

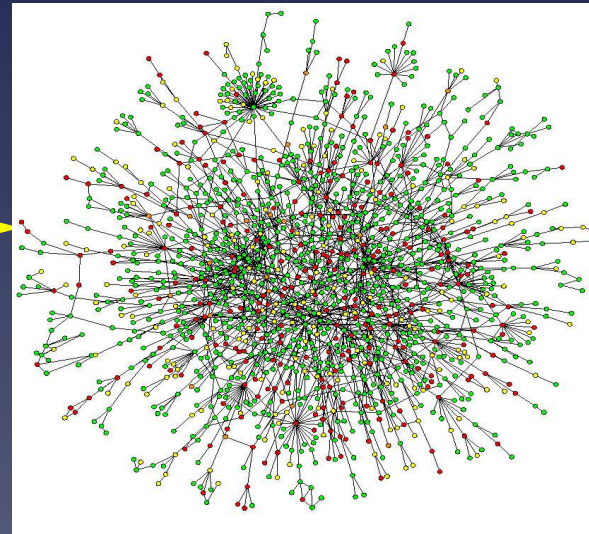
Supervised Gene Network Inference



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

LIX 2005 Workshop: "Bioinformatics: algorithms, structures and statistics", Ecole Polytechnique, Palaiseau, December 14th, 2005

Motivations: systems biology



- Gene expression
- Sequence
- Protein structure
- Protein localization, etc...

- Regulatory network
- Signaling pathways
- Metabolic pathways
- Interaction network, etc...

Outline

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

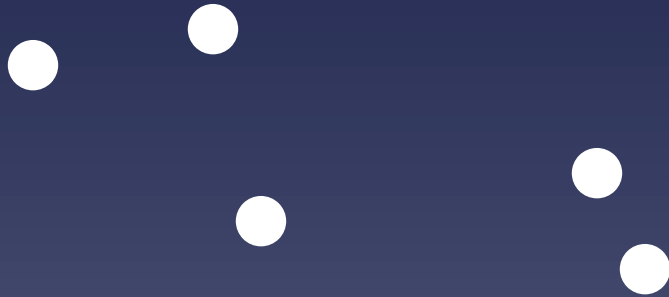
Part 1

A direct approach to network
inference

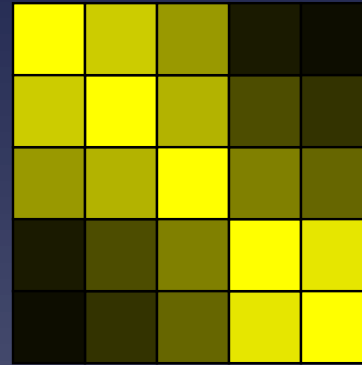
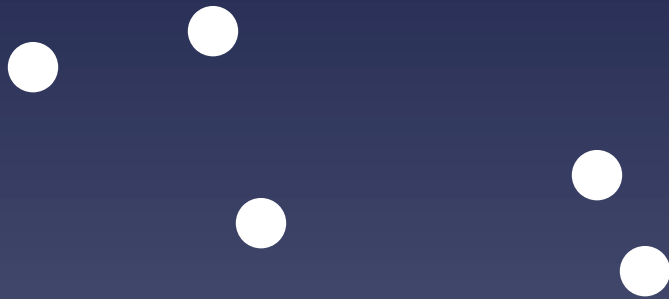
Related approaches

- Fit a statistical models (graphical models, Bayesian nets for regulatory networks...)
- Fit a dynamical system (boolean network, PDE...)
- Nearest neighbors method

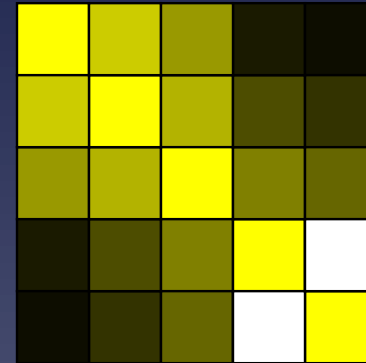
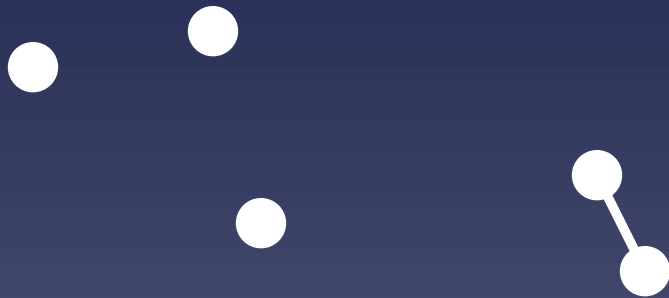
Example: nearest neighbors method



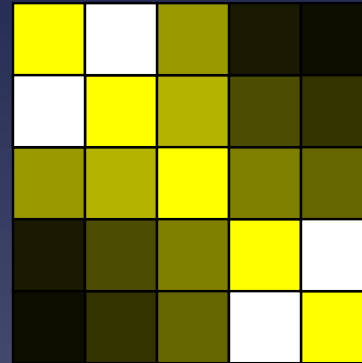
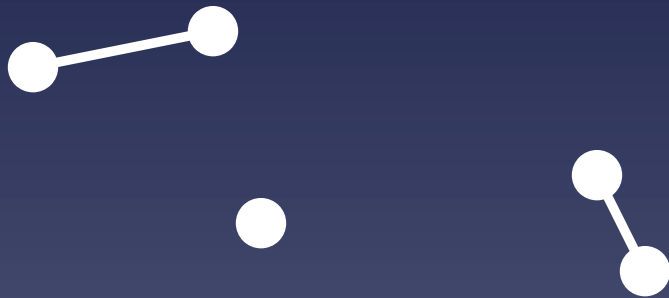
Example: nearest neighbors method



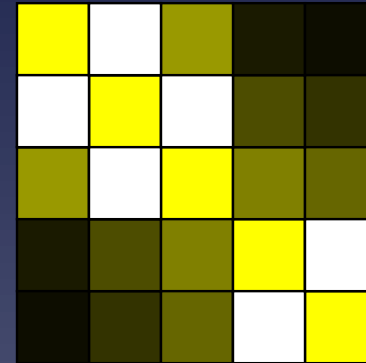
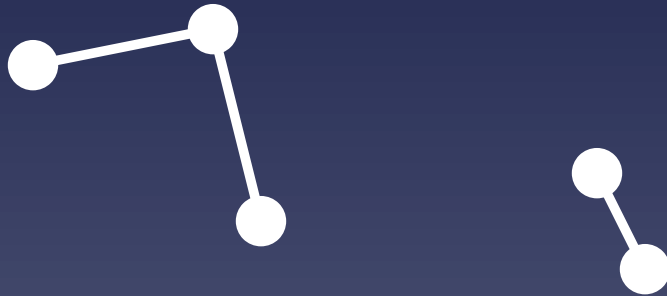
Example: nearest neighbors method



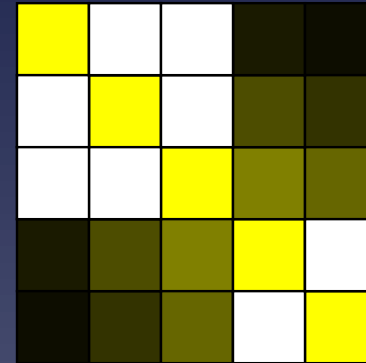
Example: nearest neighbors method



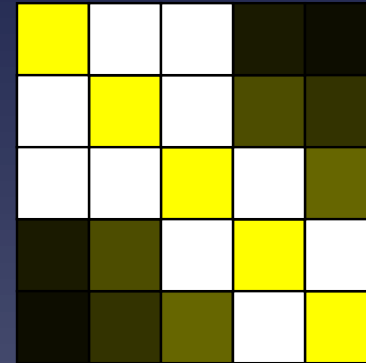
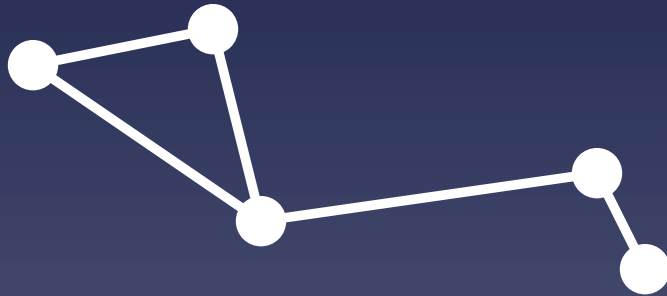
Example: nearest neighbors method



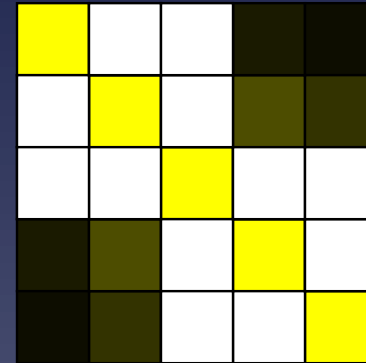
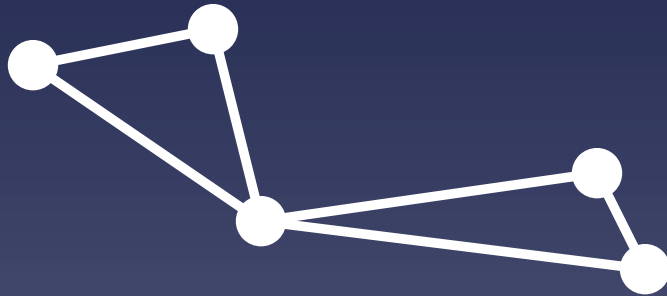
Example: nearest neighbors method



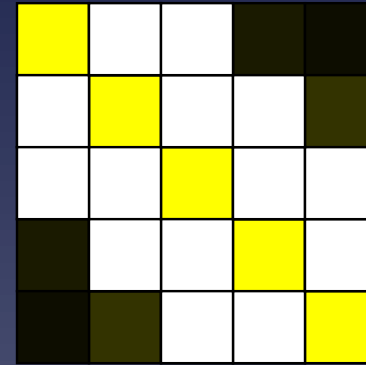
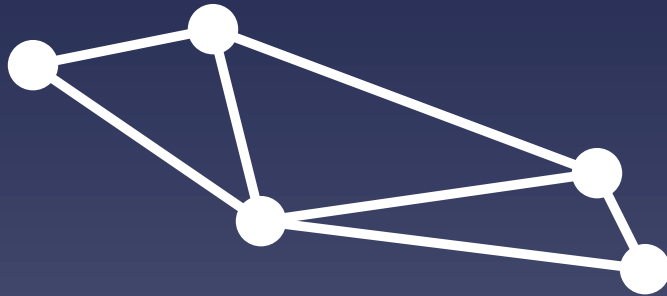
Example: nearest neighbors method



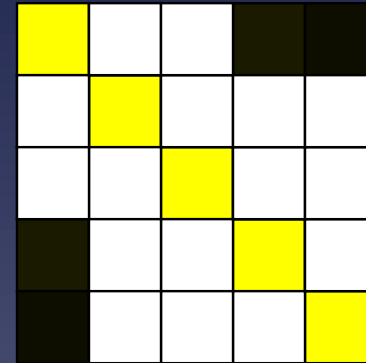
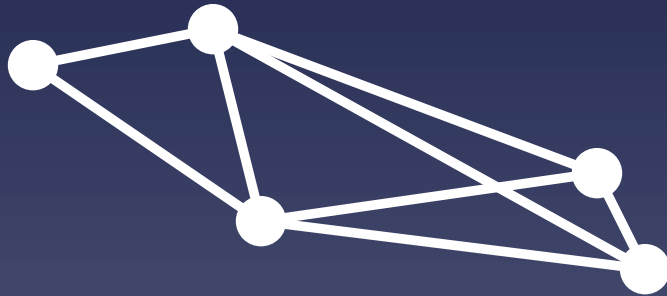
Example: nearest neighbors method



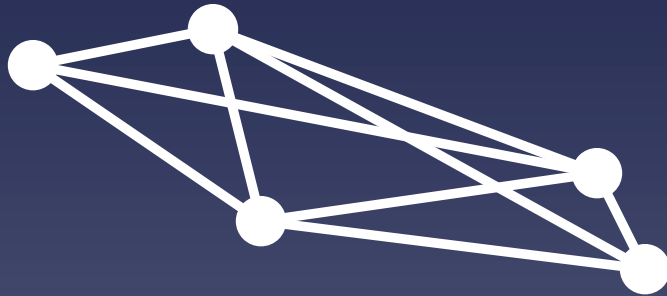
Example: nearest neighbors method



Example: nearest neighbors method

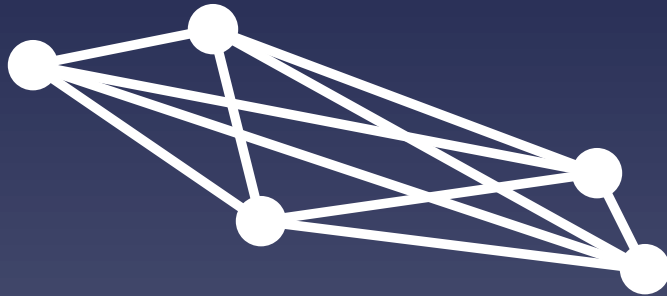


Example: nearest neighbors method



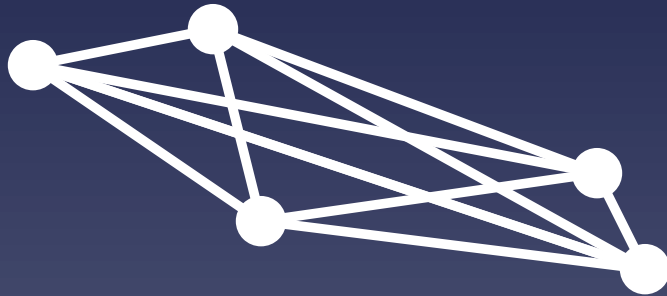
Yellow	White	White	White	Black
White	Yellow	White	White	White
White	White	Yellow	White	White
White	White	White	Yellow	White
Black	White	White	White	Yellow

