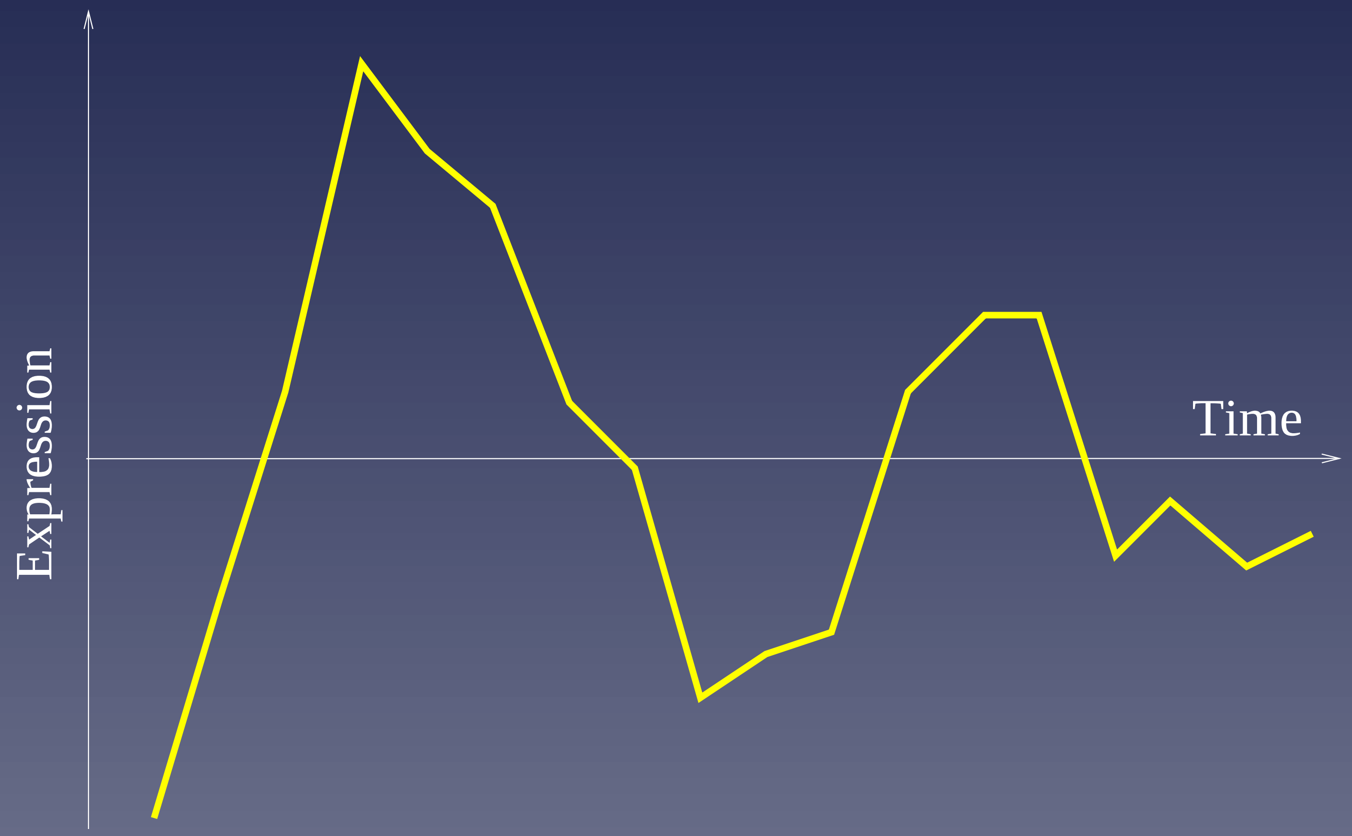
Related genes



Related genes



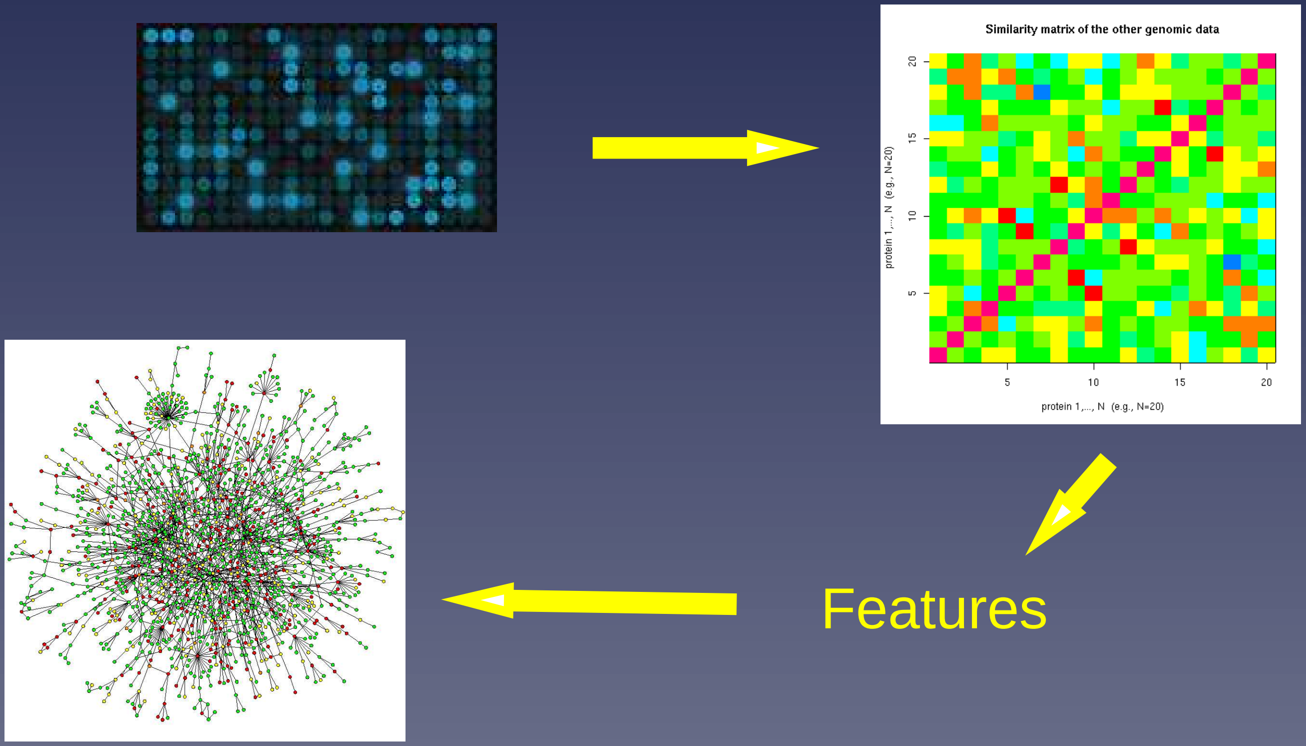
Second pattern



Part 4

Learning from several heterogeneous data

Summary of the process



Features

The “kernel trick”

- The matrix of similarity is $K_{i,j} = x_i^\top x_j$

The “kernel trick”

- The matrix of similarity is $K_{i,j} = x_i^\top x_j$
- However, more general measures are allowed: they simply must be symetric positive definite

The “kernel trick”

