Analysis and inference of gene networks from genomic data

Jean-Philippe.Vert@mines.org

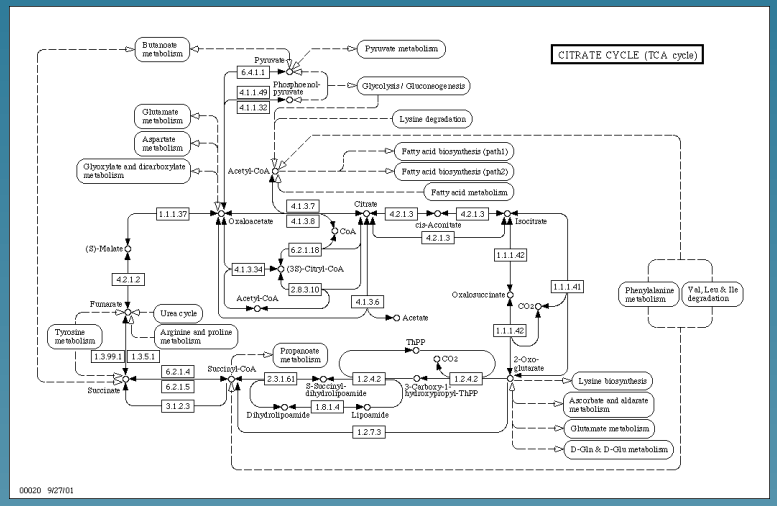
Ecole des Mines de Paris
Computational Biology group

SIG Meeting, Bioinformatics and Statistical Physics, Glasgow, UK, Julyl 30th, 2004.

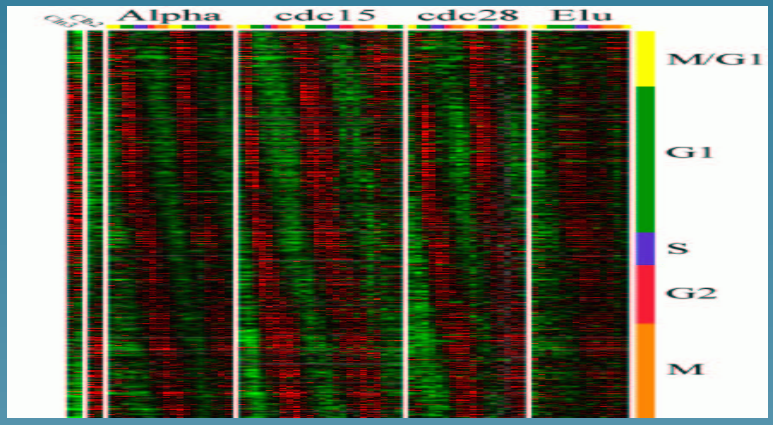
Motivations

- Many heterogeneous data about genes : sequences, expression, evolution, structures, etc...
- More and more data between genes: interactome, pathways, regulation etc...
- Goal: propose a formalism to compare and link these data.

Example:
Comparing gene expression and pathway databases



VS



Detect active pathways? Denoise expression data?
Denoise pathway database? Find new pathways?
Are there "correlations"?

Formalism

- N genes
- $x_1, \ldots, x_N \in \mathcal{X}$ the data about genes
  - ⋆ gene expression: $\mathcal{X} = \mathbb{R}^d$
  - ⋆ phylogenetic profile: $\mathcal{X} = \{0, 1\}^p$
  - ⋆ primary sequence: $\mathcal{X} = \mathcal{A}^*$
- $G = (V, E)$ a (weighted) graph, with $V = (v_1, \ldots, v_N)$ to represent the information between genes

3 related questions

- How to quantify how much the data "fits" the graph?
- How to infer features $f : \mathcal{X} \to \mathbb{R}$ that "fit" the graph ("graph-driven feature construction")?
- How to update/correct the graph from the genomic data about genes (e.g., to add new nodes to the graph)?

Part 1

Graph-driven feature extraction

Linear features for $\mathcal{X} = \mathbb{R}^d$

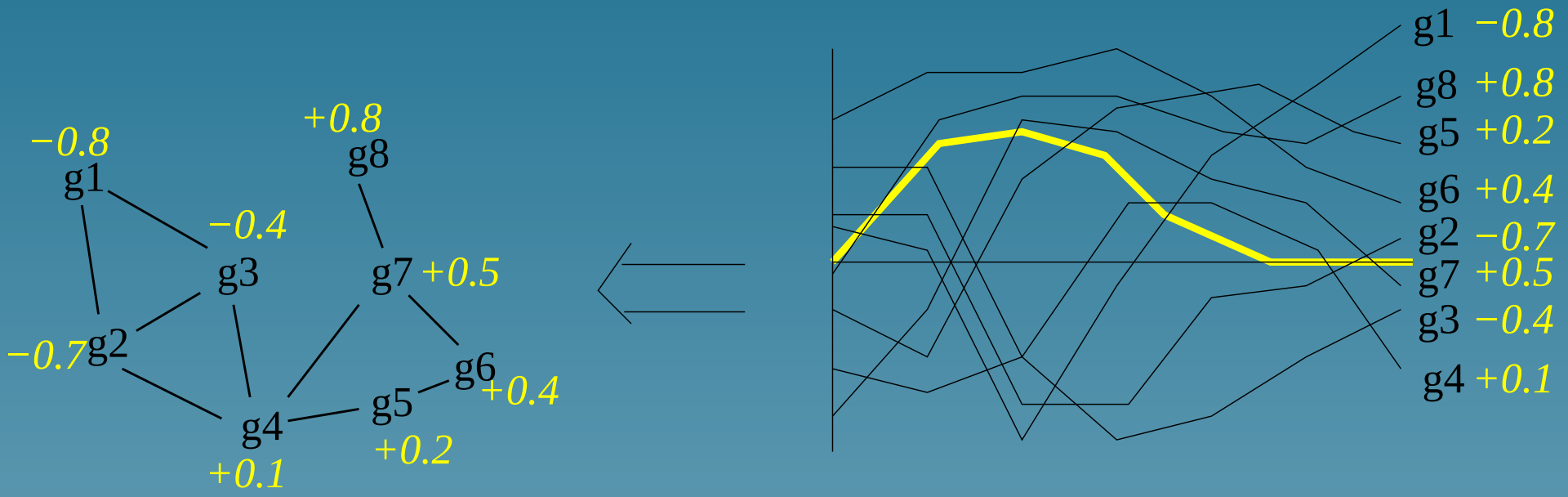
When $\mathcal{X} = \mathbb{R}^d$, let us consider linear features defined for any $w \in \mathbb{R}^d$ by:

$$\forall x \in \mathcal{X}, \quad f_w(x) = w \cdot x.$$

Principal component analysis (PCA) extract features $(w_1, \ldots, w_d)$ by:

$$\begin{aligned} w_i &= \underset{w \perp \{w_1, \ldots, w_{i-1}\}, ||w||=1}{\arg\max} \hat{\mathsf{var}}(f_w) \\ &= \underset{w \perp \{w_1, \ldots, w_{i-1}\}, \hat{\mathsf{var}}(f_w)=1}{\arg\min} ||w||^2. \end{aligned} \tag{1}$$

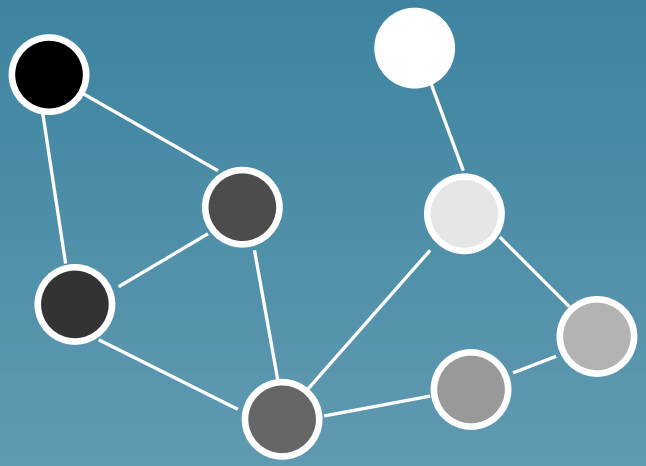
Mapping f_w onto the gene network



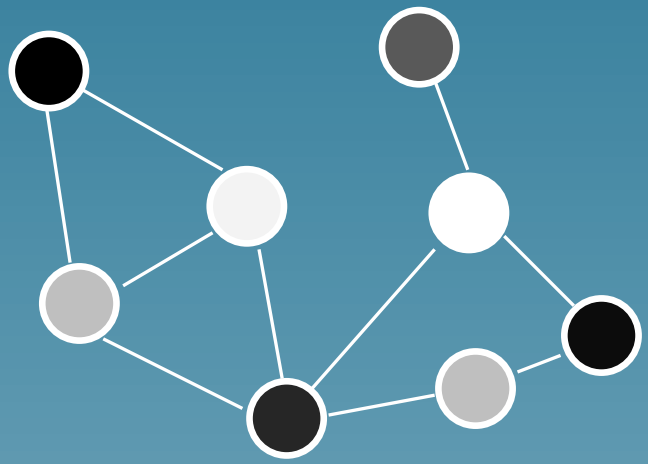
Does it look interesting or not?