Example: nearest neighbors method



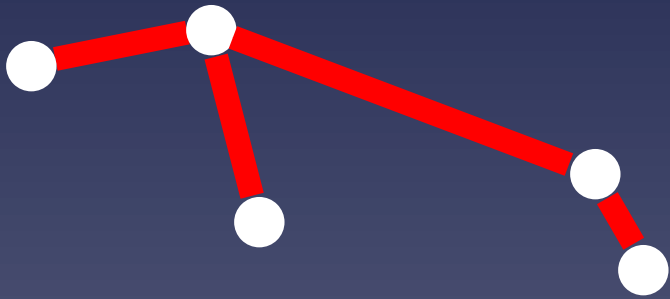
■				
	■			
		■		
			■	
				■

Example: nearest neighbors method

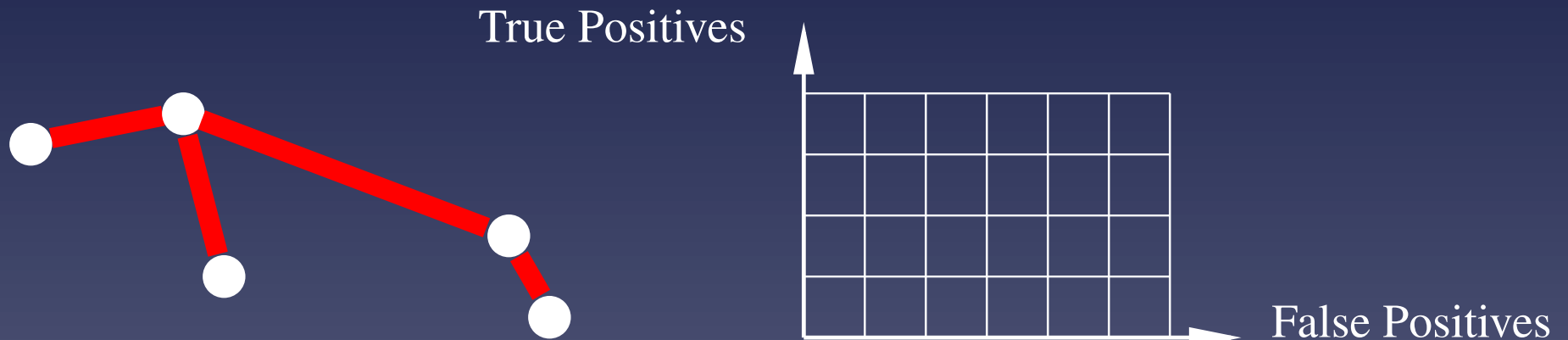


■				
	■			
		■		
			■	
				■

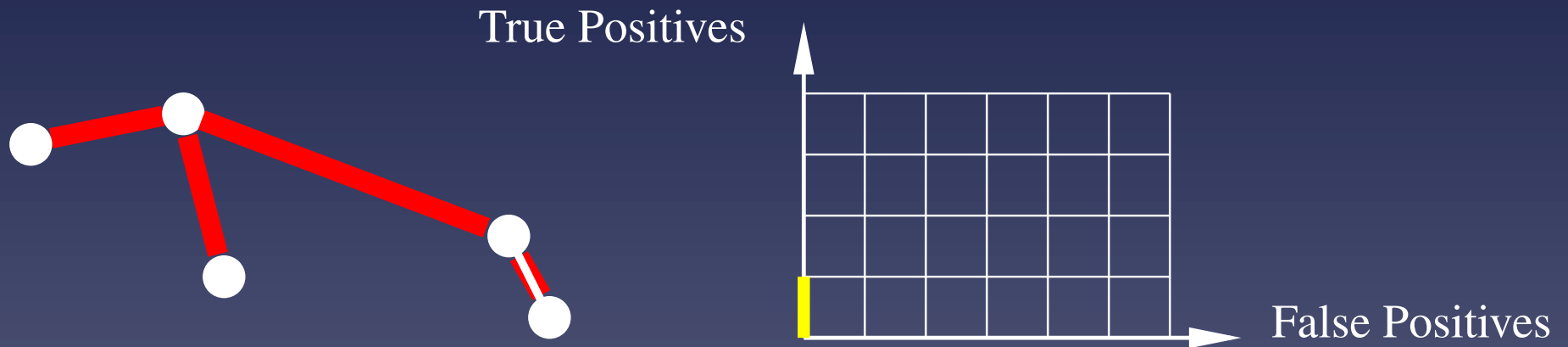
Evaluation of the performance : the ROC curve



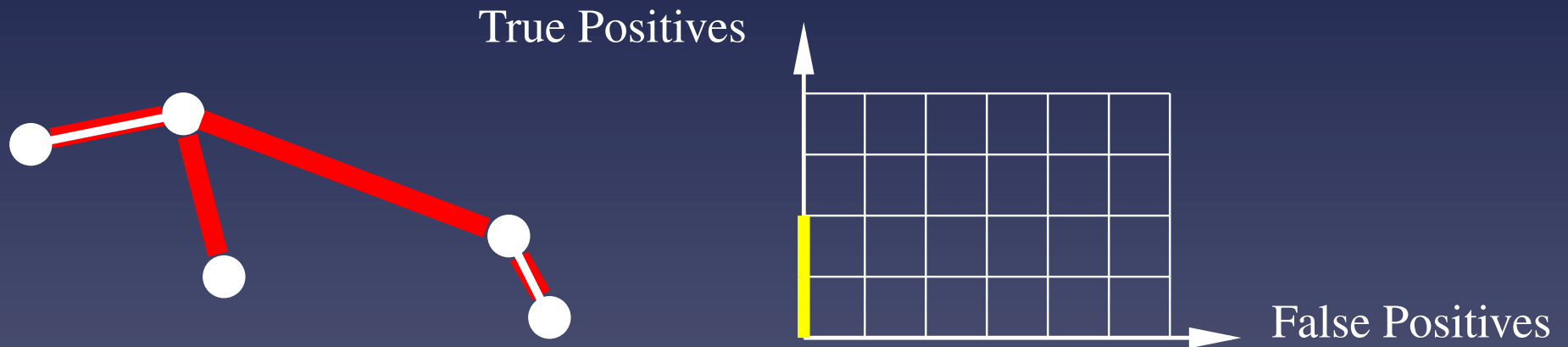
Evaluation of the performance : the ROC curve



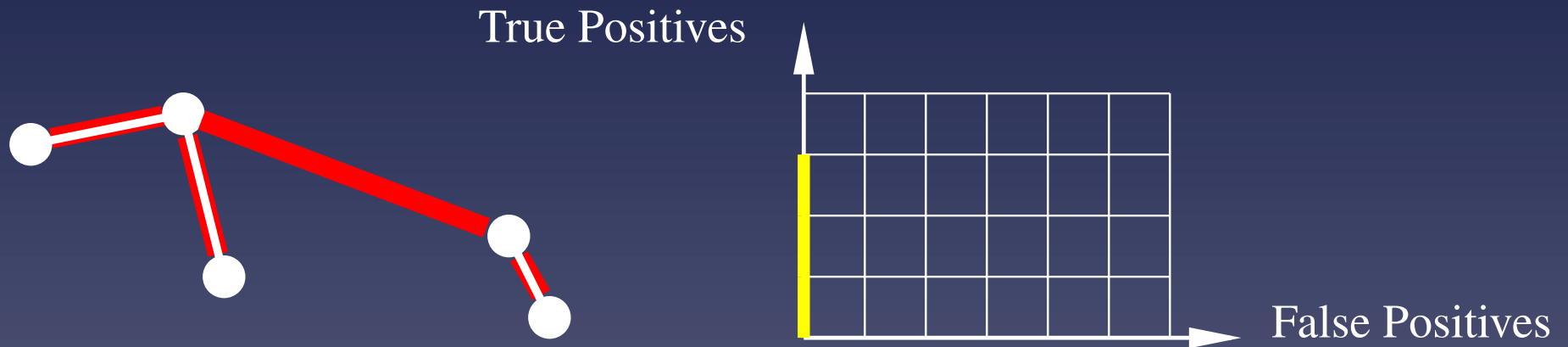
Evaluation of the performance : the ROC curve



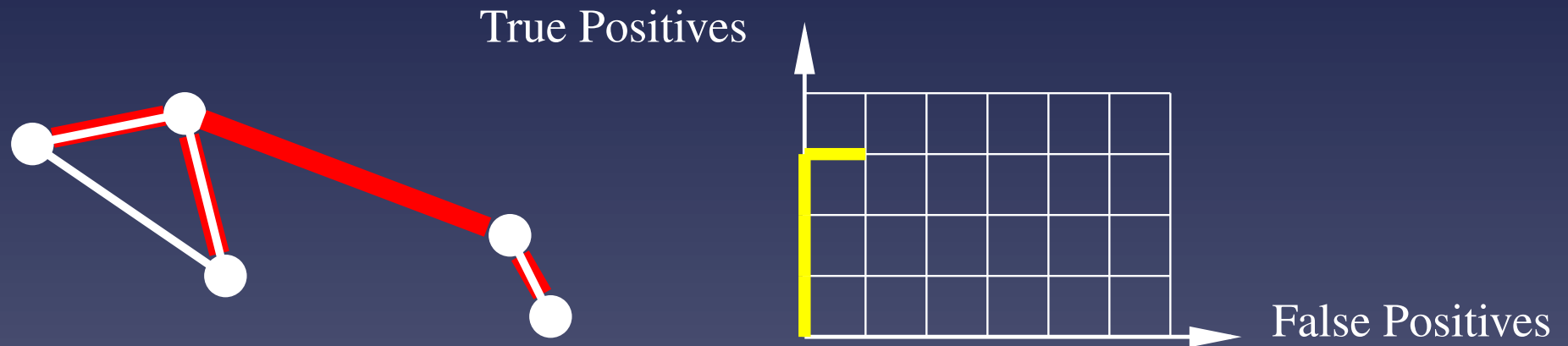
Evaluation of the performance : the ROC curve



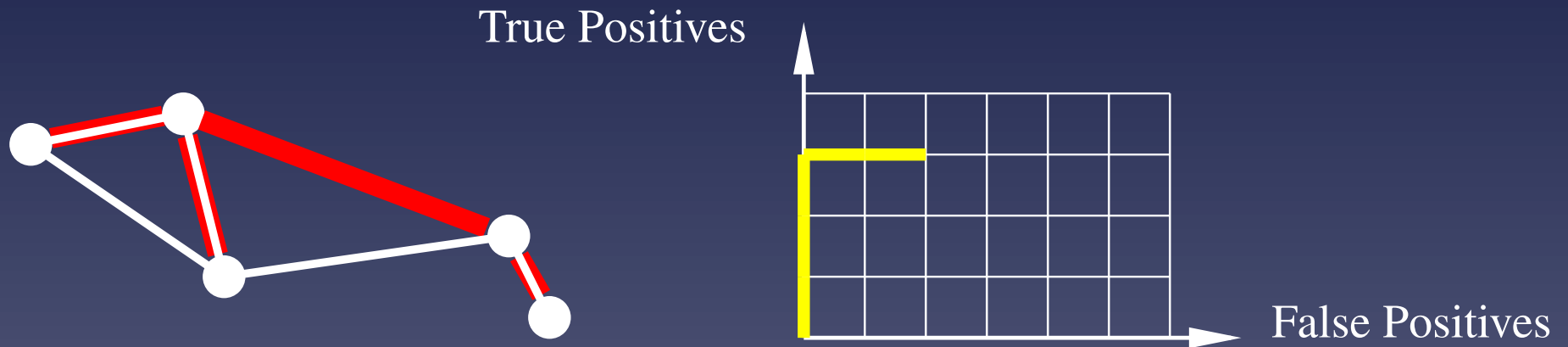
Evaluation of the performance : the ROC curve



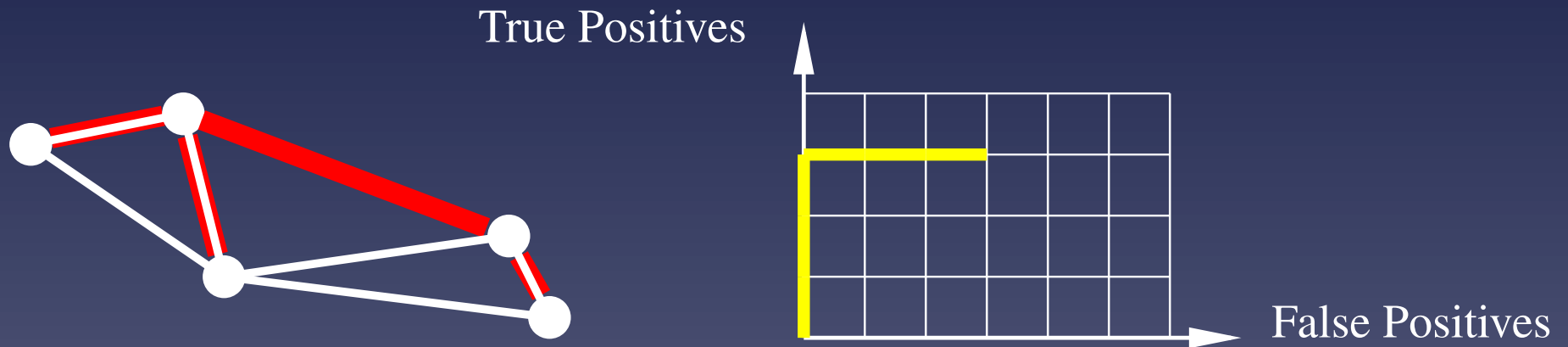
Evaluation of the performance : the ROC curve