- The matrix of similarity is $K_{i,j} = x_i^\top x_j$
- However, more general measures are allowed: they simply must be symetric positive definite
- This enables nonlinear features, as well as features from other types of data, as soon as a symetric p.d. function $K(x, y)$ is defined

Kernels

Several kernels have been developed recently:

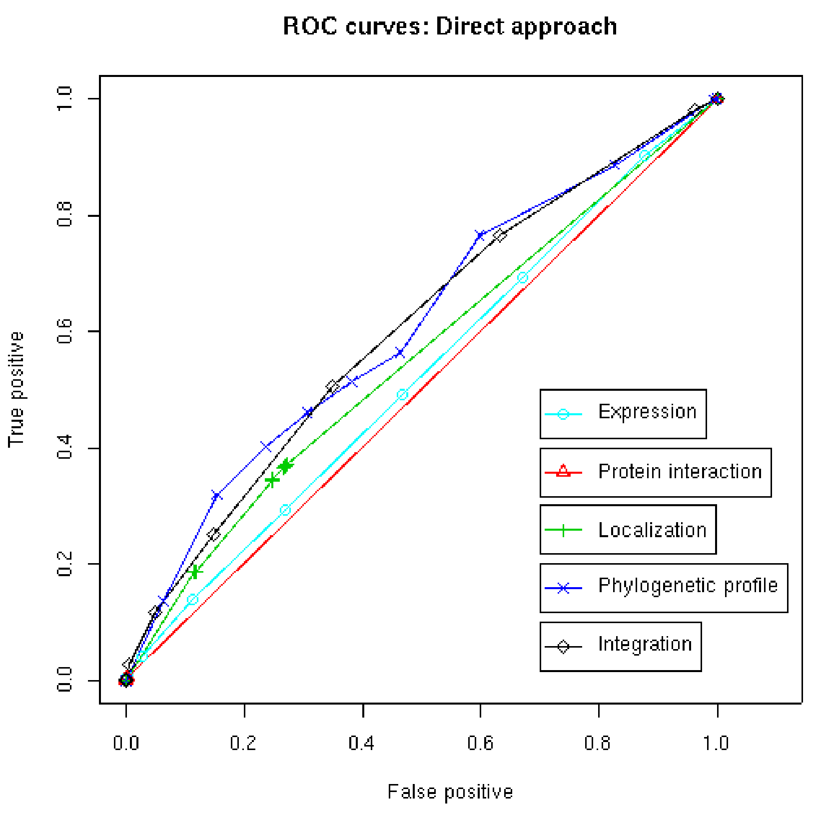
- for phylogenetic profiles (JPV. 2004)
- for gene sequences (Leslie et al. 2003, Saigo et al. 2004, ...)
- for nodes in a network (Kondor et al. 2000)