Important hypothesis

A feature f_w is relevant ("fits the graph") if it varies "smoothly" on the graph

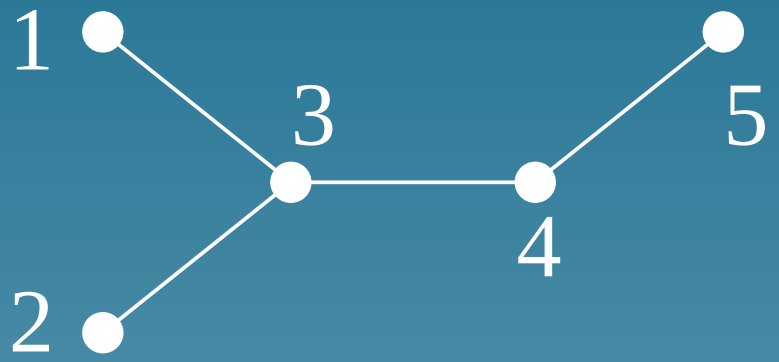


Smooth



Rugged

Graph Laplacian $L = D - A$



$$L = \begin{pmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ -1 & -1 & 3 & -1 & 0 \\ 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & 0 & -1 & 1 \end{pmatrix}$$

Smoothness quantification

For a feature $f : \mathcal{X} \to \mathbb{R}$ with unit variance,

$$h_2(f) = \sum_{i \sim j} (f(x_i) - f(x_j))^2 = f^\top L f$$

or

$$h_2(f) = \sum_i \hat{f}_{x_i}^2 e^{\beta \omega_i} = f^\top \exp(\beta L) f$$

is small when f is smooth

Graph-driven PCA

In order to extract features that better "fit" the graph, we can modify PCA as follows:

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

The trade-off between catching variance and fitting the data is controlled by the parameter λ:

- $\lambda \to +\infty$: PCA
- $\lambda \to 0$: second smallest eigenvector of the graph

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 $f : \mathcal{X} \to \mathbb{R}$ 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).

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:

$$\underset{f \perp \{f_1, \ldots, f_{i-1}\}, \hat{\mathrm{var}}(f)=1}{\arg\min} ||f||_k^2.$$

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:

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

- 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

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

Solving the problem (cont.)

- The problem can then be rewritten:

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

  where K_V is the centered $n \times n$ Gram matrix

- It is equivalent to solving the generalized eigenvalue problem:

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

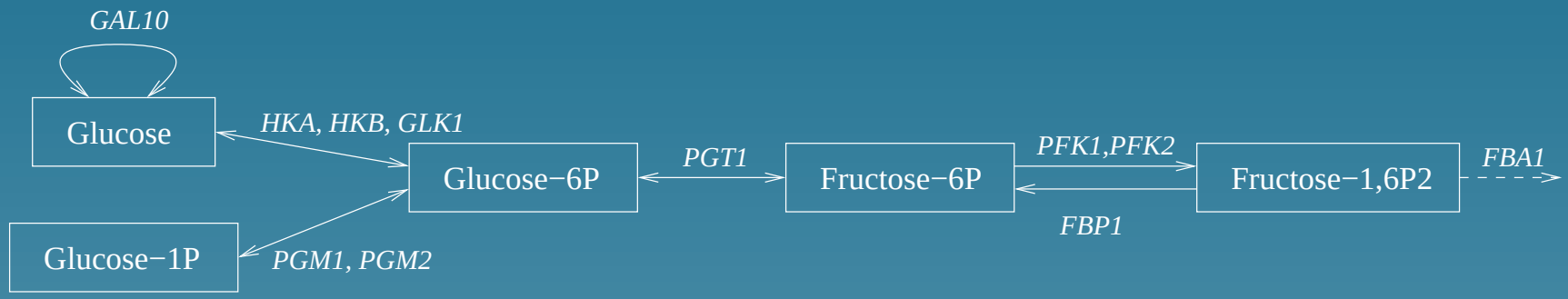
Part 3

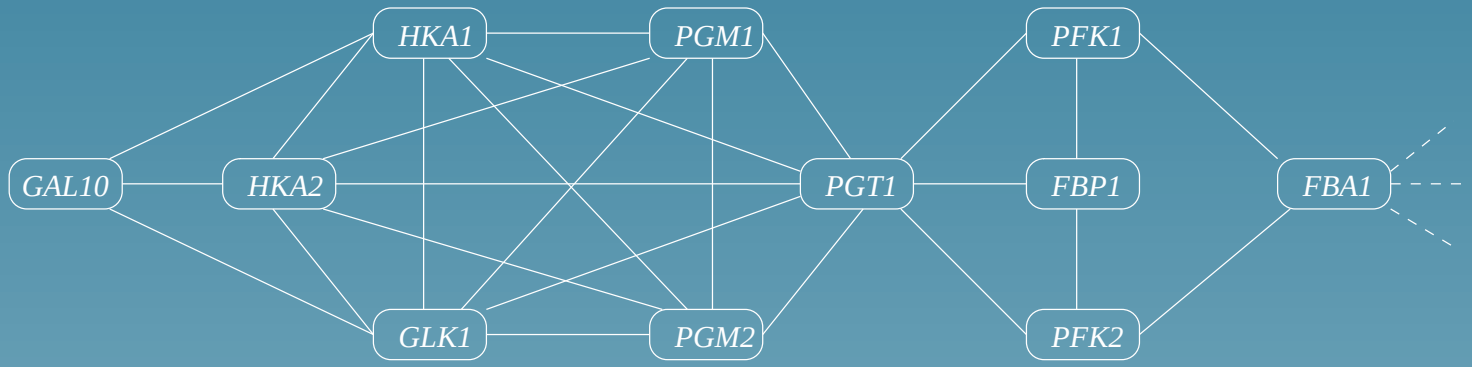
Experiments

Data

- 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

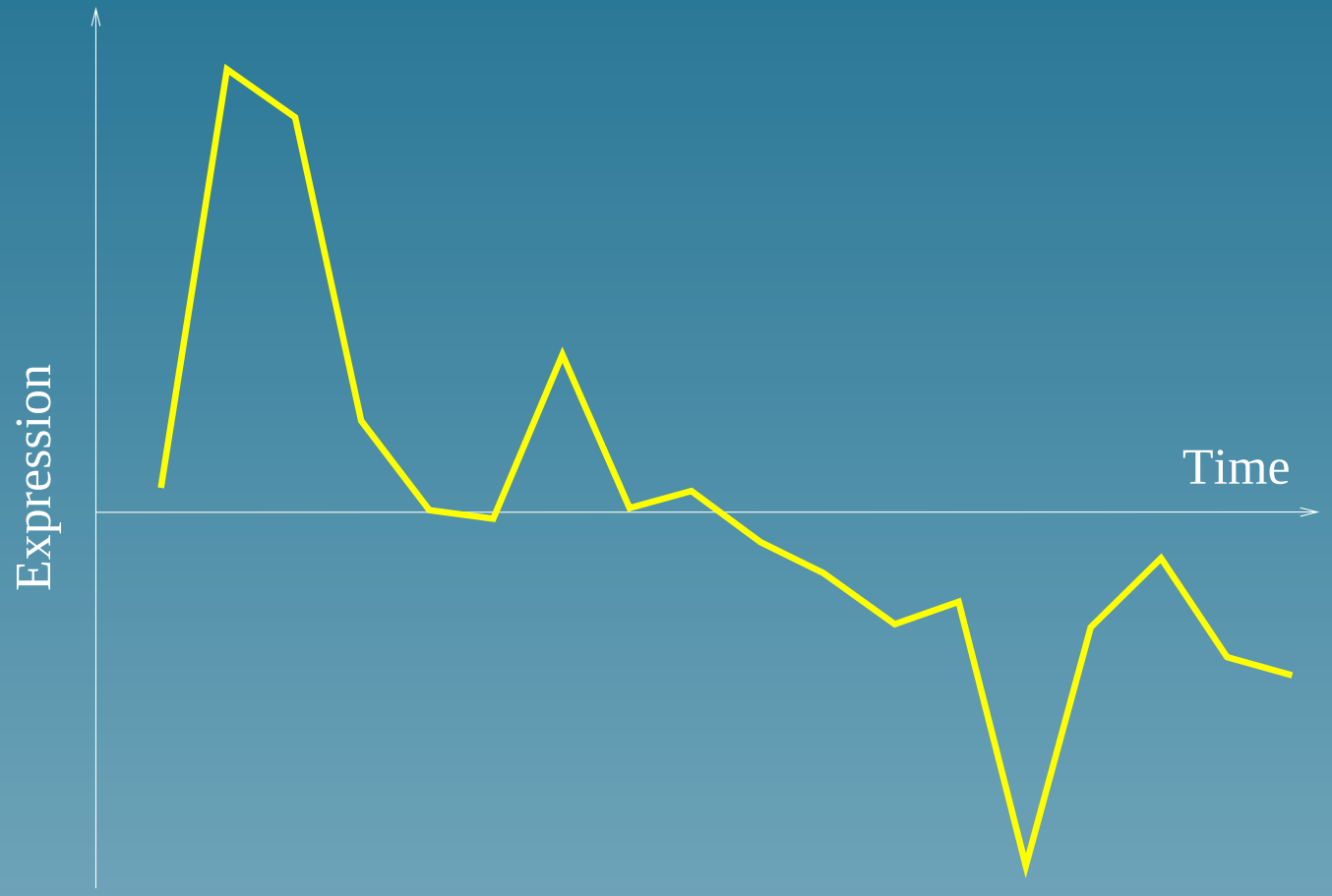