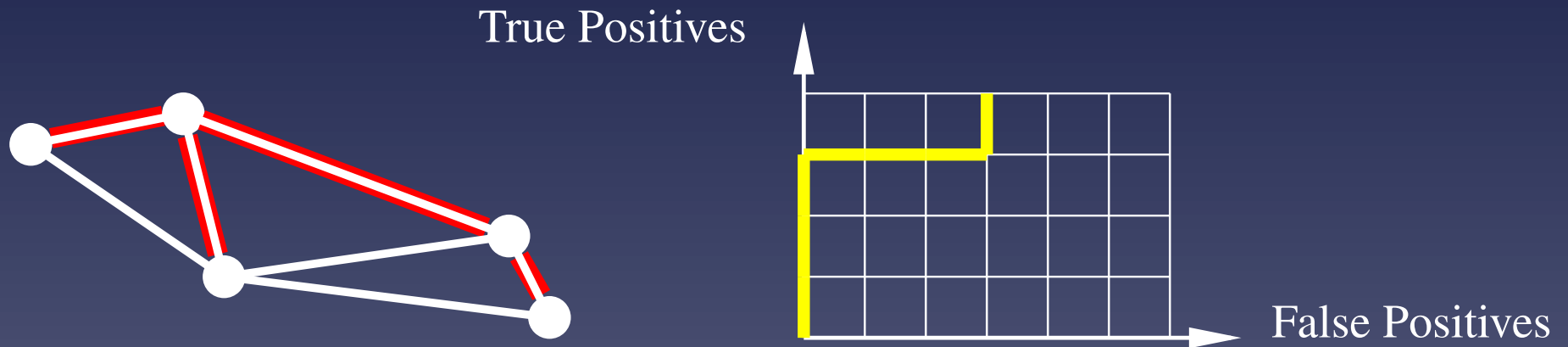
Evaluation of the performance : the ROC curve



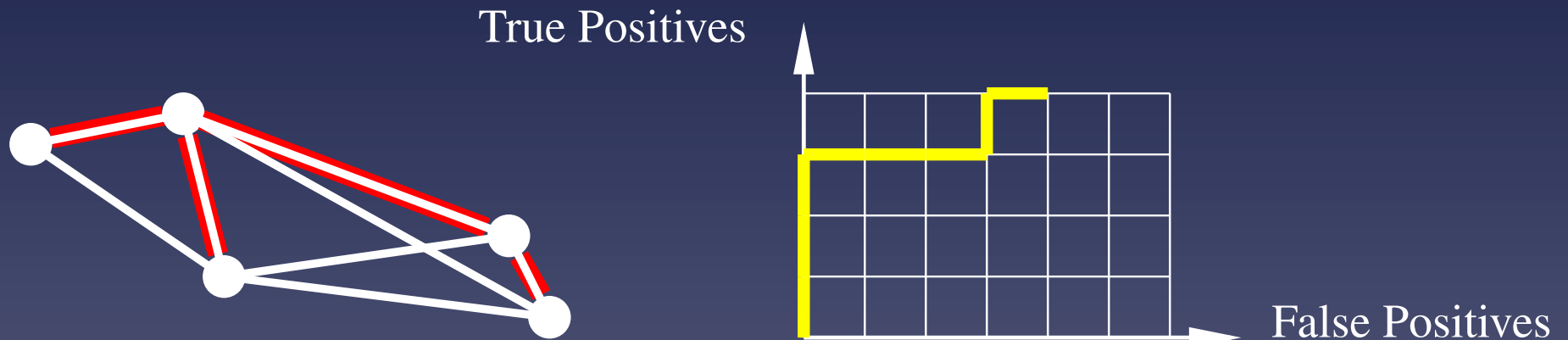
Evaluation of the performance : the ROC curve



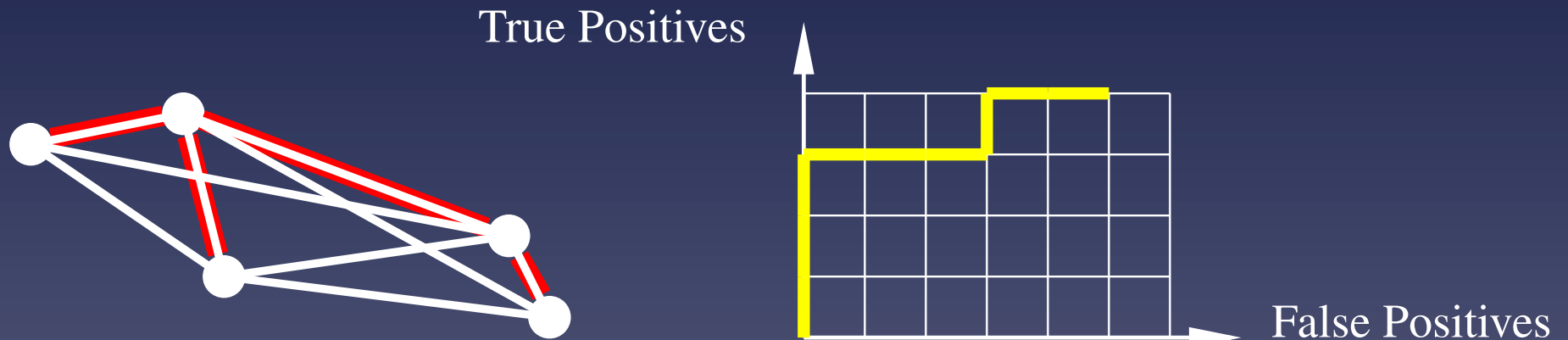
Evaluation of the performance : the ROC curve



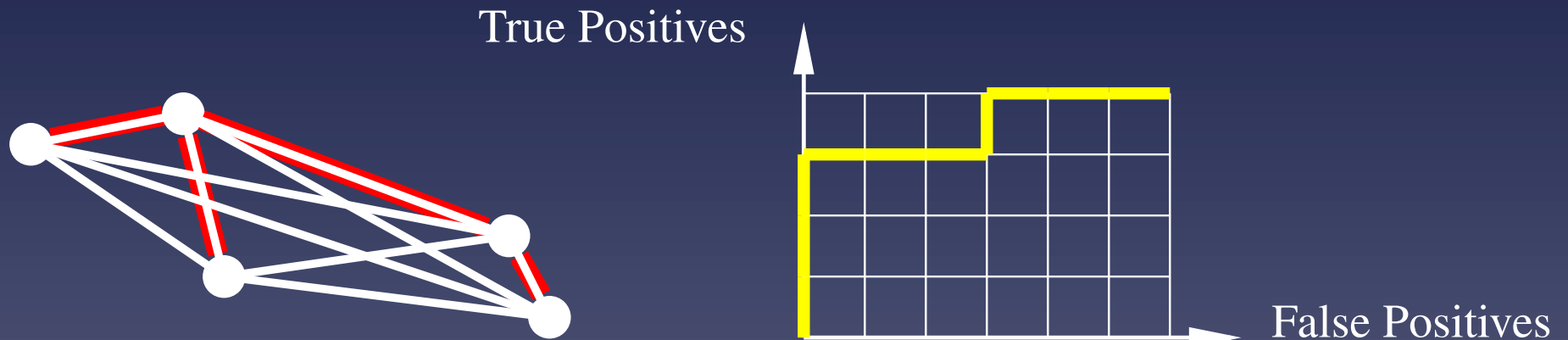
Evaluation of the performance : the ROC curve



Evaluation of the performance : the ROC curve

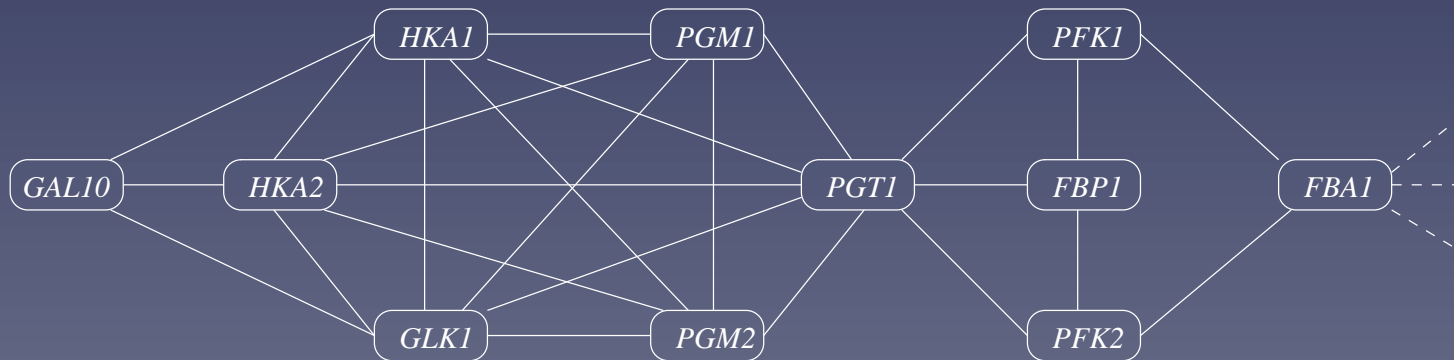
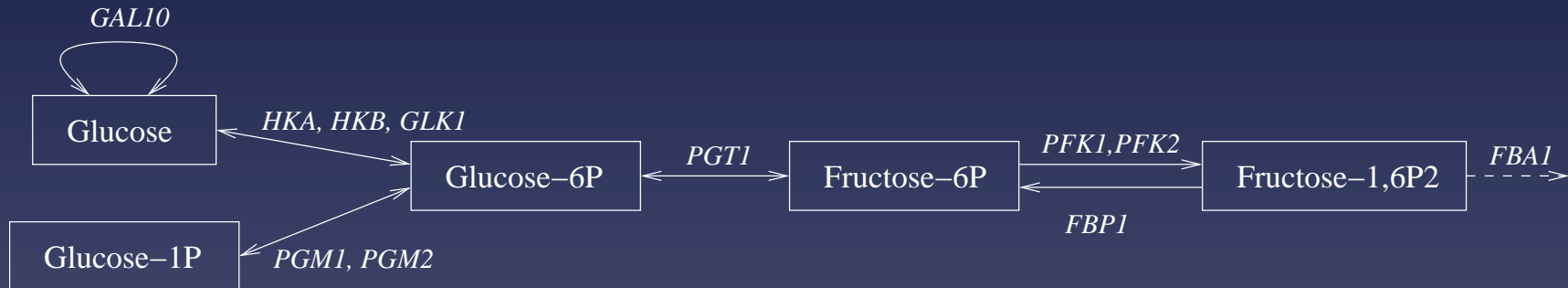


Evaluation of the performance : the ROC curve



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

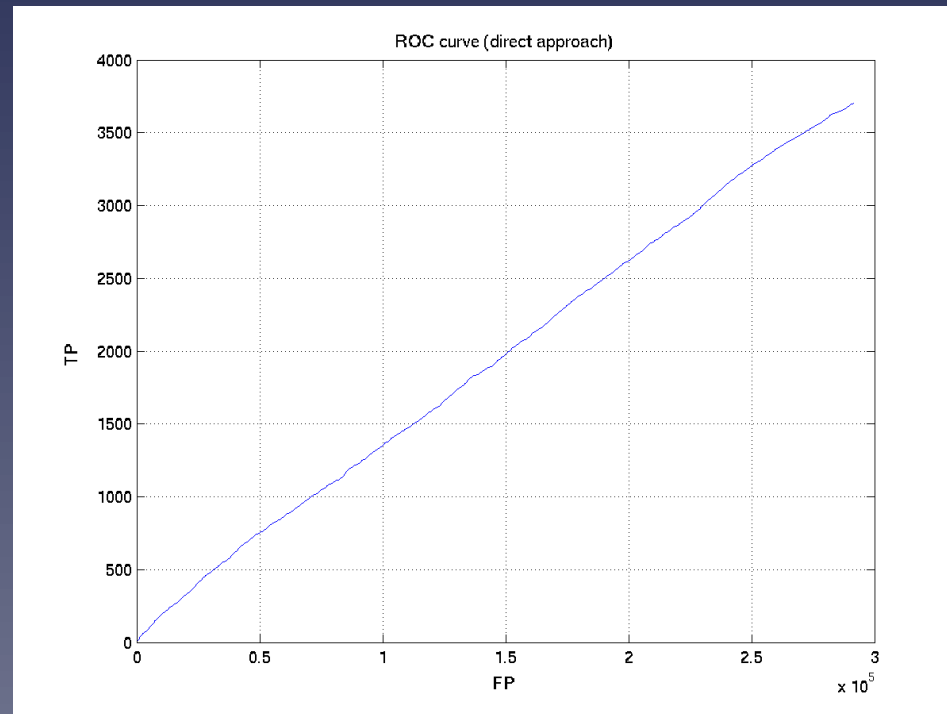
Application: the metabolic gene network



Link two genes when they can **catalyze two successive reactions**

Performance of metabolic network reconstruction

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



What is wrong?

- What **similarity measure** between profiles should be use?

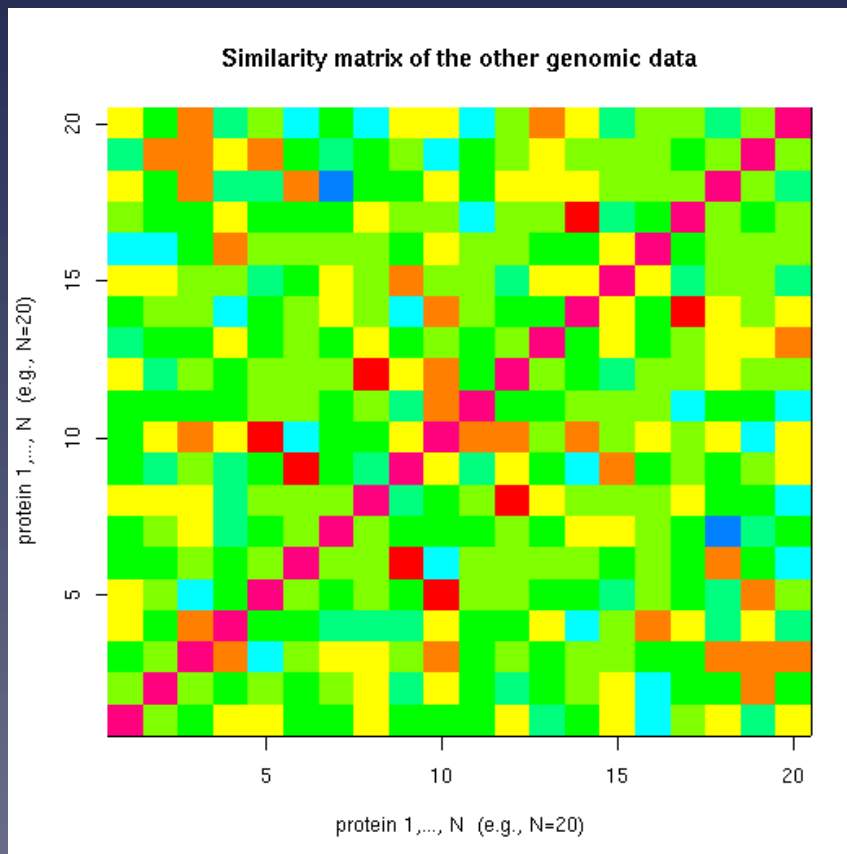
What is wrong?

