

# Kernel methods in computational biology: Three examples

Jean-Philippe.Vert@mines.org

Ecole des Mines de Paris  
Computational Biology group

UC Berkeley, May 9, 2003.

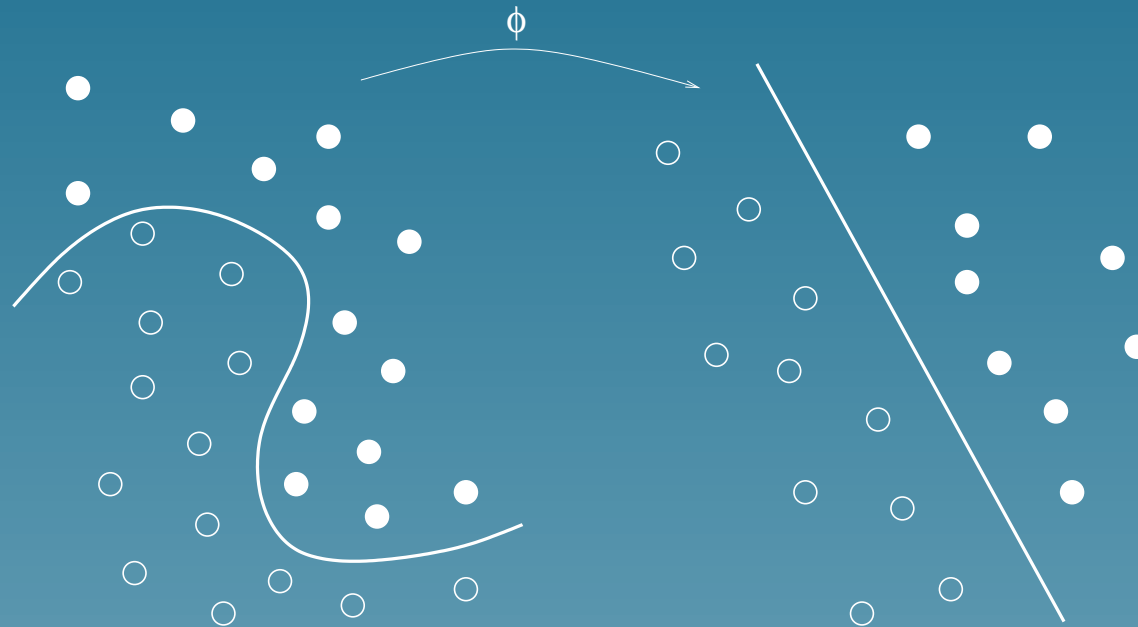
# Outline

1. SVM and kernel methods
2. Gene function prediction from phylogenetic profiles
3. Remote protein homology detection
4. Detection of active metabolic pathways from gene expression data

## Part 1

# SVM and kernel methods

# Support vector machines



- Objects to classify  $x$  mapped to a feature space
- Largest margin separating hyperplan in the feature space

# The kernel trick

- Implicit definition of  $x \rightarrow \Phi(x)$  through the kernel:

$$K(x, y) \stackrel{def}{=} \langle \Phi(x), \Phi(y) \rangle$$

# The kernel trick

- Implicit definition of  $x \rightarrow \Phi(x)$  through the kernel:

$$K(x, y) \stackrel{def}{=} \langle \Phi(x), \Phi(y) \rangle$$

- Simple kernels can represent complex  $\Phi$

# The kernel trick

- Implicit definition of  $x \rightarrow \Phi(x)$  through the kernel:

$$K(x, y) \stackrel{def}{=} \langle \Phi(x), \Phi(y) \rangle$$

- Simple kernels can represent complex  $\Phi$
- For a given kernel, not only SVM but also clustering, PCA, ICA... possible in the feature space = **kernel methods**

## Part 2

# Gene function prediction from phylogenetic profiles

*(ISMB 02)*



## Mini introduction

- **Genes** are small parts of the DNA which encode proteins.
- About 6,000 genes in the baker yeast, 30,000 in human
- The **sequences** of the genes are (almost) known (sequencing projects)
- Next big challenge: understand their **functions**

## Phylogenetic profile

- The phylogenetic profile of a gene is a vector of bits which indicates the presence (1) or absence (0) of the gene in every fully sequenced genome.