The metabolic gene network





Link two genes when they can catalyze two successive reactions

First pattern of expression

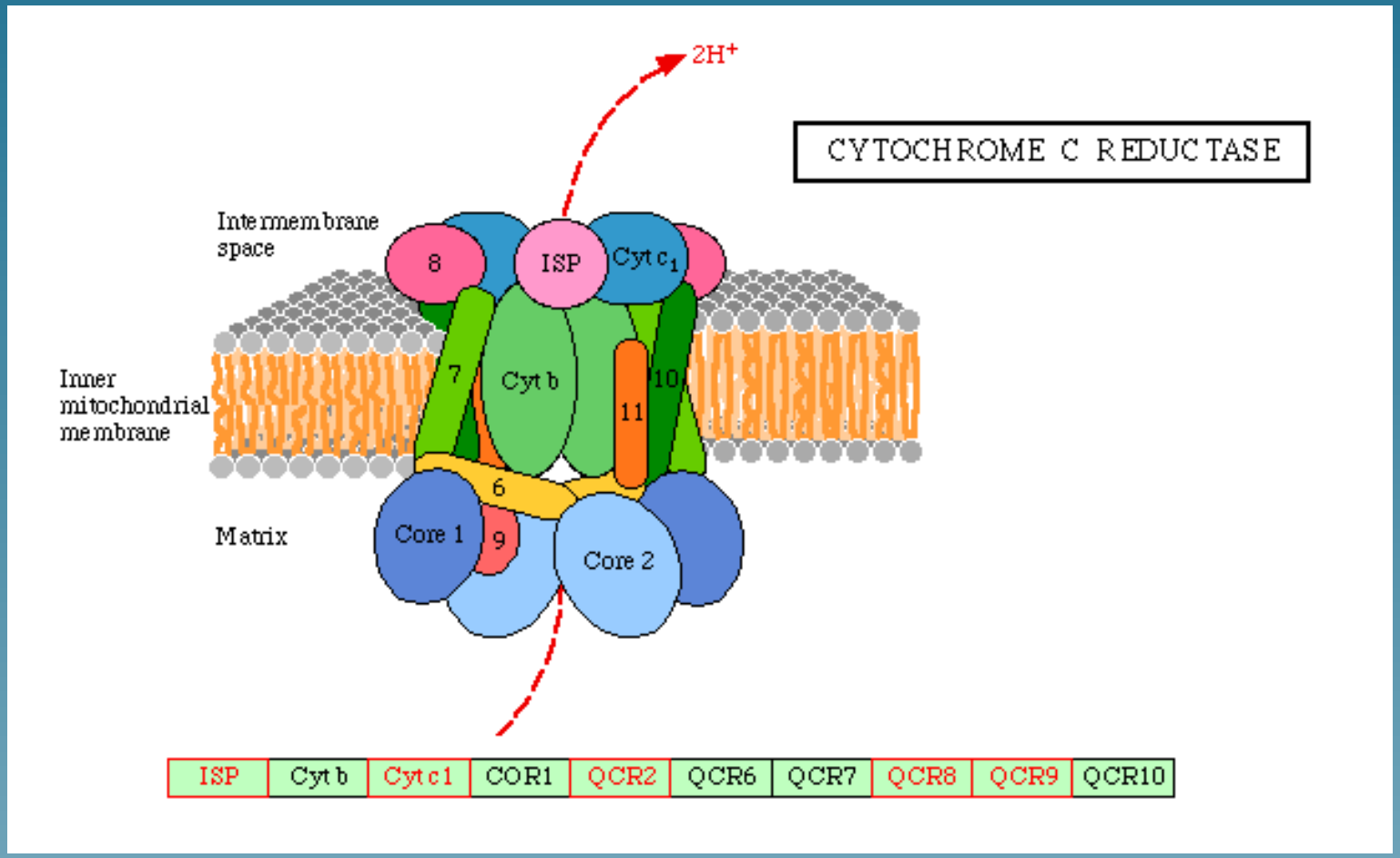


Related metabolic pathways

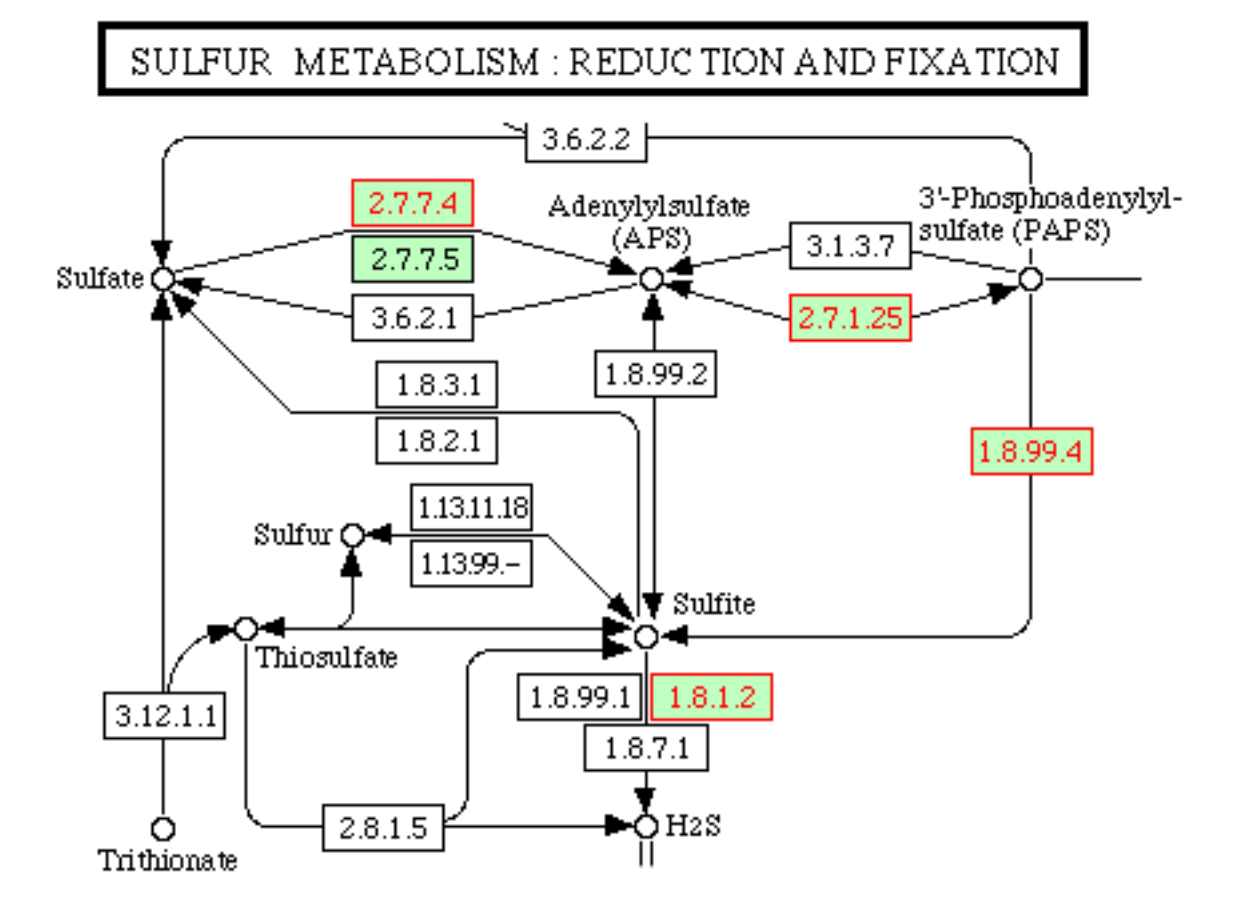
50 genes with highest $s_2 - s_1$ belong to:

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

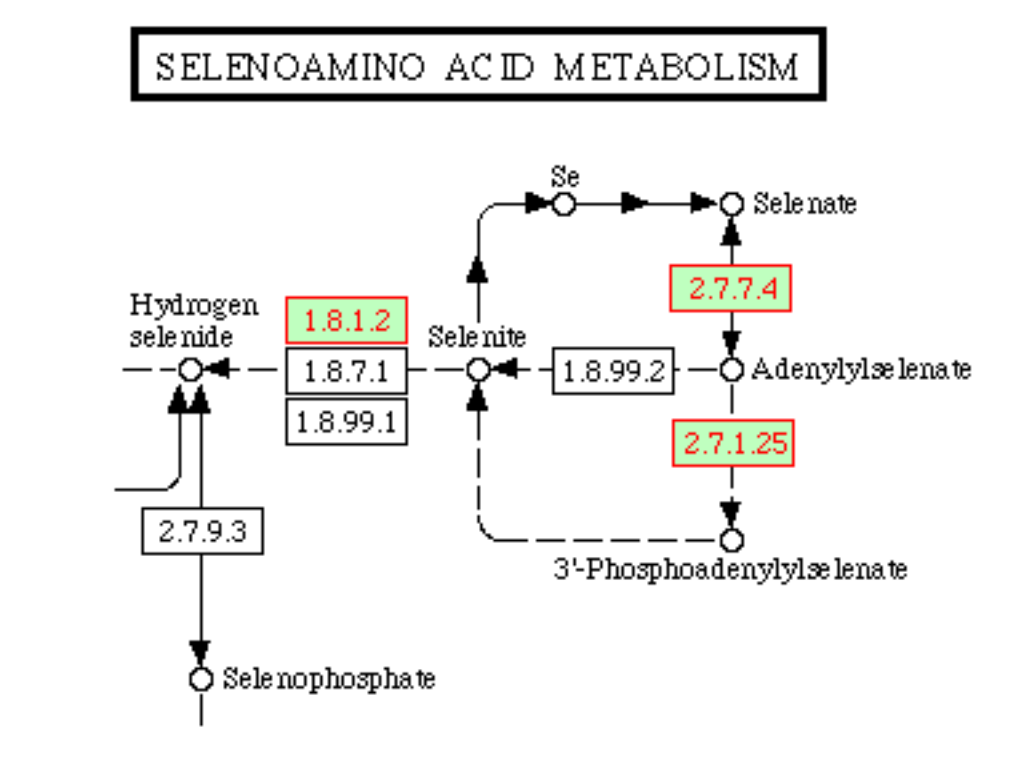
Related genes



Related genes



Related genes



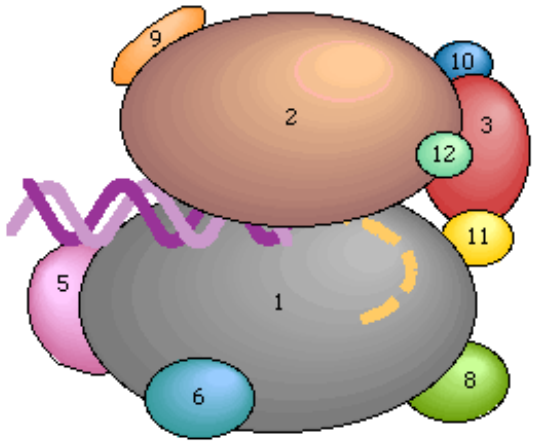
Opposite pattern

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

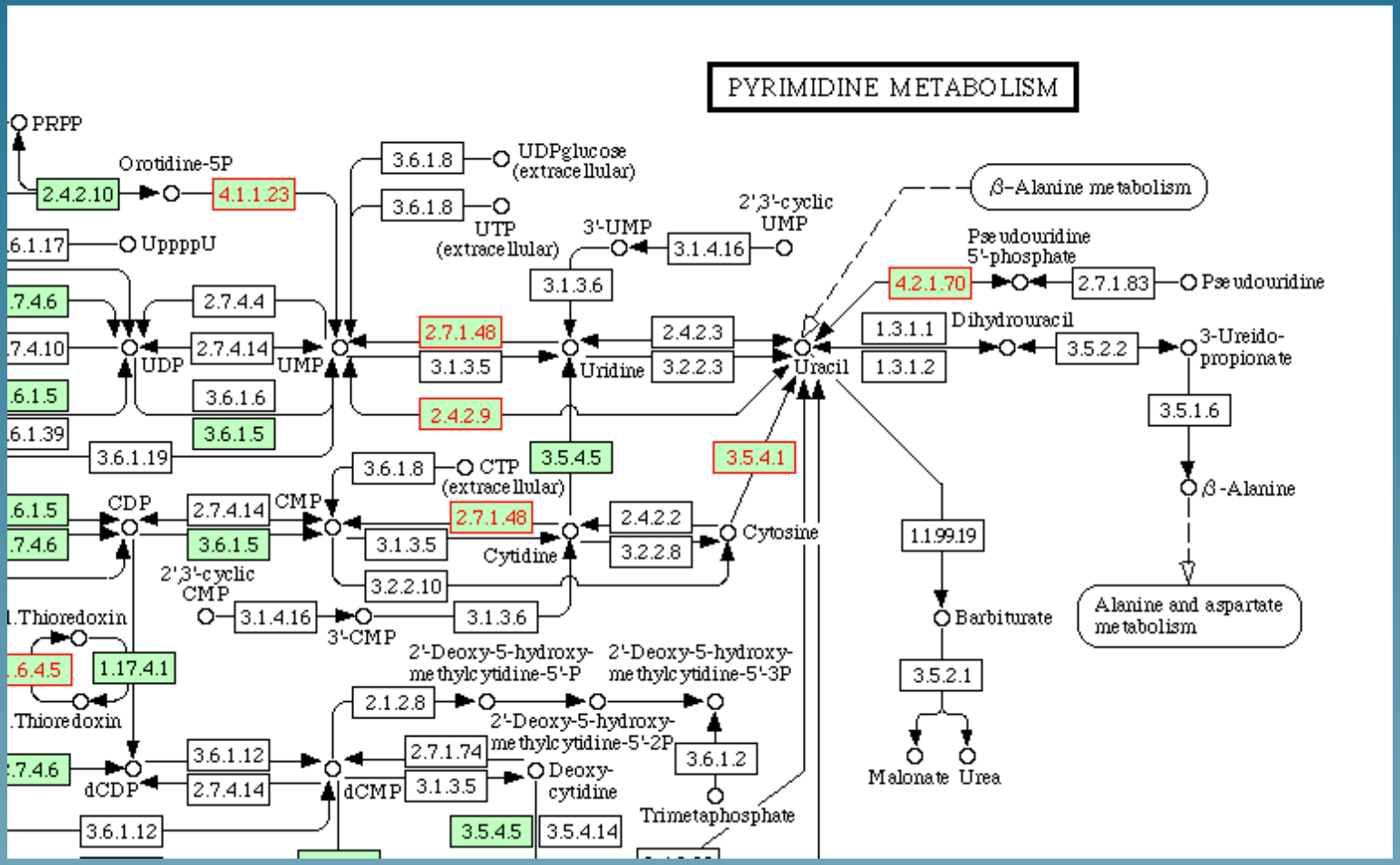
Eukaryotic Pol III

C2	C3	C4	C5	C11
C1	C19	C25	C31	C34

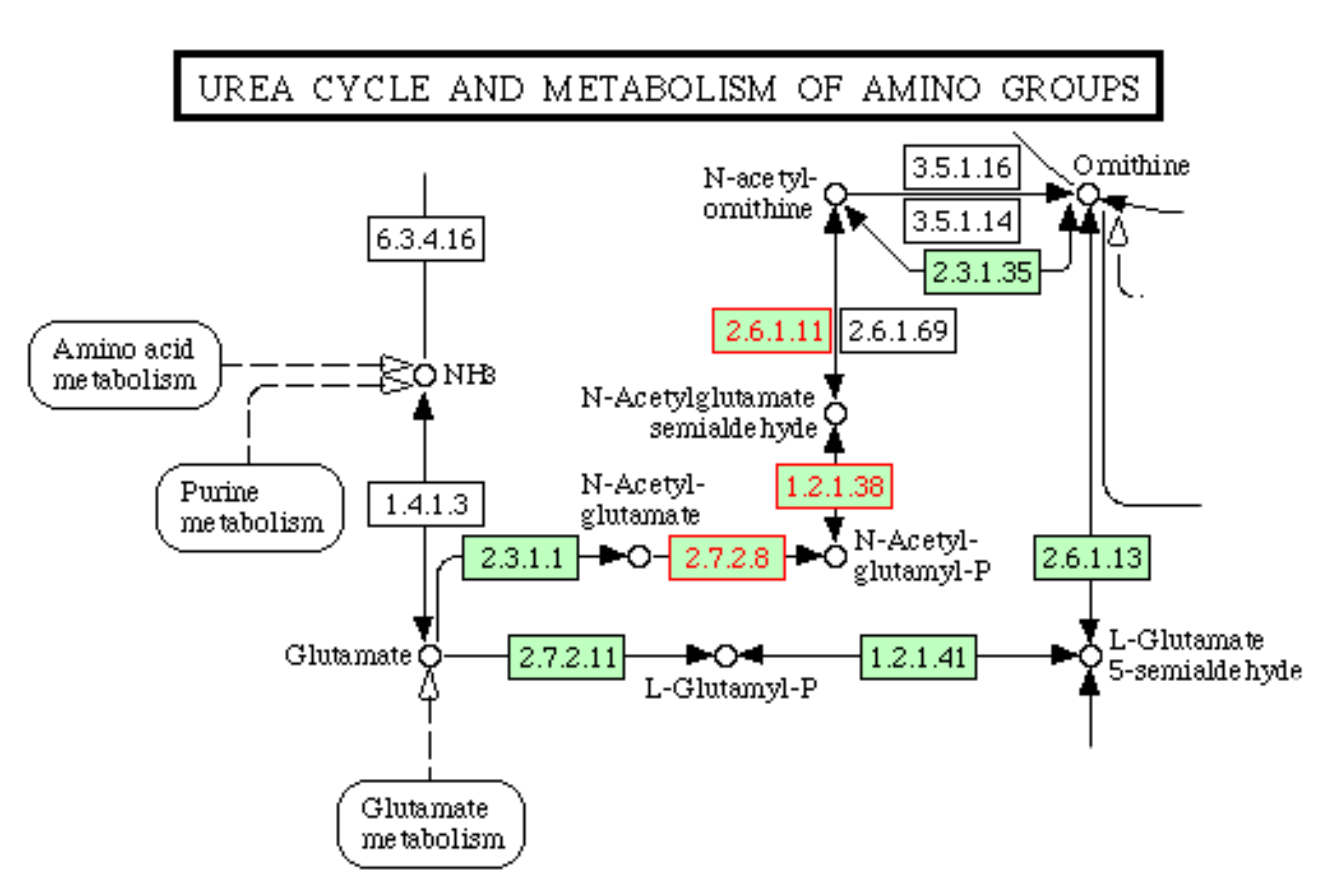
Eukaryotic Pol I

A2	A12	A14	A34	A43	A49
A1					

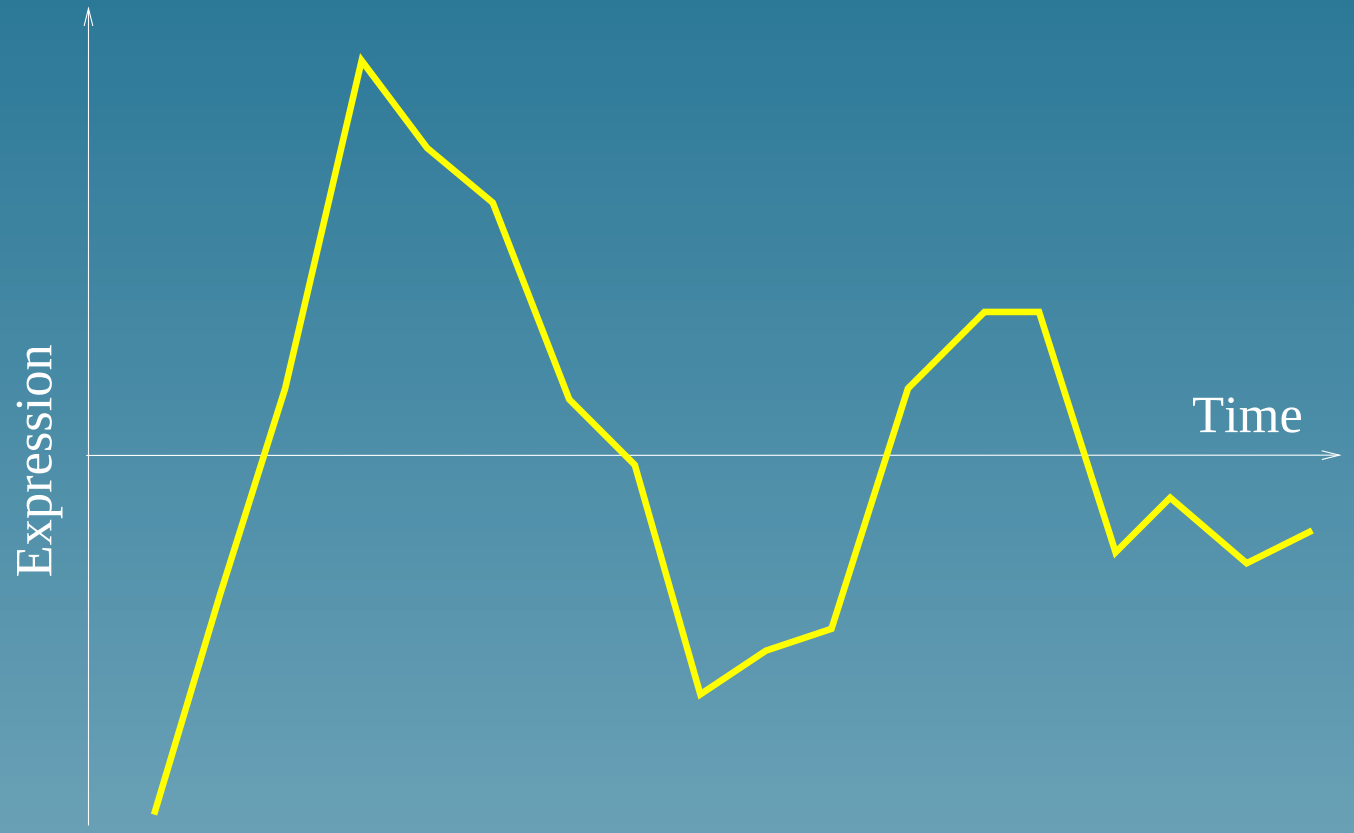
Related genes



Related genes



Second pattern



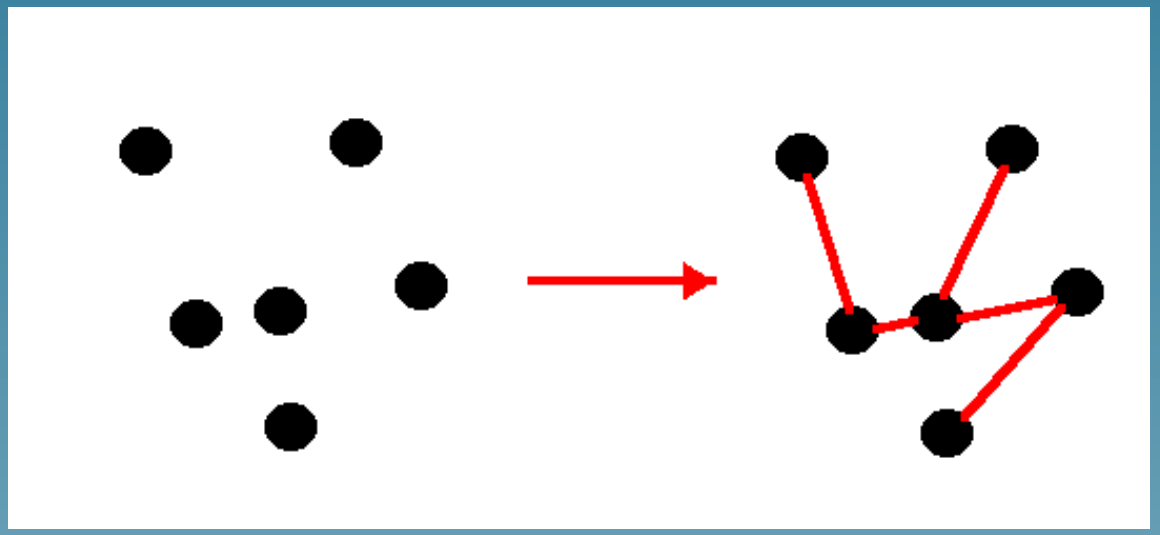
Part 4

Inferring new pathways

(with Y.Yamanishi)

The network inference problem