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

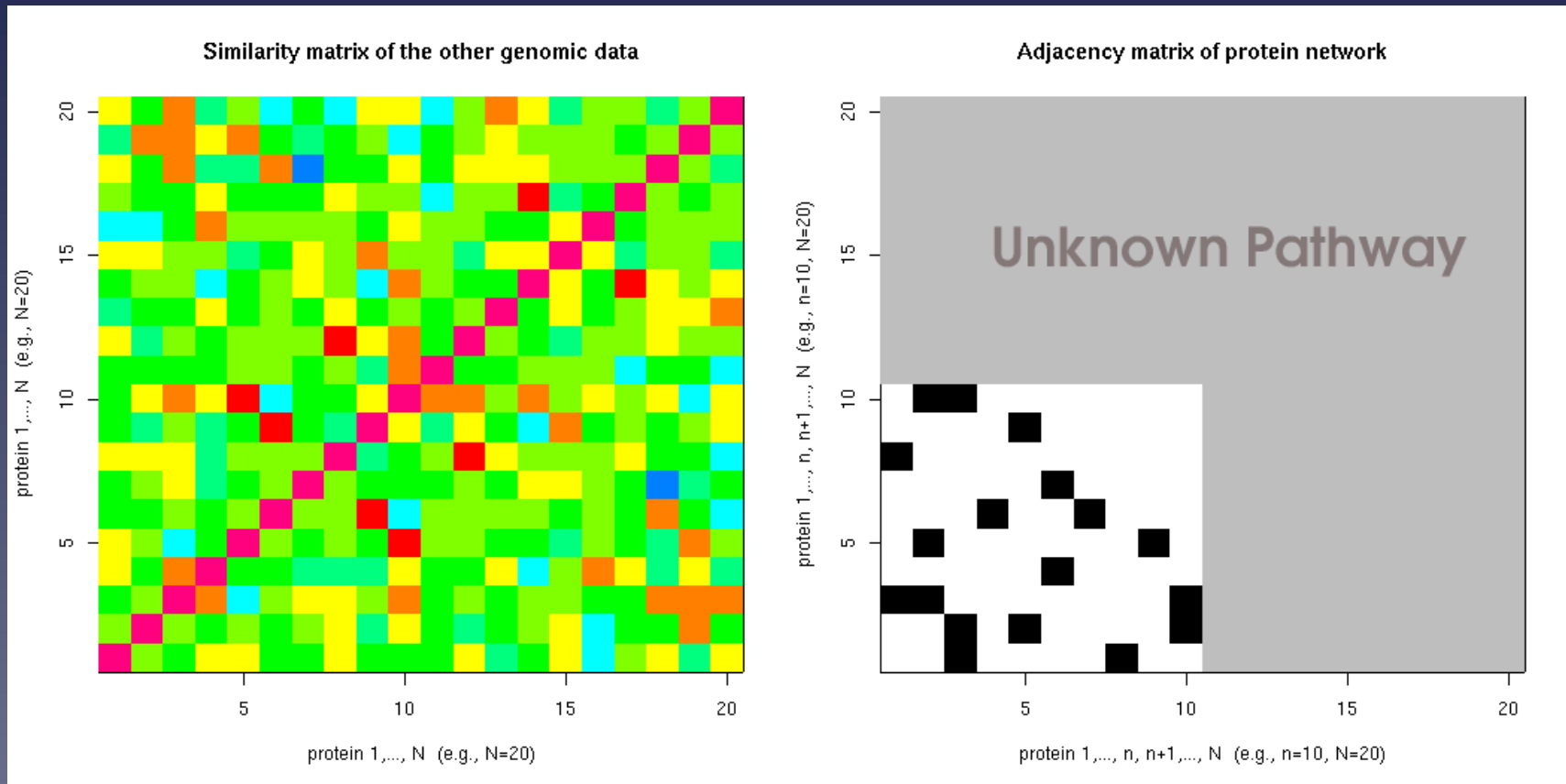
Part 2

Supervised network inference

The supervised gene inference problem

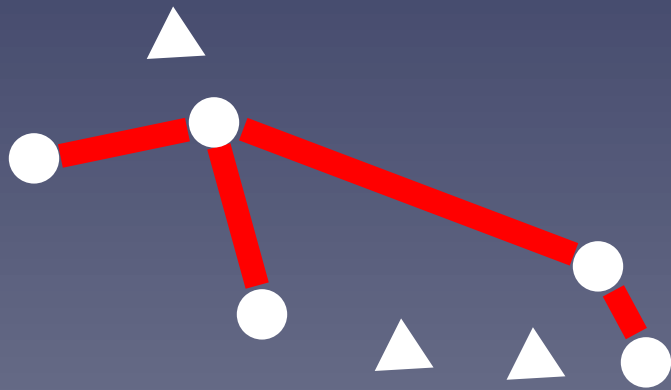


The supervised gene inference problem



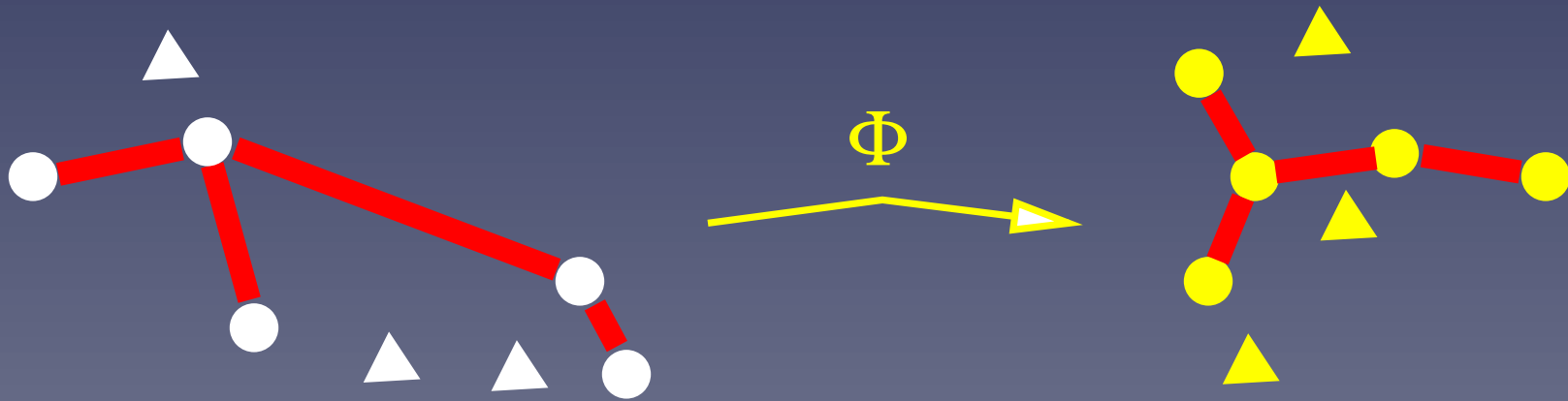
The main idea

Supervised graph inference
through
distance metric learning



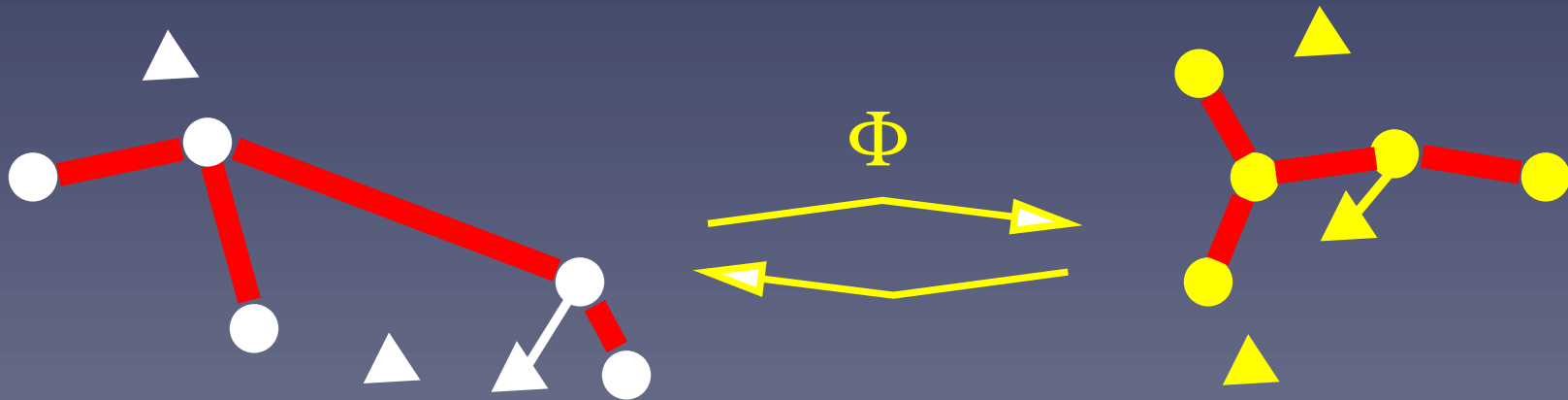
The main idea

Supervised graph inference
through
distance metric learning



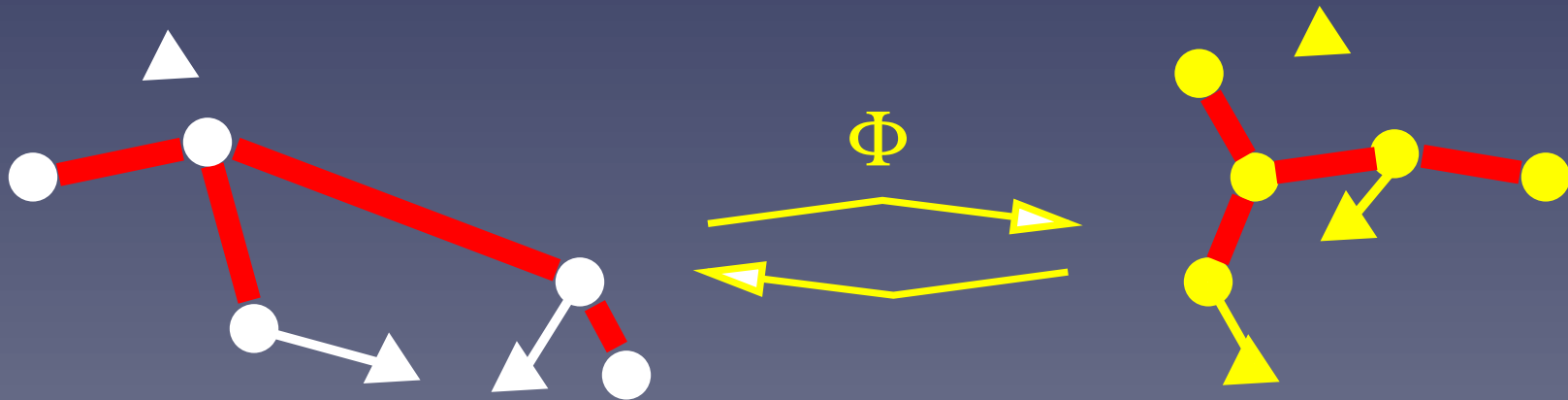
The main idea

Supervised graph inference
through
distance metric learning



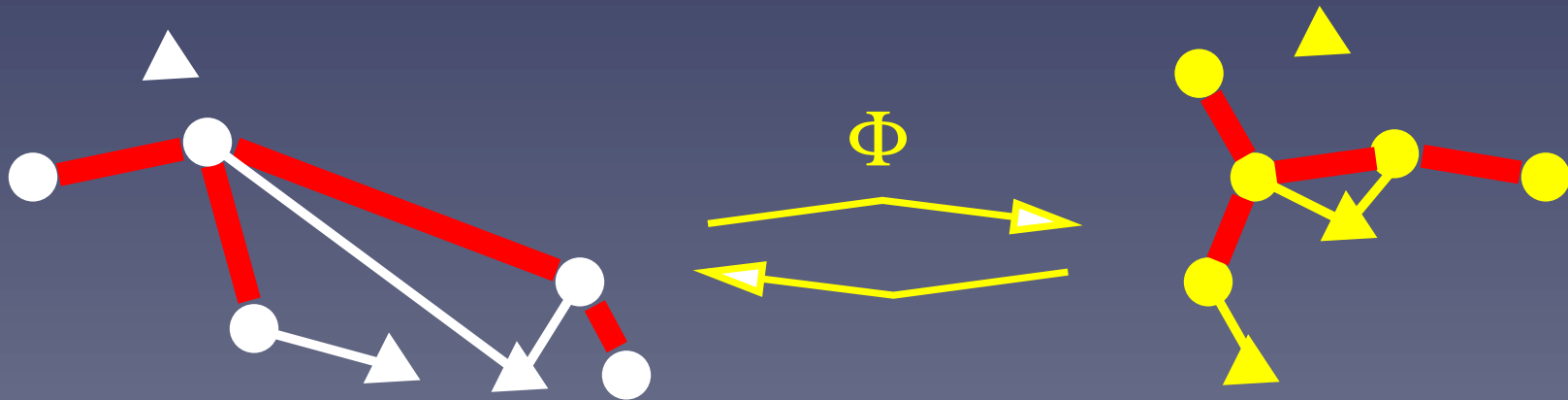
The main idea

Supervised graph inference
through
distance metric learning



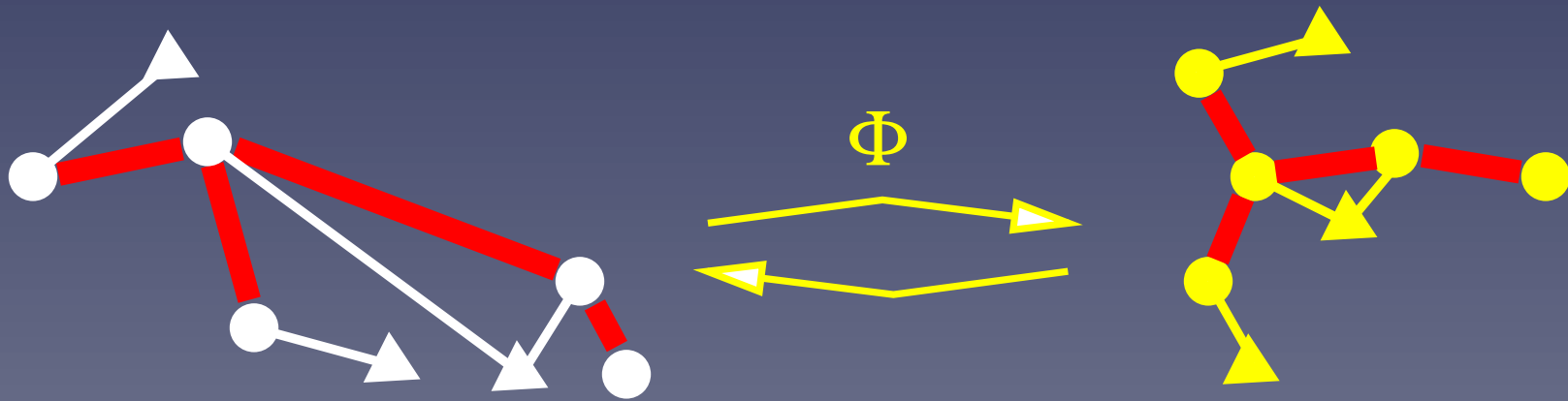
The main idea

Supervised graph inference
through
distance metric learning



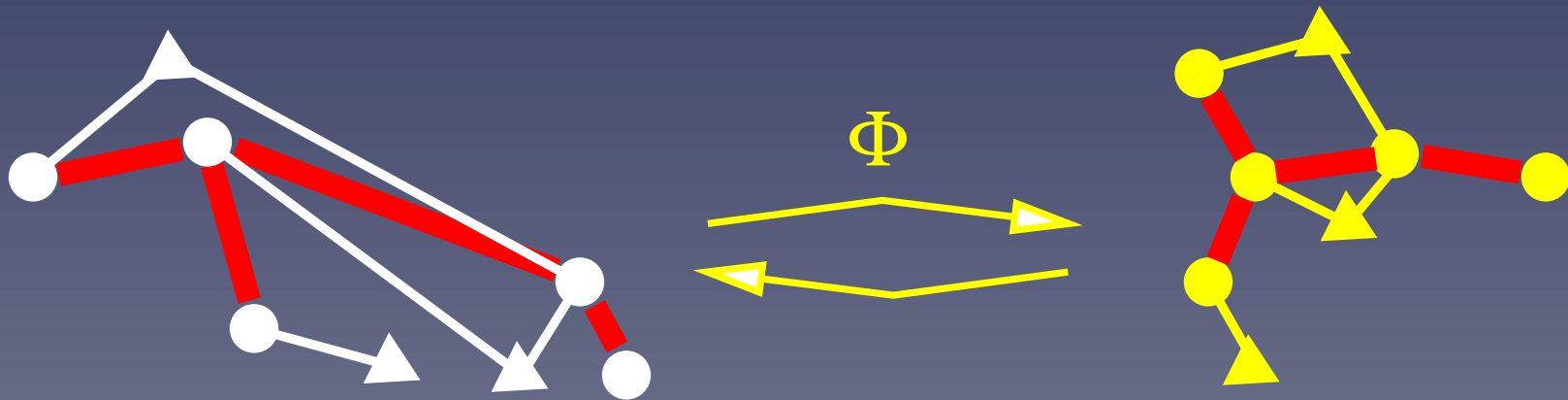
The main idea

Supervised graph inference
through
distance metric learning



The main idea

Supervised graph inference
through
distance metric learning



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:

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

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

“Good” features

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

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

“Good” features

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

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

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

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

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:

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

Extracting successive features

- Successive features to form Φ can be obtained by:

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

- Generalizes Principal Component Analysis (PCA)

Limitations

- How to generalize to **non-linear features**?
- How to process **non-vectorial data** (sequences, phylogenetic profiles, ...)

Overcoming the limitations

- Remember:

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