Gene	human	yeast	...	HIV	E. coli
YAL001C	1	1	...	0	0
YAB002W	0	0	...	0	1
⋮	⋮	⋮	⋮	⋮	⋮

- Can be estimated *in silico* by sequence similarity search

## From profile to function

- Genes are likely to be transmitted together during evolution when they participate:
  - ★ to a common structural complex,
  - ★ to a common pathway.
- Consequently genes with **similar phylogenetic** profiles are likely to have **similar functions**
- **How to infer the function from the profile?**

## Naive approach

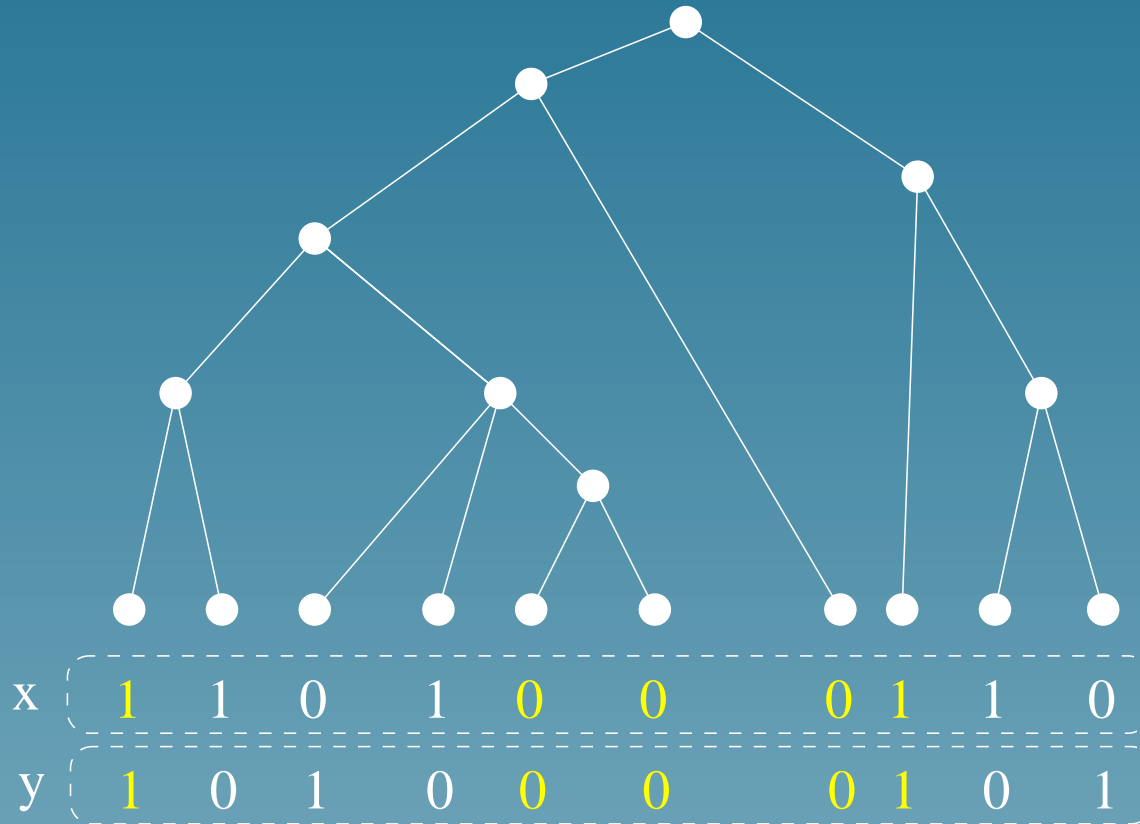
- Count the number of bits in common:

x	1	1	0	1	0	0	0	1	1	0
y	1	0	1	0	0	0	0	1	0	1

$$s(x, y) = 5$$

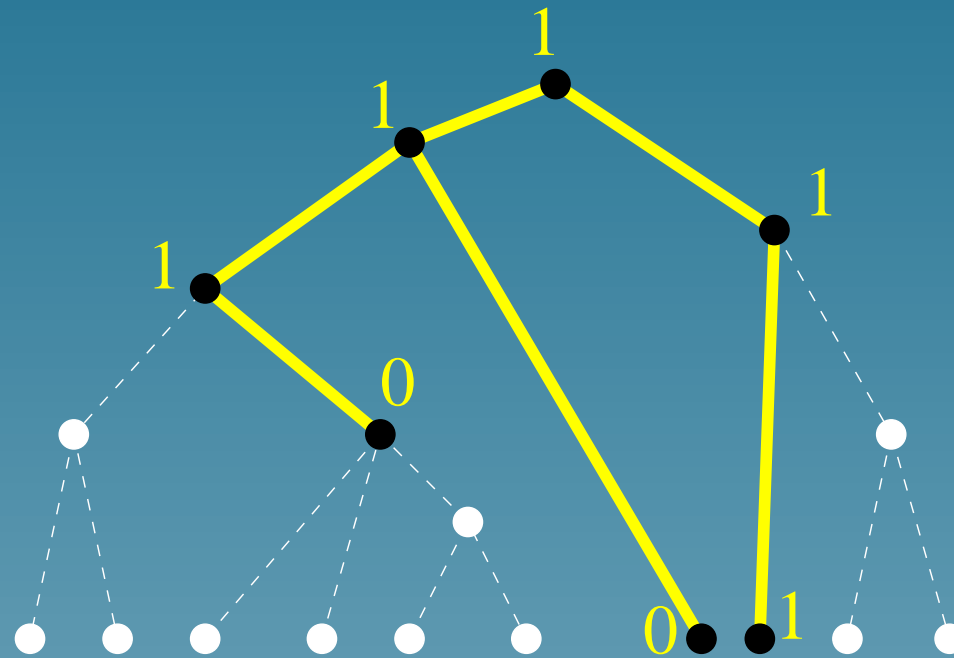
- Cluster or use k-NN for gene function prediction with this similarity measure (Pellegrini et al., 1999)