Given some measurement/observation about the genes (sequences, structure, expression, ...), infer "the" gene network



Related approaches

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

A direct (unsupervised) approach

- Let $K(x, y)$ be a measure of similarity (a kernel) between genes x and y based on available measurements, e.g.,

$$K(x, y) = \exp\left(-\frac{||e(x) - e(y)||^2}{2\sigma^2}\right)$$

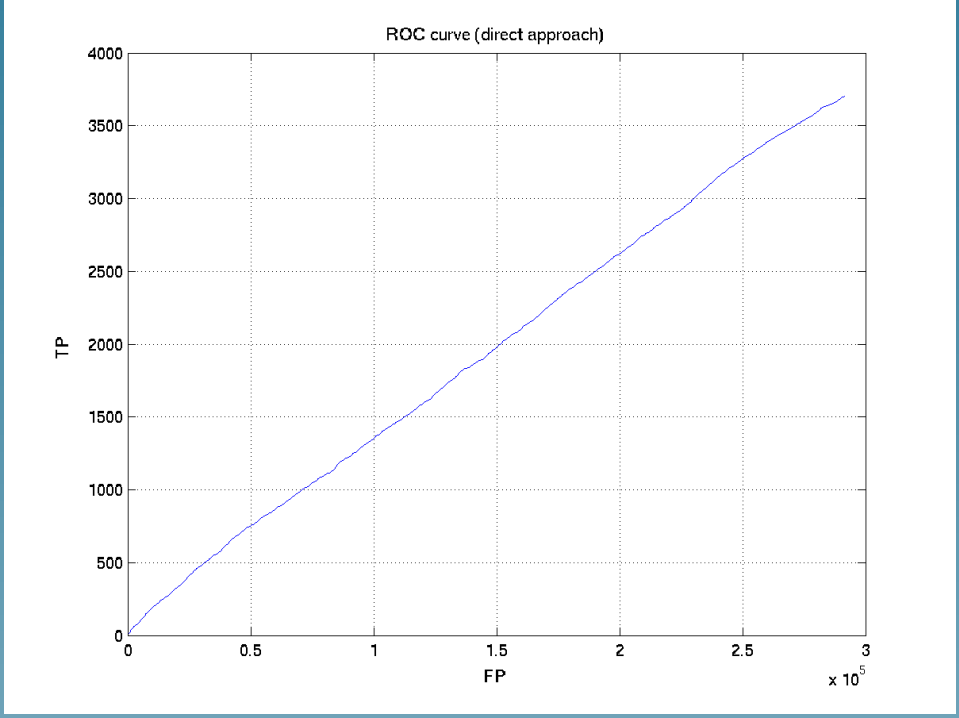
- For a set of n genes $\{x_1, \ldots, x_n\}$, let K be the $n \times n$ matrix of pairwise similarity (Gram matrix)

- Direct strategy: add edges between genes by decreasing similarity.