- In order to allow nonlinear features, we need to replace:
 - ★ $\|w\|^2$ by $\|f\|^2$
 - ★ $w_i \perp w_j$ by $f_i \perp f_j$

Positive definite kernels

Let \mathcal{X} be a set (not necessarily vectors) endowed with a symmetric measure of similarity $k : \mathcal{X}^2 \rightarrow \mathbb{R}$ that satisfies:

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

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

- $k(x, y) = x \cdot y$ for $\mathcal{X} = \mathbb{R}^d$
- $k(x, y) = \exp(-\|x - y\|^2 / (2\sigma^2))$ for $\mathcal{X} = \mathbb{R}^d$

Reproducing kernel Hilbert space

- 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 space is called the **reproducing kernel Hilbert space** (RKHS)

Example: linear RKHS

For $\mathcal{X} = \mathbb{R}^d$ and $k(x, y) = x \cdot y$, we have:

- $f(x) = \sum_{i=1}^n c_i x_i \cdot x = f_w(x)$ with $w = \sum_{i=1}^n c_i x_i$.
- $\|f\|_k^2 = \sum_{i,j=1}^n c_i c_j x_i \cdot x_j = \|w\|^2$
- If $f(x) = w \cdot x$ and $g(x) = v \cdot x$ then:

$$\langle f, g \rangle_k = w \cdot v$$

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:

$$f_i = \arg \min_{f \perp \{f_1, \dots, f_{i-1}\}, \hat{\text{var}}(f)=1} \left\{ \sum_{i \sim j} (f(x_i) - f(x_j))^2 + \lambda \|f\|_k^2 \right\}.$$

- 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

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

Solving the problem (cont.)

- The problem can then be rewritten:

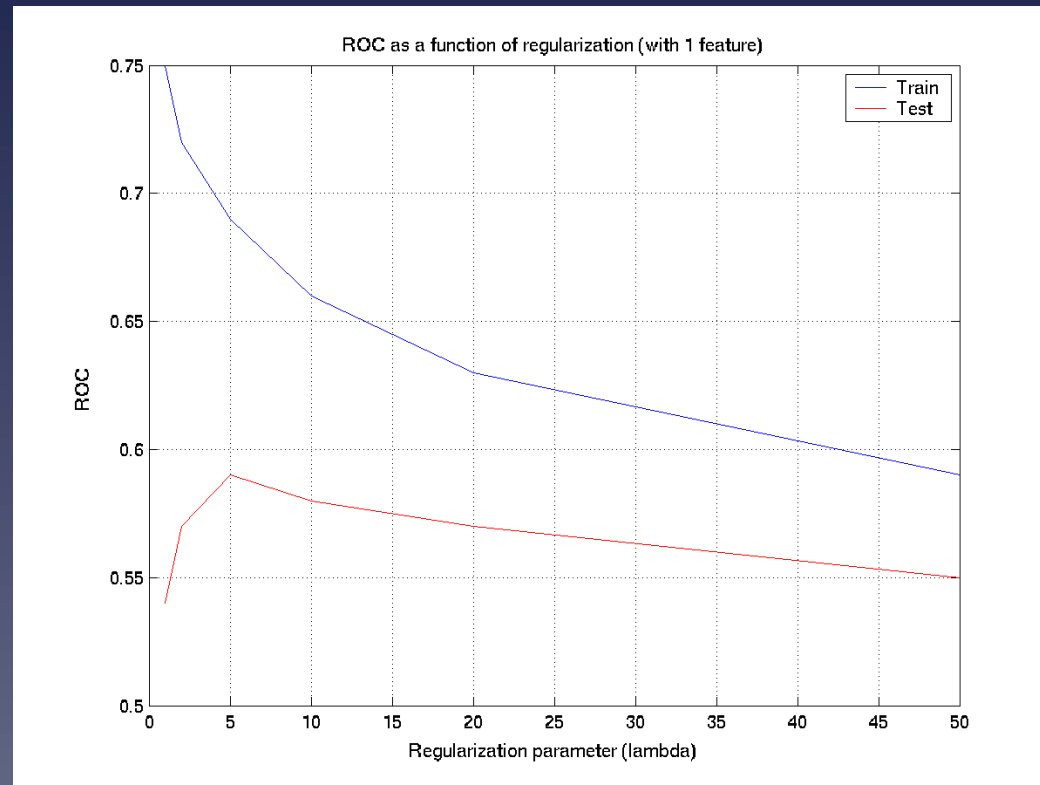
$$\alpha_i = \arg \min_{\alpha \in \mathbb{R}^n, \alpha K_V \alpha_1 = \dots = \alpha K_V \alpha_{i-1} = 0} \left\{ \frac{\alpha^\top K_V L K_V \alpha + \lambda \alpha^\top K_V \alpha}{\alpha^\top K_V^2 \alpha} \right\}$$

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:

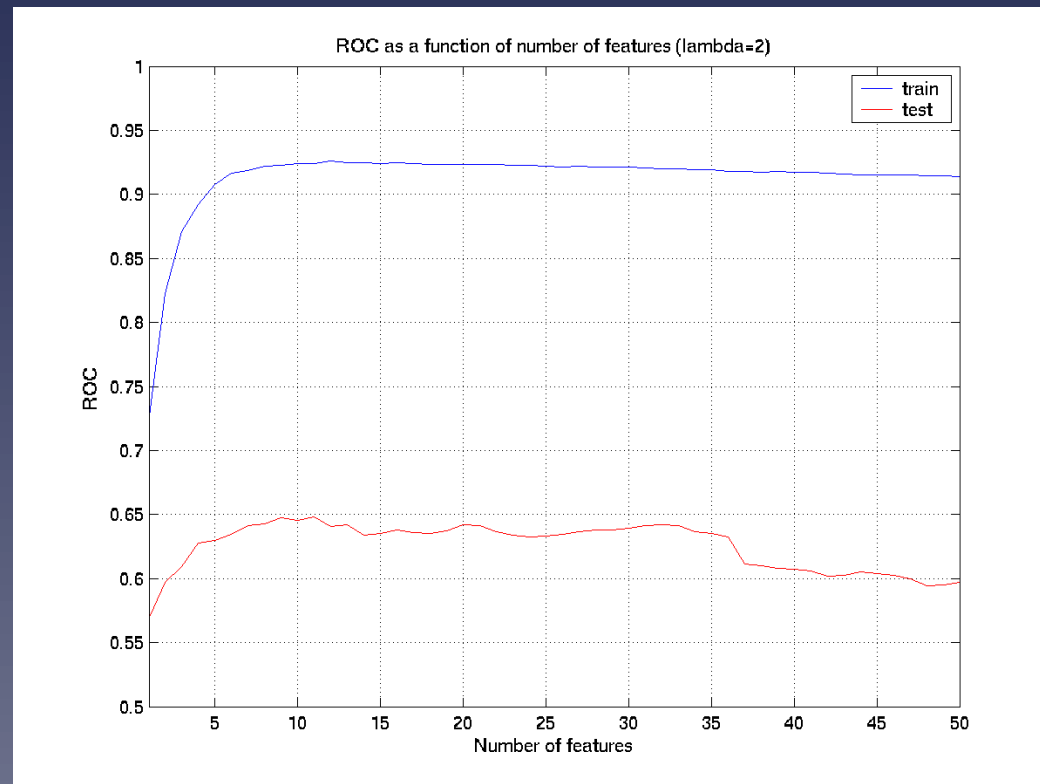
$$(LK_V + \lambda I)\alpha = \mu K_V \alpha.$$

Evaluation of the supervised approach: effect of λ



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

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



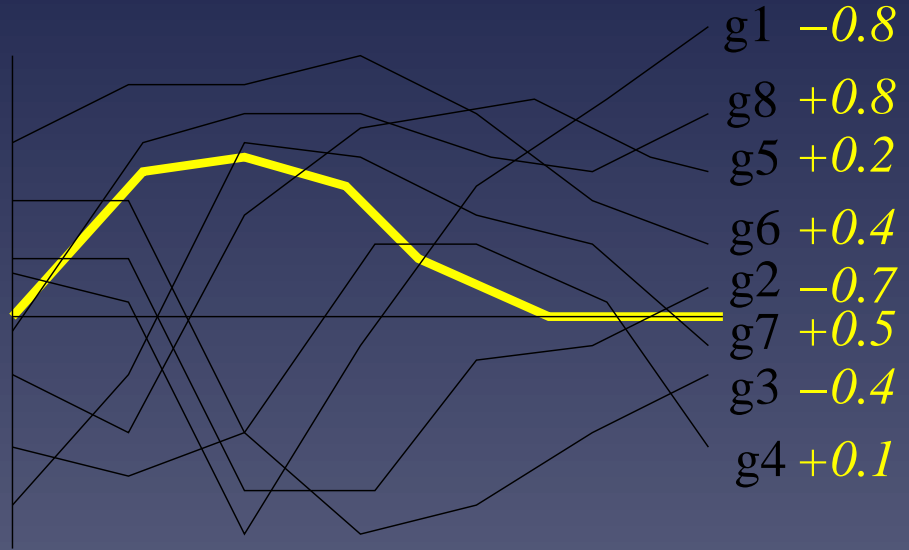
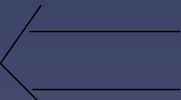
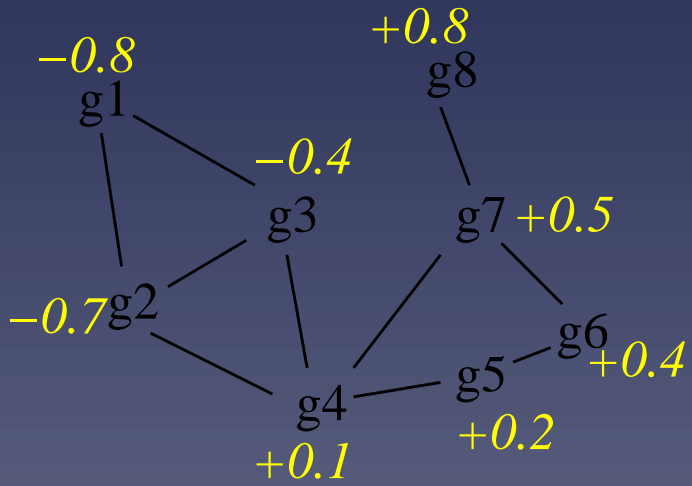
Part 3