# What is not used in the naive approach



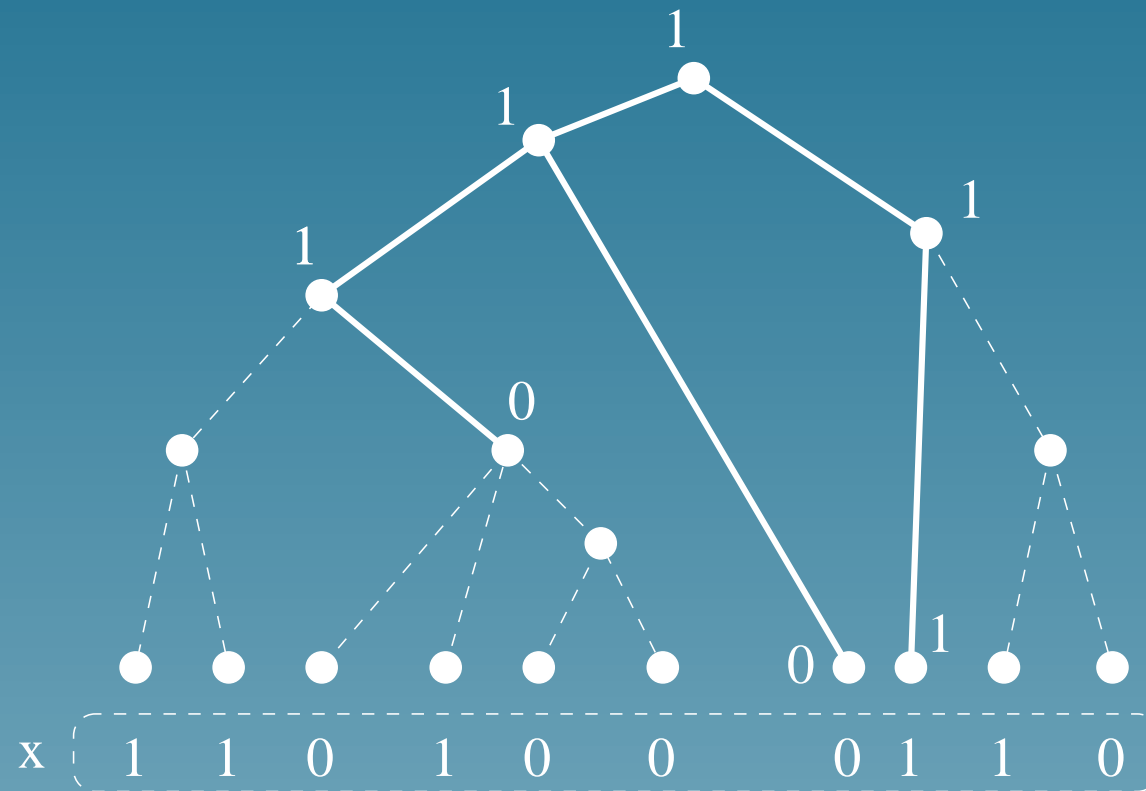
The knowledge of the *phylogenetic tree*.

# Evolution pattern



A possible **pattern of transmission** during evolution defined by a rooted subtree with nodes labeled 0 or 1.

# Evolution patterns and phylogenetic profiles



Is it the true story? **We don't know, but...**

# Probabilistic model of gene transmission

- The phylogenetic tree as a **tree graphical model**
- Simplified model:
  - ★  $P(1) = 1 - P(0) = 0.9$ , at the root,
  - ★ Along each branch transmission follows the transition matrix:

$$\begin{pmatrix} 0.9 & 0.1 \\ 0.1 & 0.9 \end{pmatrix}$$



# Probabilistic assignment of evolution pattern

For a phylogenetic profile  $x$  and an evolution pattern  $e$ :

- $P(e)$  quantifies how “natural” the pattern is
- $P(x|e)$  quantifies how likely the pattern  $e$  is the “true history” of the profile  $x$