Learning from heterogeneous data

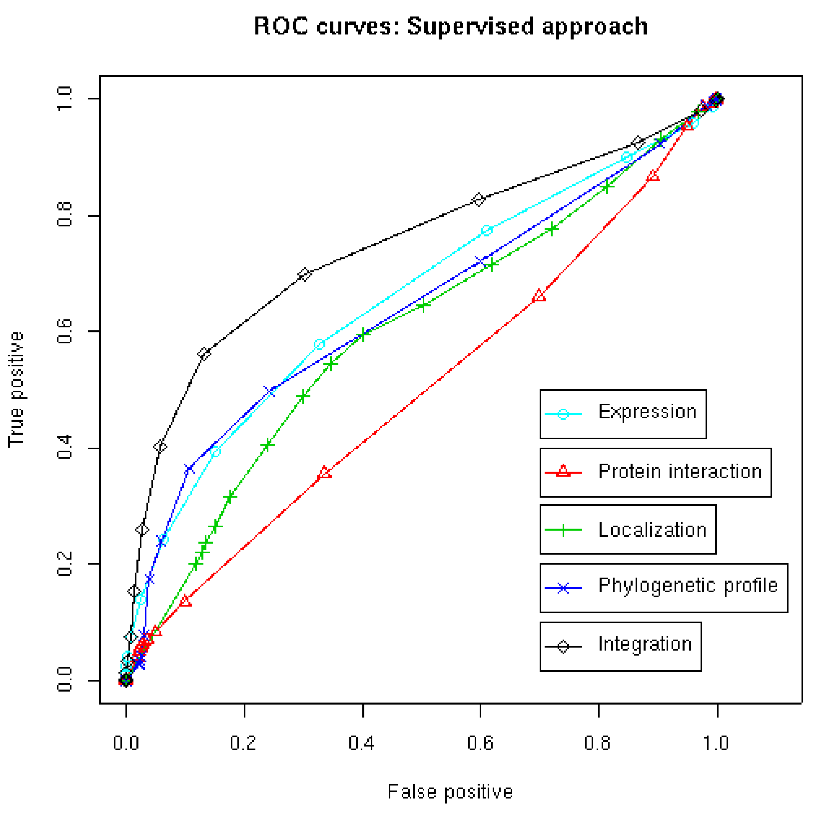
- Suppose several data are available about the genes, e.g., expression, localization, struture, predicted interaction etc...
- Each data can be represented by a positive definite similarity matrix $K_1, \ldots, K_p$ called kernels
- Kernel can be combined by various operations, e.g., addition:

$$K = \sum_{i=1}^{p} K_i$$

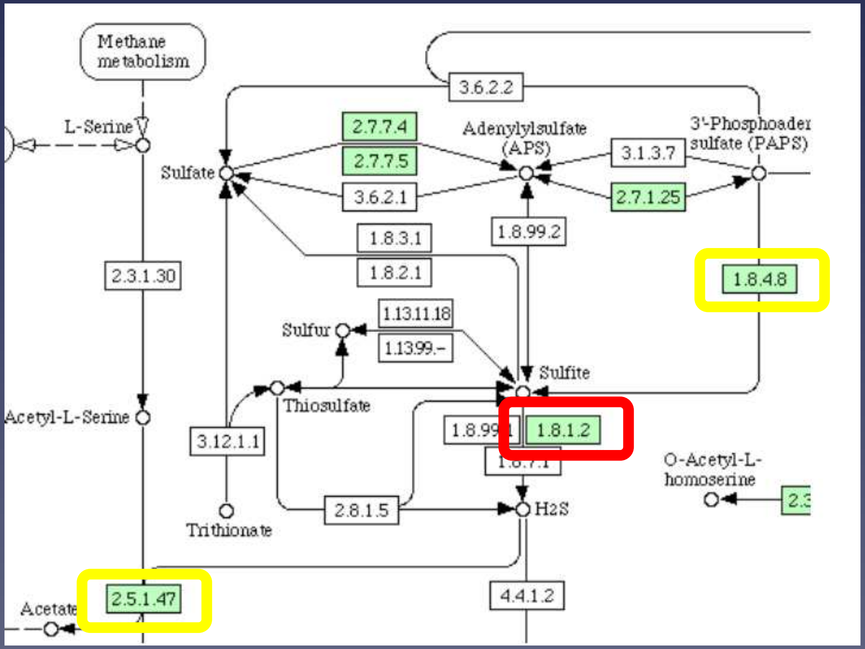
Learning from heterogeneous data (unsupervised)



Learning from heterogeneous data (supervised)



Application: missing enzyme prediction



The gene YJR137C was predicted in 09/2003 between $EC: 1.8.4.8$ and $EC: 2.5.1.47$. It was recently annotated as EC:1.8.1.2

Conclusion

Conclusion

- A new approach to feature extractions and supervised network inference, many possible variants and extensions
- Straightforward generalization to any network (e.g., interactome): the same data can be used to infer different networks
- Currently tested on characterization of tumor cells (with Institut Curie) and metabolism of P. falciparum (with Institut Pasteur).