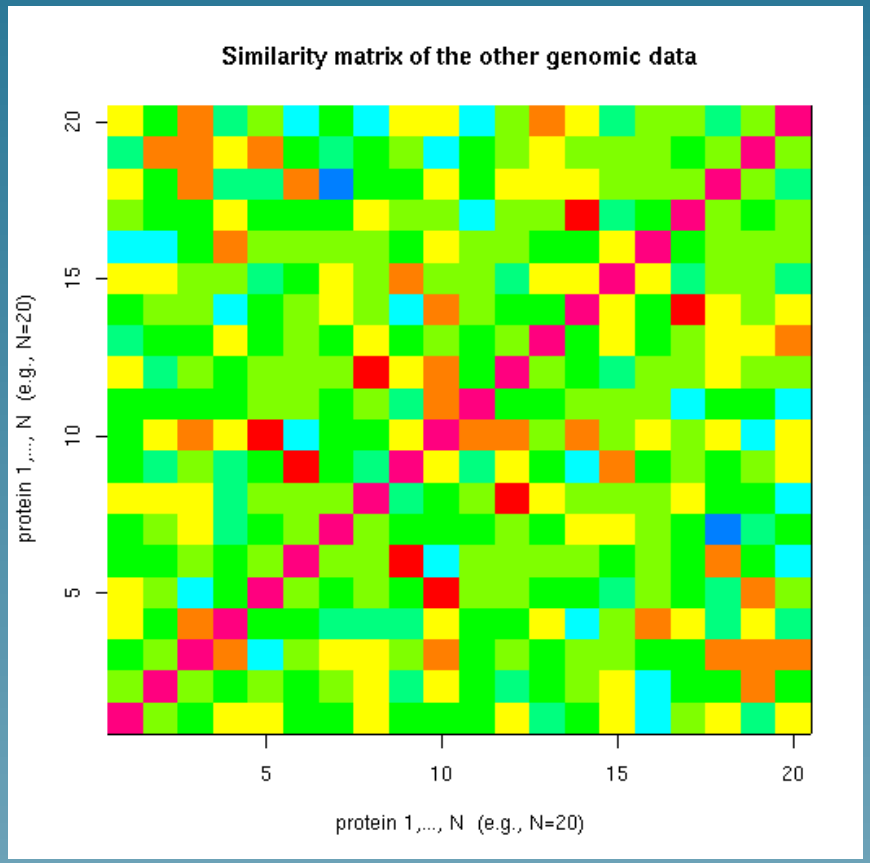
Example of similarity matrix

Evaluation of the direct approach

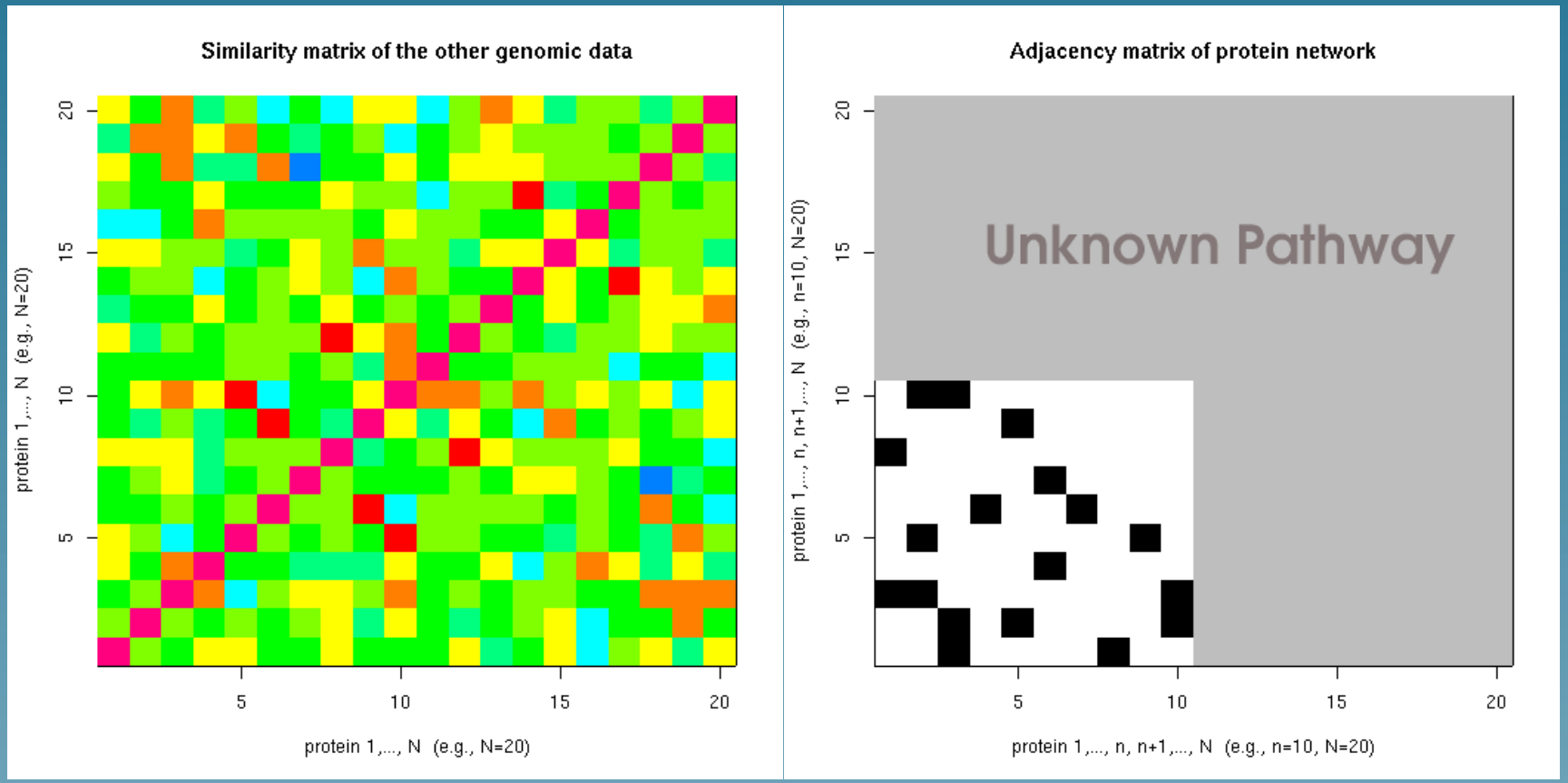
The metabolic network of the yeast involves 769 genes. Each gene is represented by 157 expression measurements. (ROC=0.52)



The supervised gene inference problem



The supervised gene inference problem



The idea in a nutshell

- Use the known network to define a more relevant measure of similarity
- For any positive definite similarity $n \times n$ matrix, there exists a representation as n-dimensional vectors such that the matrix similarity is exactly the similarity between vectors.
- In this space, look for projections onto small-dimensional spaces that better fit the known network.

A two-step strategy

- First map any gene x onto a vector

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

A two-step strategy

- First map any gene x onto a vector

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

- Then apply the direct strategy to reconstruct the graph from the images $\{\Phi(x_1), \ldots, \Phi(x_n)\}$

A two-step strategy

- First map any gene x onto a vector

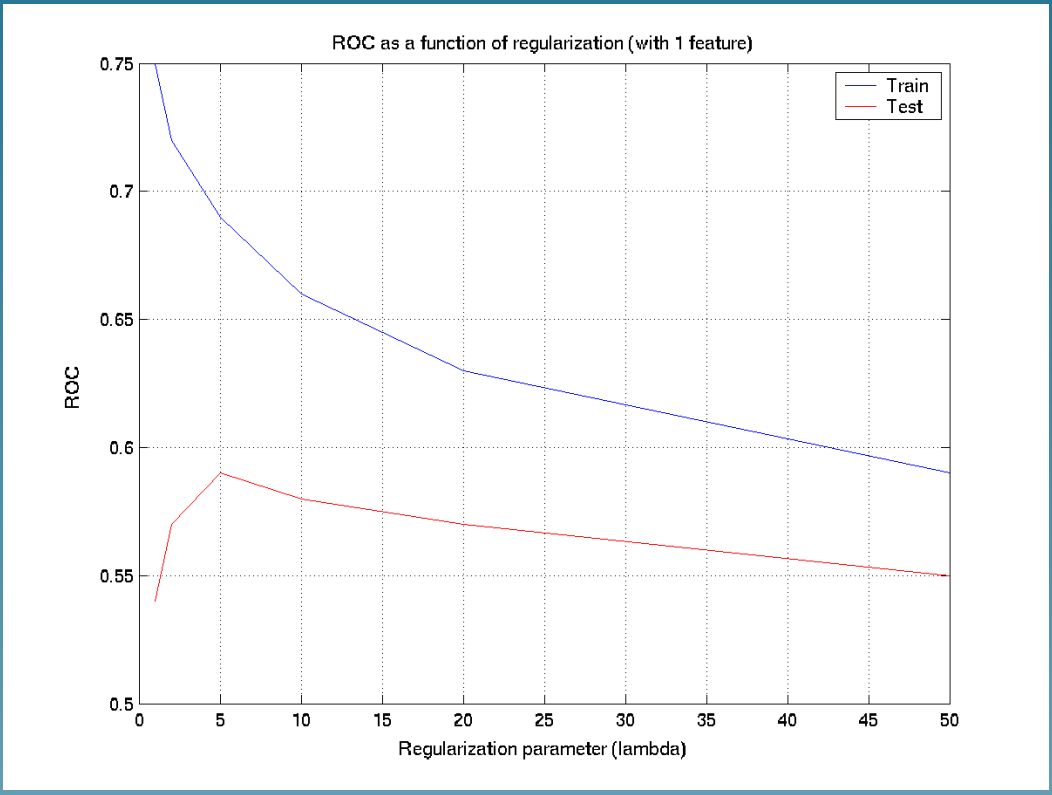
$$\Phi(x) = (f_1(x), \ldots, f_d(x))' \in \mathbb{R}^d$$

- Then apply the direct strategy to reconstruct the graph from the images $\{\Phi(x_1), \ldots, \Phi(x_n)\}$
- The functions $f_1, \ldots, f_d$ can be learned from the knowledge of the graph on the first n genes