Extraction and visualization of pathway activities

The idea

- The previous approach is a way to extract features from gene expression data: $f(x) = w^T x$.
- 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**

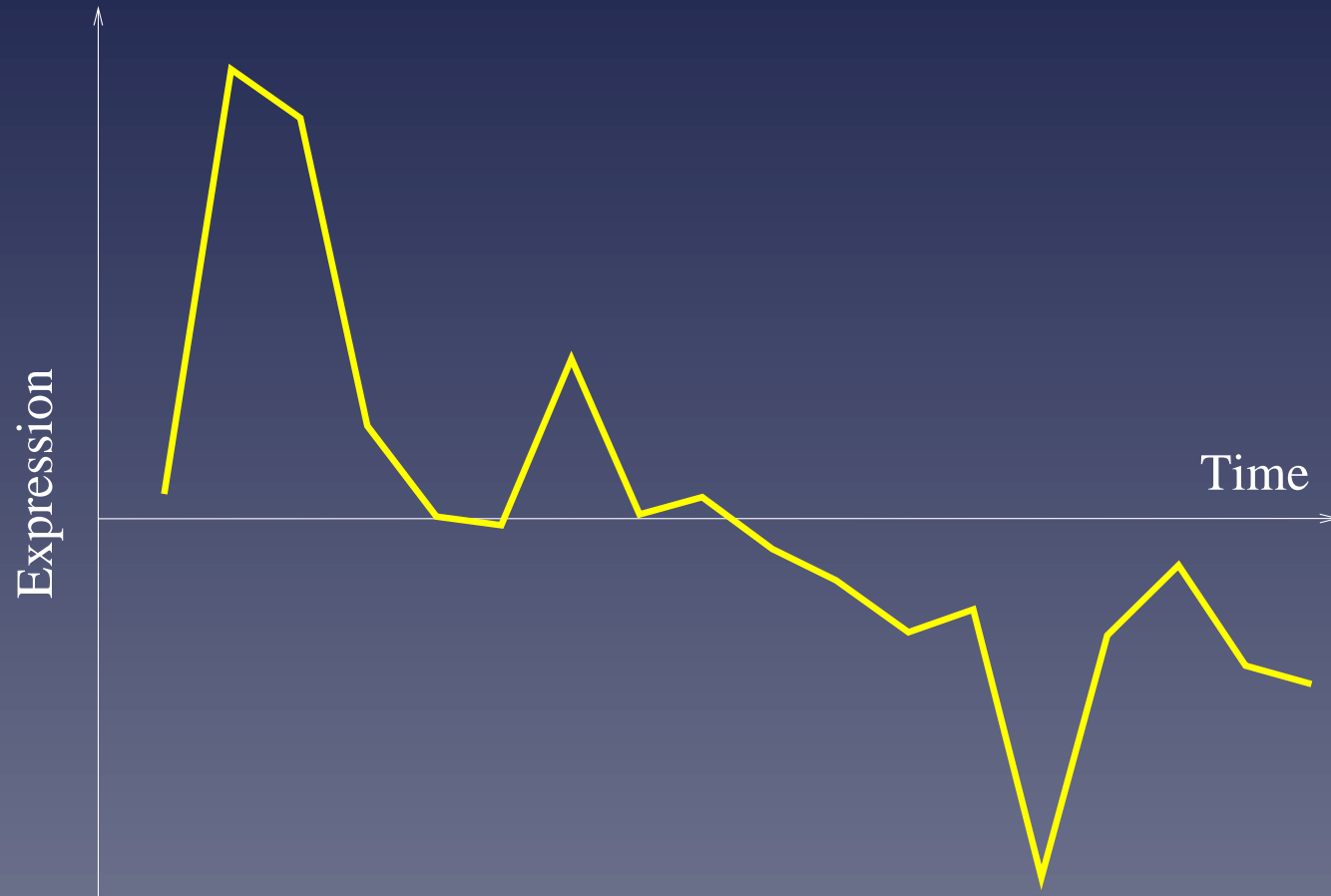
Illustration



Experiment

- **Gene network:** two genes are linked if they 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

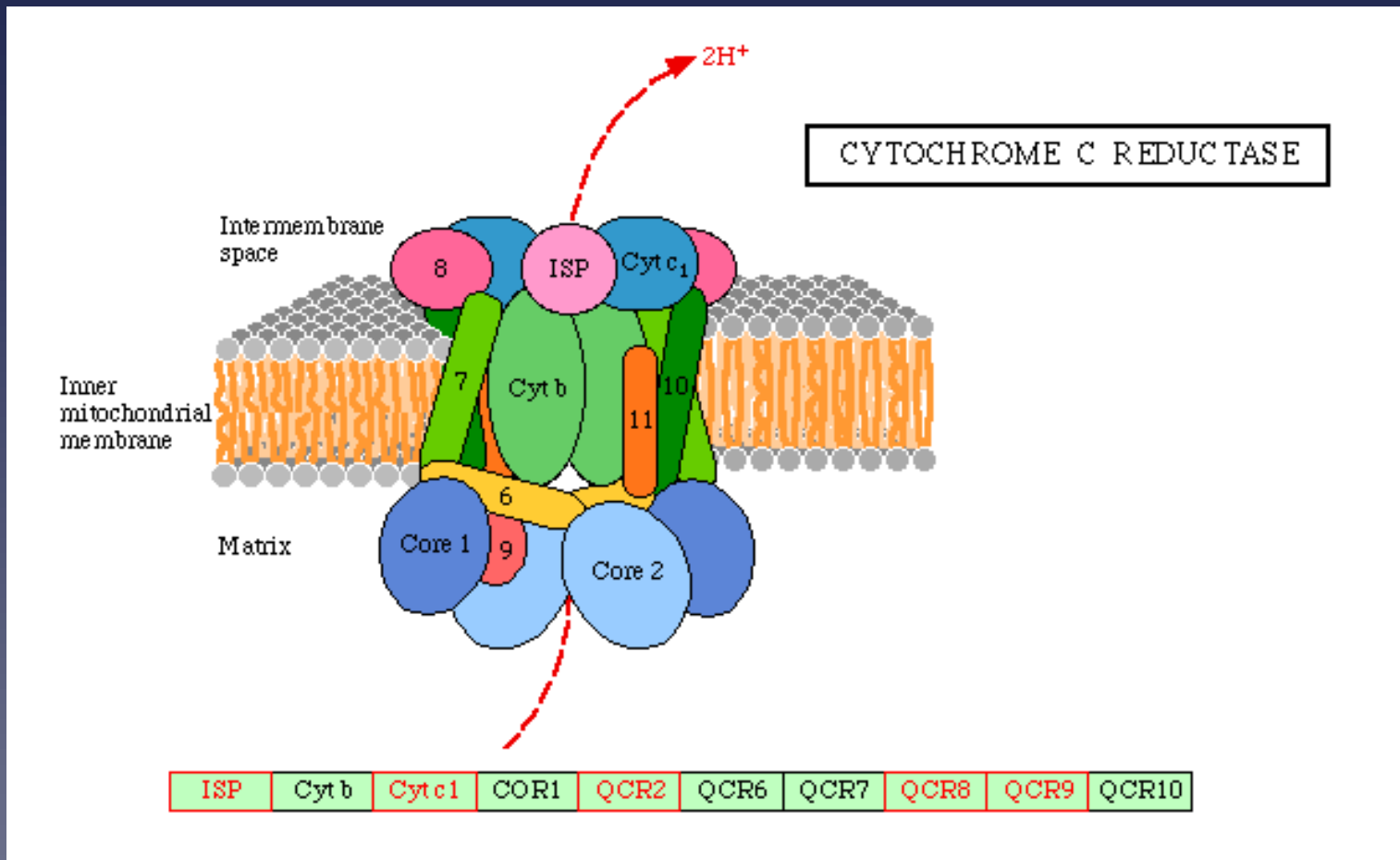


Related metabolic pathways

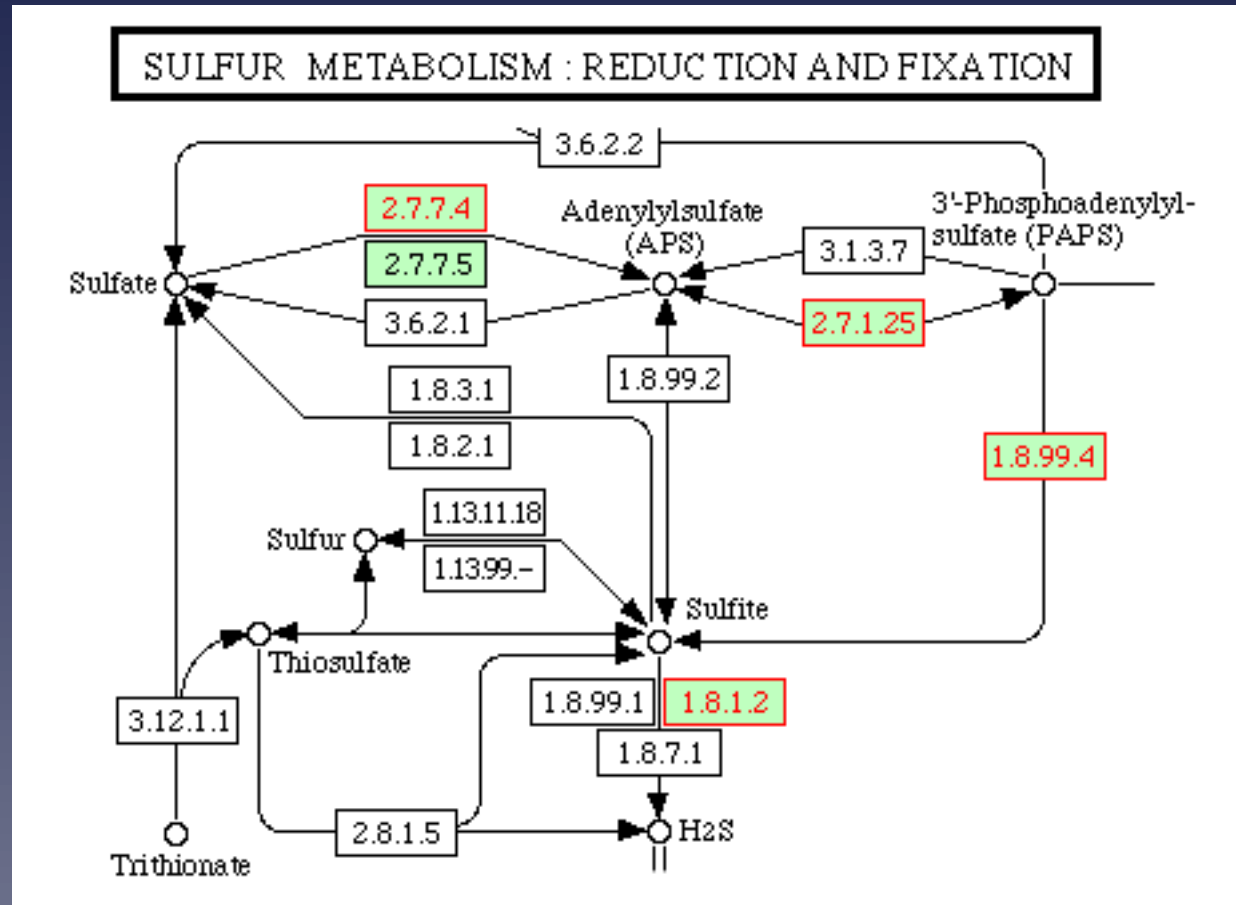
50 genes with highest $s_2 - s_1$ belong to:

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

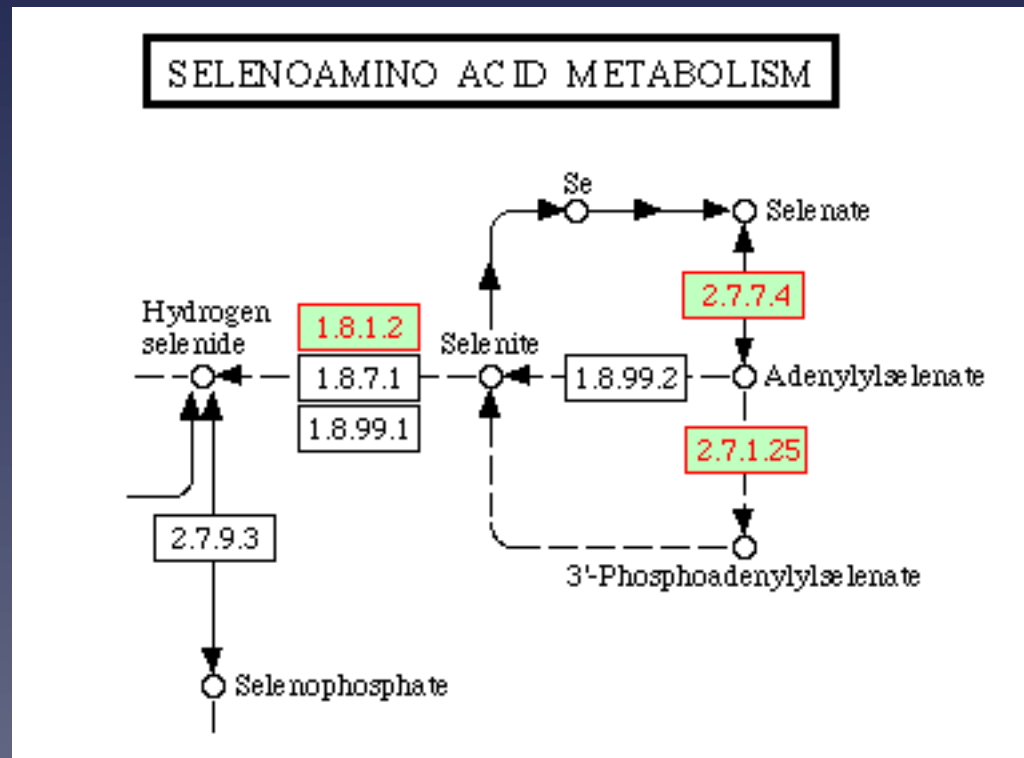
Related genes



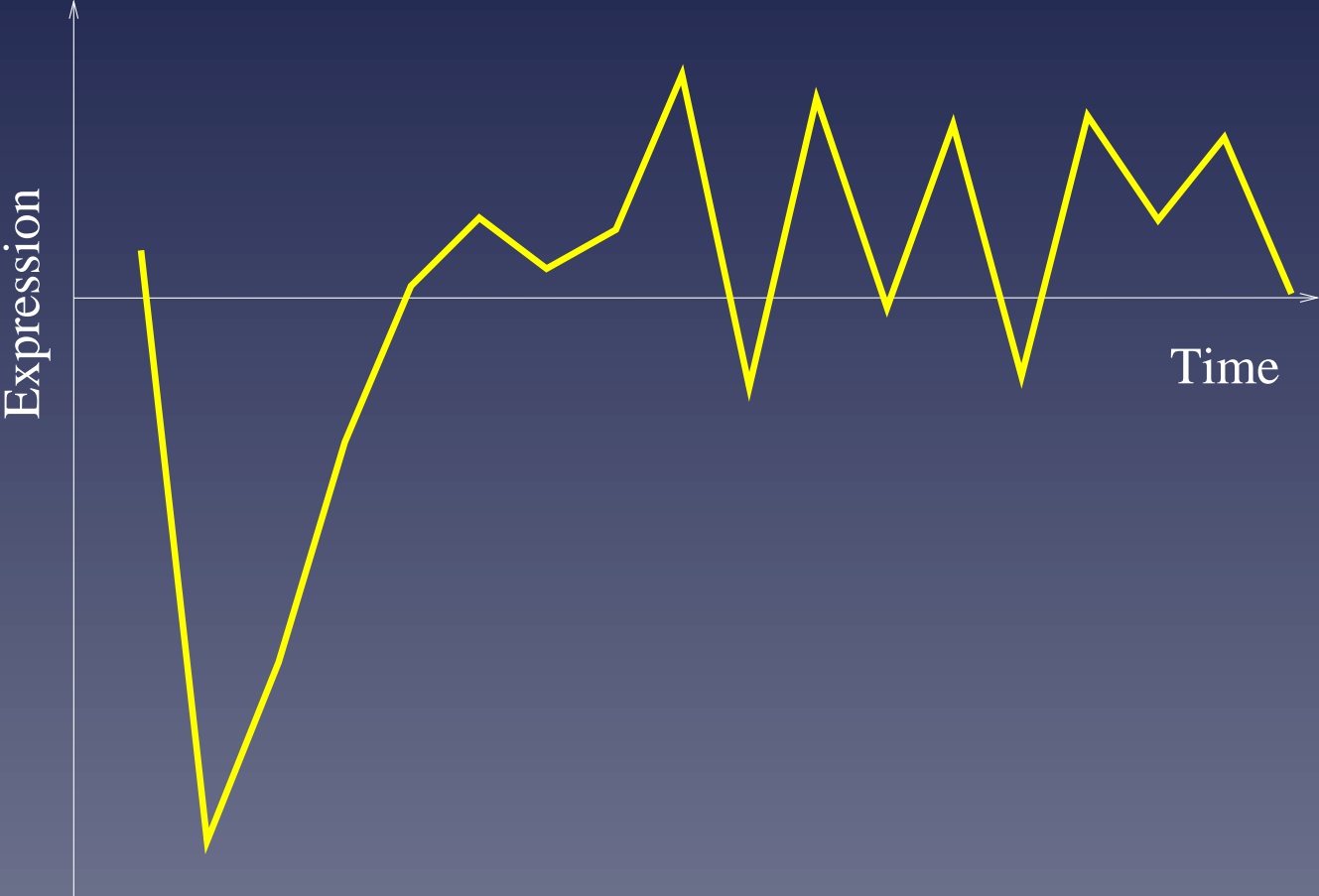
Related genes



Related genes



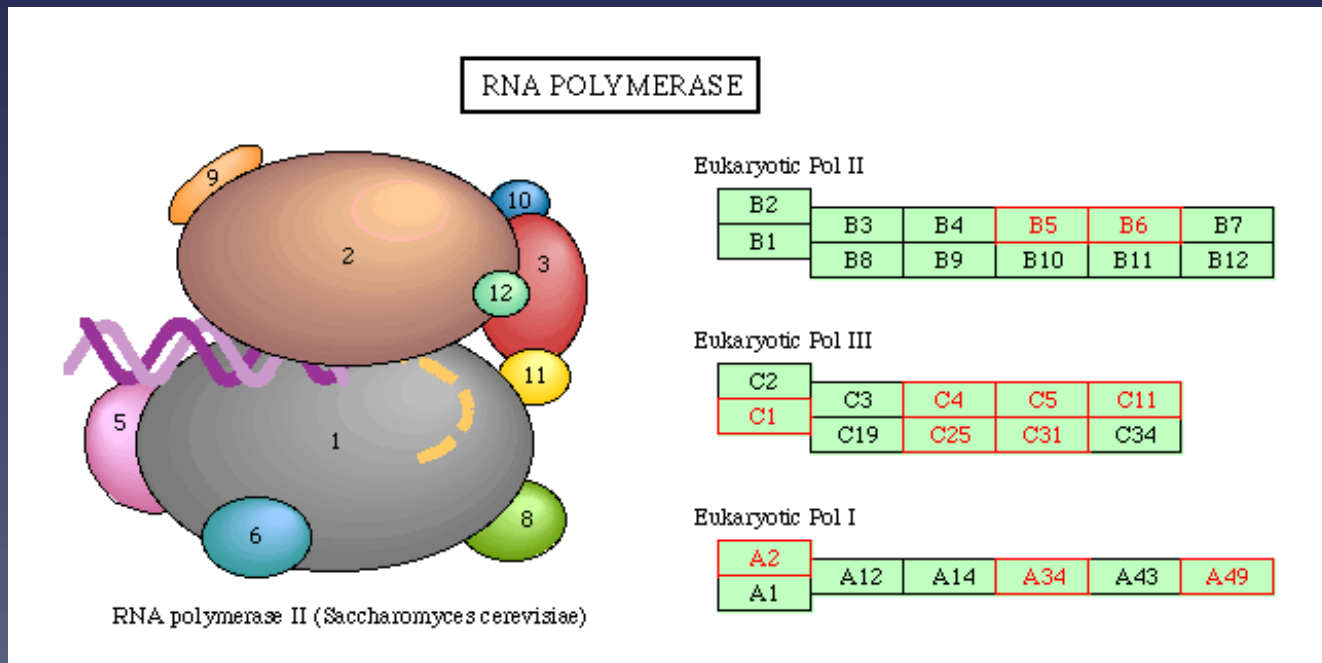
Opposite pattern



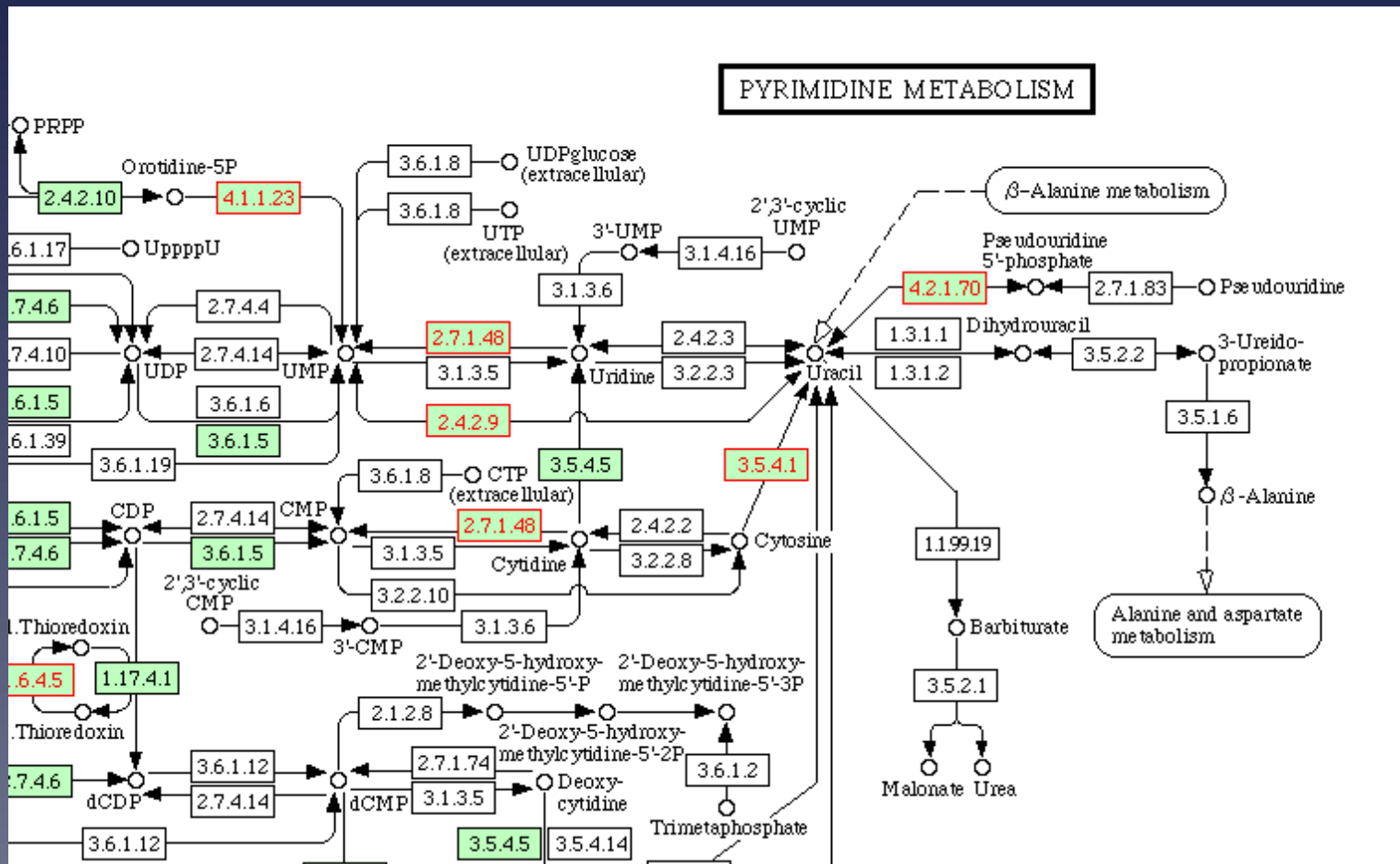
Related genes

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

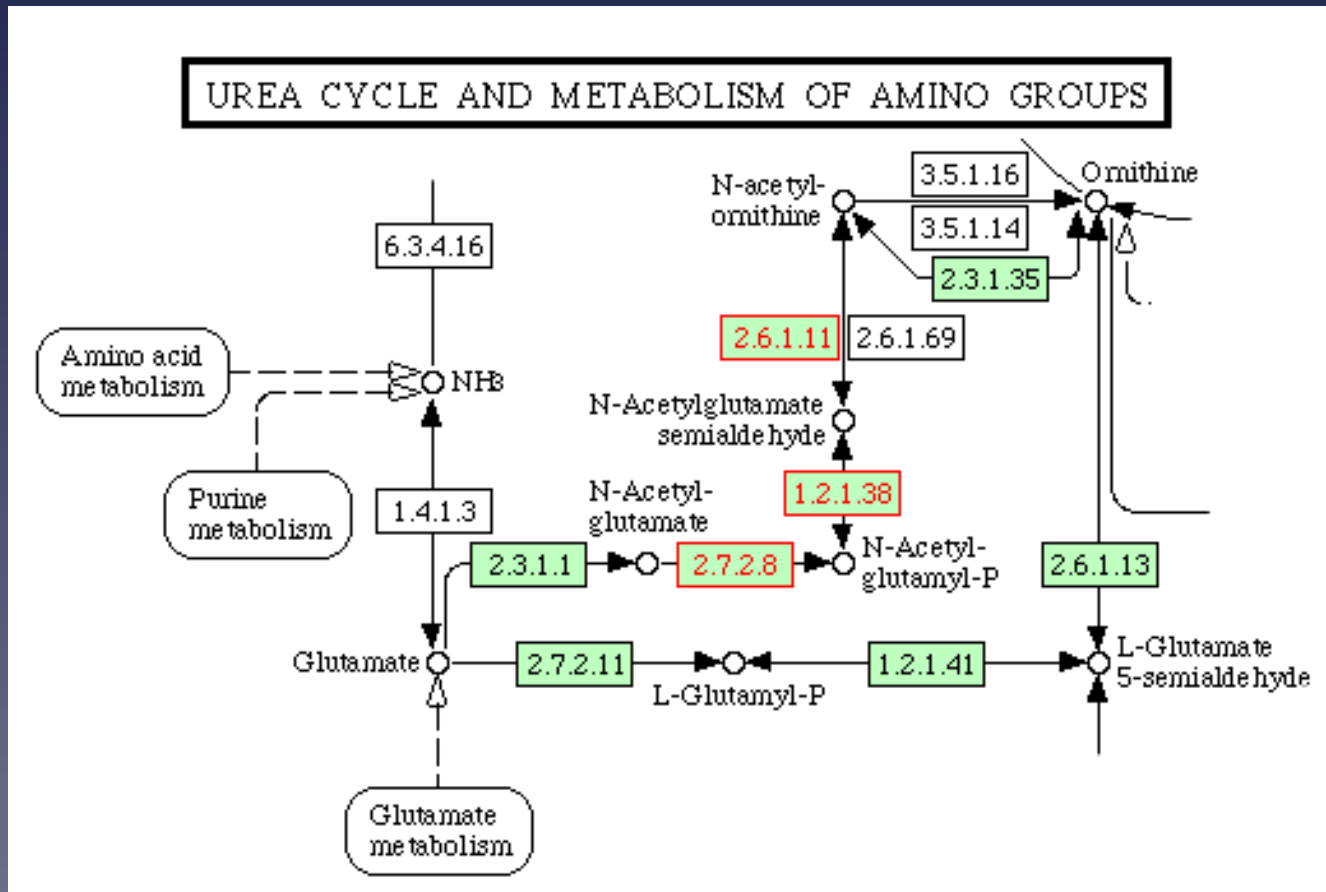
Related genes



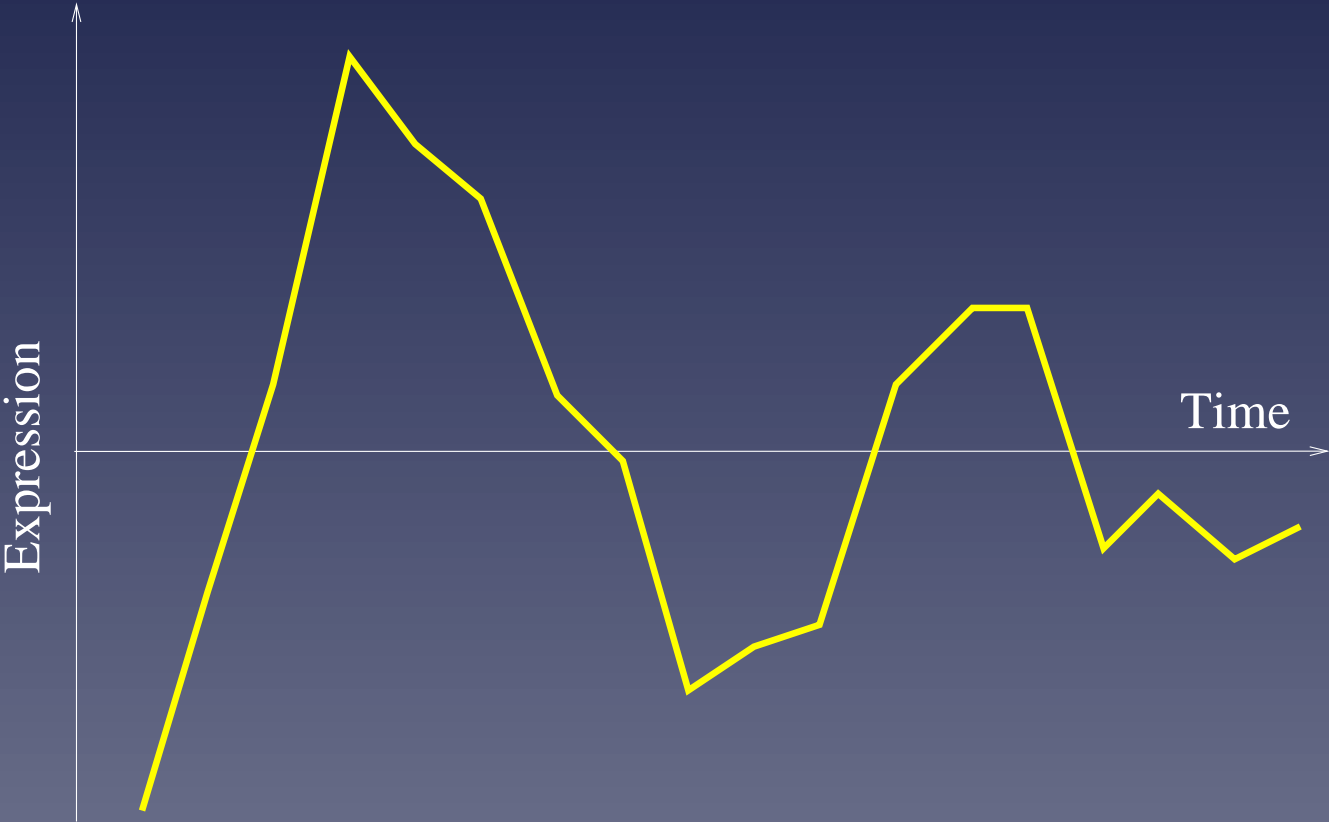
Related genes



Related genes



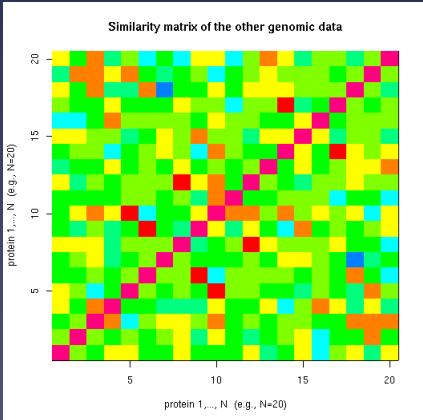
Second pattern



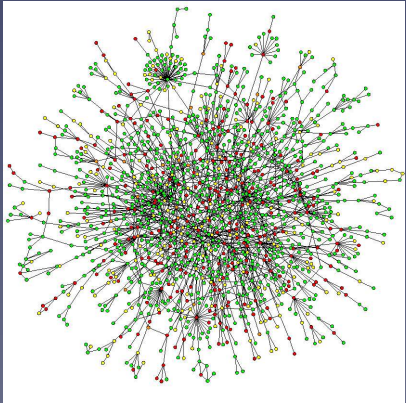
Part 4

Learning from several
heterogeneous data

Summary of the process



Features



Kernels

Several similarity kernels have been developed recently:

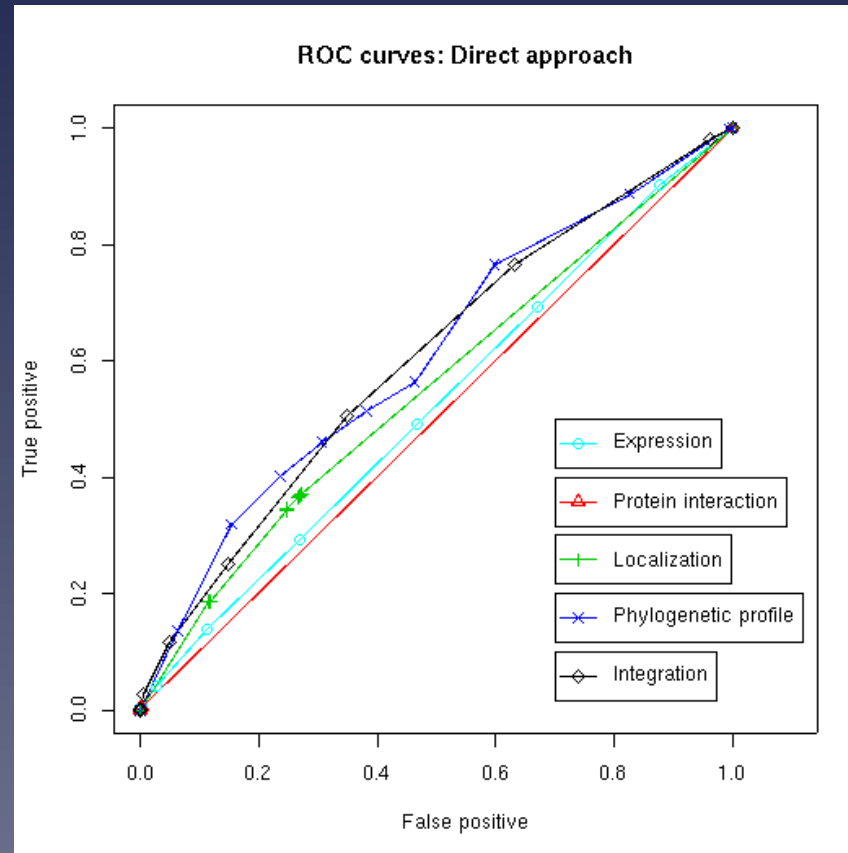
- for phylogenetic profiles (JPV. 2004)
- for gene sequences (Leslie et al. 2003, Saigo et al. 2004, ...)
- for nodes in a network (Kondor et al. 2000)

Learning from heterogeneous data

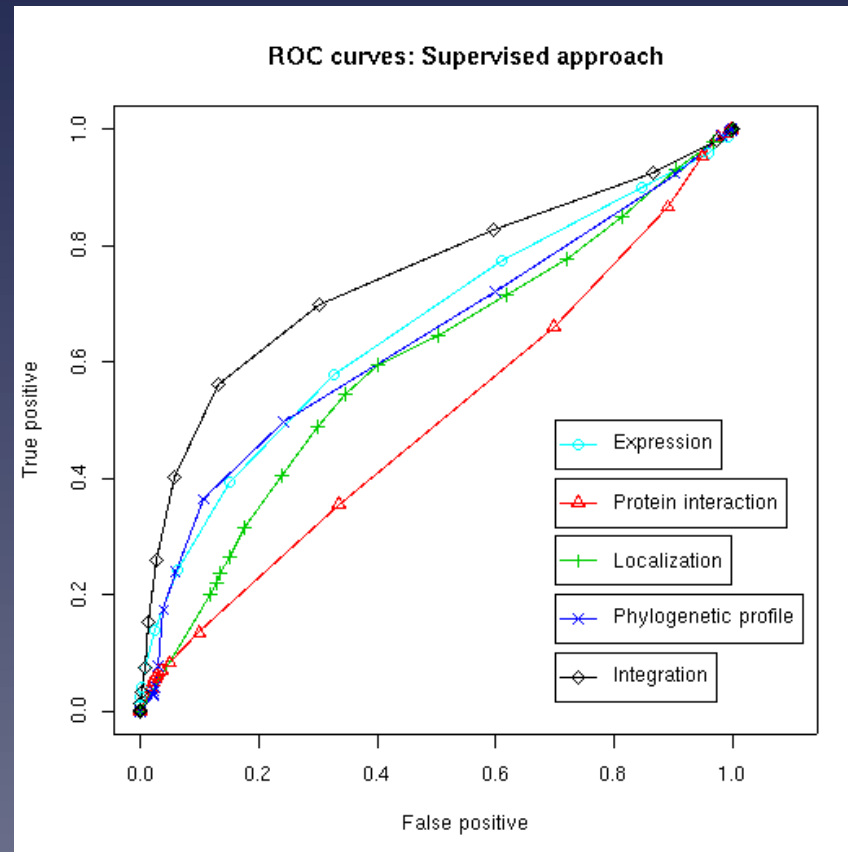
- Suppose **several data** are available about the genes, e.g., expression, localization, structure, predicted interaction etc...
- Each data can be represented by a **positive definite** similarity matrix K_1, \dots, K_p
- Kernel can be combined by various operations, e.g., addition:

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

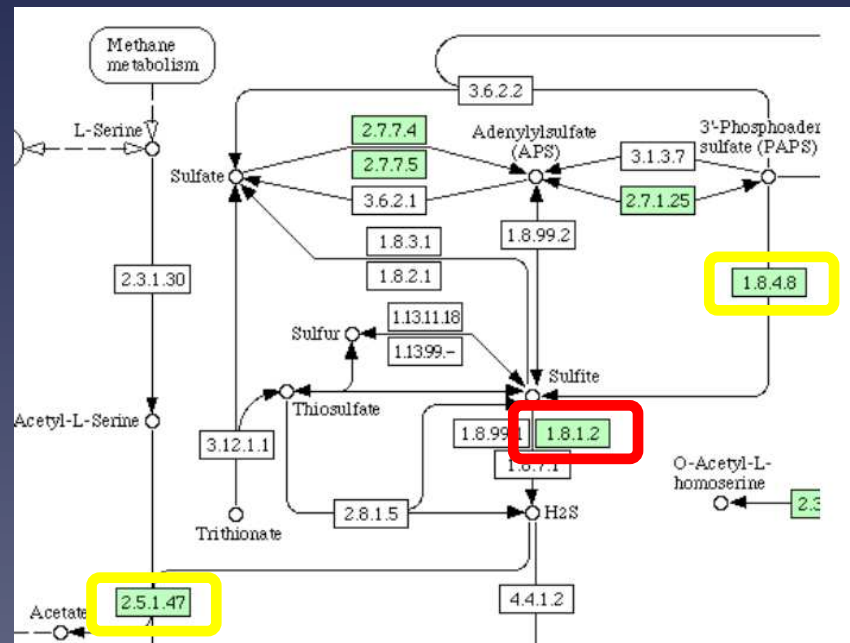
Learning from heterogeneous data (unsupervised)



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

1. **Supervised inference** is better than unsupervised

Conclusion

1. **Supervised inference** is better than unsupervised
2. Supervised graph inference can be performed by **distance metric learning**

Conclusion

1. **Supervised inference** is better than unsupervised
2. Supervised graph inference can be performed by **distance metric learning**
3. **Data integration with kernels** is simple and powerful

Conclusion

1. **Supervised inference** is better than unsupervised
2. Supervised graph inference can be performed by **distance metric learning**
3. **Data integration with kernels** is simple and powerful
4. **Few assumptions** about the network to infer (works well for the metabolic network and the protein interaction network)

Thanks

- Yoshihiro Yamanishi
- Computational biology at the Ecole des Mines