Choice of f

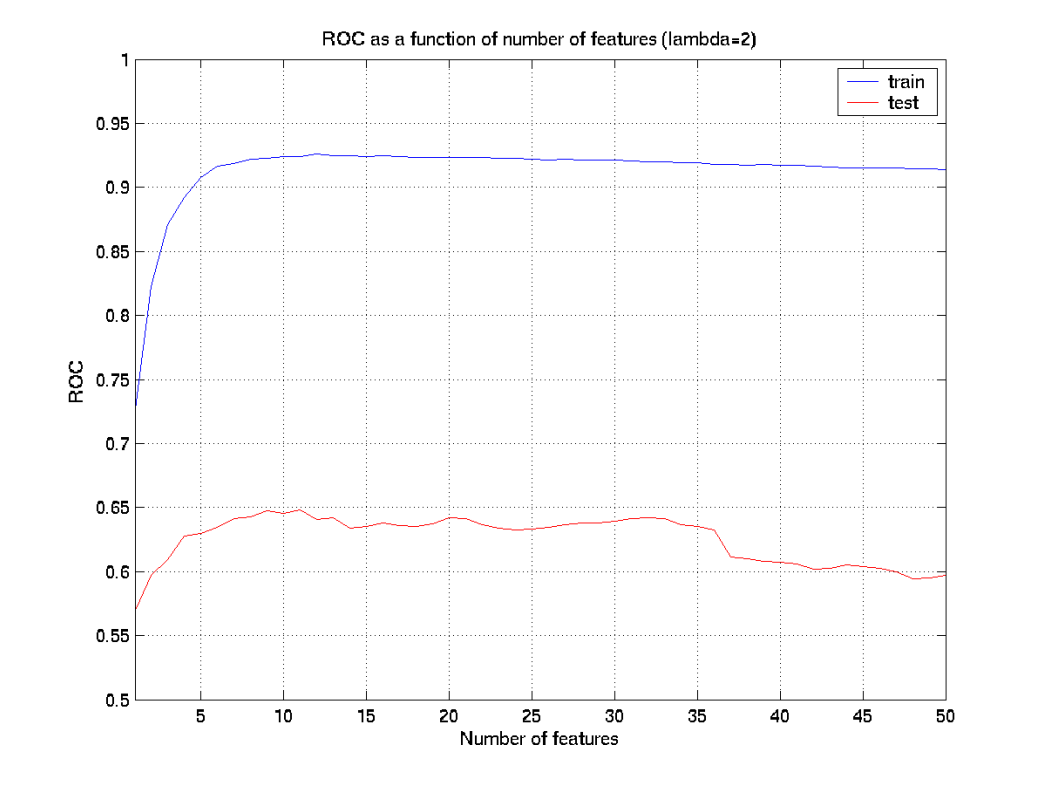
- A feature $f : \mathcal{X} \to \mathbb{R}$ is good on the training set if connected genes have similar value.
- This is exactly what we did in the previous part!
- So use the features already extracted to map new genes onto a vector space by projection

Evaluation of the supervised approach: effect of λ



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

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

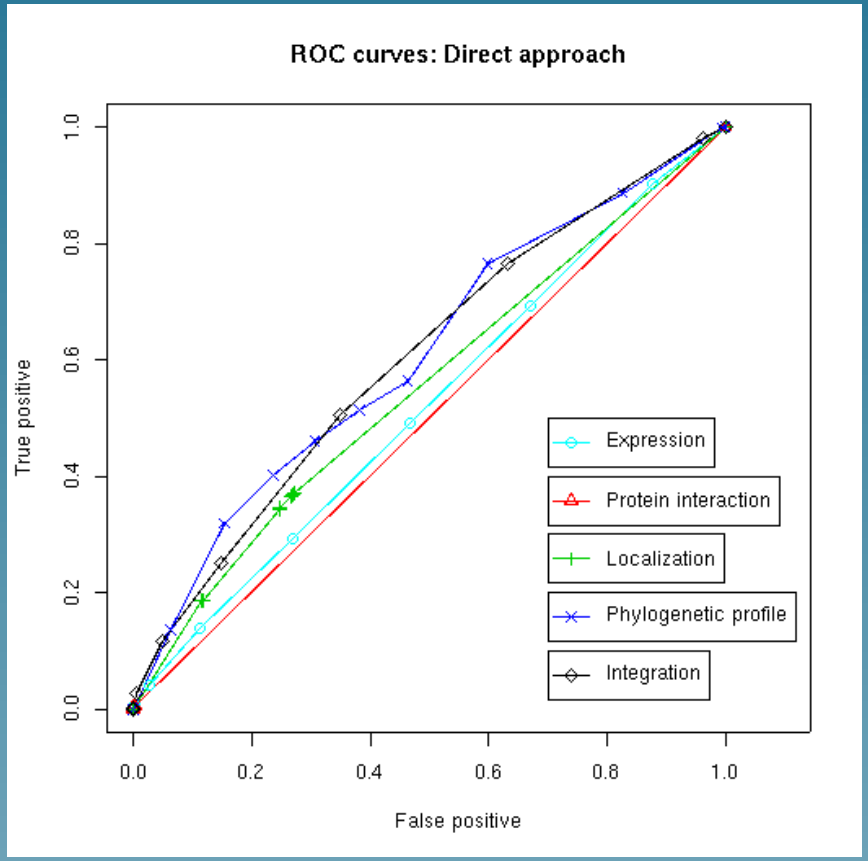


Learning from heterogeneous data

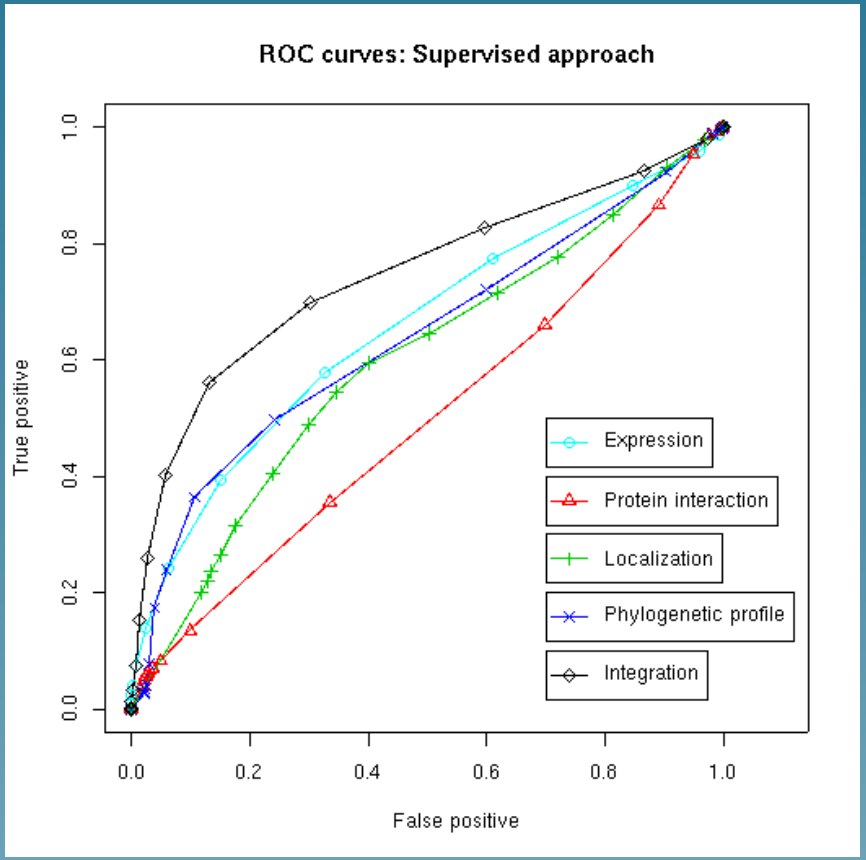
- Suppose several data are available about the genes, e.g., expression, localization, struture, predicted interaction etc...
- Each data can be represented by a positive definite similarity matrix $K_1, \ldots, K_p$ called kernels
- Kernel can be combined by various operations, e.g., addition:

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

Learning from heterogeneous data (unsupervised)



Learning from heterogeneous data (supervised)



Extensions

- The diffusion kernel can be replaced by another graph kernel
- Other formulations can lead to kernel CCA (NIPS 02, ISMB 04)

Open questions / Ongoing work

- What should be the number of features (problem of embedding a graph in low dimension)
- Other cost functions
- How to better integrate several similarities? (semi-definite programming?)

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
- Possible connections with other algorithms (SVM, kernel CCA..)