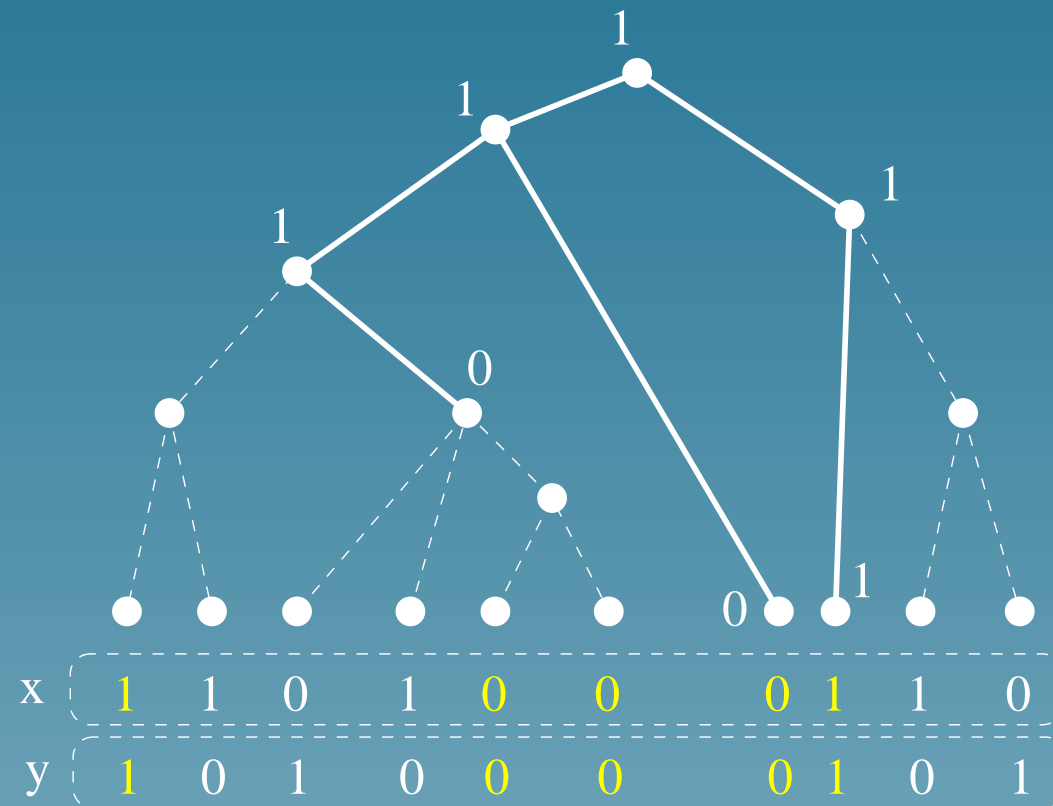
## Representation of a profile in terms of evolution patterns

- Consider **all possible evolution patterns**  $(e_1, \dots, e_N)$ , and represent each gene  $x$  by the vector:

$$\Phi(x) = \begin{pmatrix} \sqrt{P(e_1)}P(x|e_1) \\ \vdots \\ \sqrt{P(e_N)}P(x|e_N) \end{pmatrix}$$

- Comparing  $\Phi(x)$  and  $\Phi(y)$  gives a precise idea of **which evolution patterns are shared or not by  $x$  and  $y$ .**

# Comparing two profiles through evolution patterns



# Tree kernel

- Kernel methods (SVM, kernel-PCA, kernel-clustering...) only require the computation of the **kernel function**:

$$K(x, y) = \Phi(x) \cdot \Phi(y).$$

- In our case we obtain the **tree kernel**:

$$K(x, y) = \sum_e P(e)P(x|e)P(y|e),$$

where the sum is over **all possible evolution patterns**.

## Kernel computation: trick 1

- For any given pattern  $e$ , the term:

$$\alpha(e) = P(e)P(x|e)P(y|e)$$

can be factorized and computed **recursively** by working up the tree from the leaves

- Classical trick for computing likelihood with tree graphical models, cf. Felsenstein's algorithm

## Kernel computation: trick 2

- The sum

$$\sum_e \alpha(e)$$

over **all subtrees** can also be factorized and computed recursively by working up the tree from the leaves

- Similar in spirit to the Context Tree Weighting algorithm (Willems et al., 1995).

## Combining tricks

- Both tricks can be combined
- $K(x, y)$  can be computed by two post-order traversals of the tree
- The complexity is linear with the length of the profile.

## Gene function prediction with SVM

- Profiles for 2465 genes of *S. Cerevisiae* were computed by BLAST search (cf Pavlidis et al. 2001), using 24 genomes.
- Consensus phylogenetic tree (cf. Liberles et al. 2002) with simplified probabilistic model of gene transmission
- SVM trained to predict all functional classes of the MIPS catalog with at least 10 genes (cross-validation)
- Comparison of the tree kernel with the naive kernel



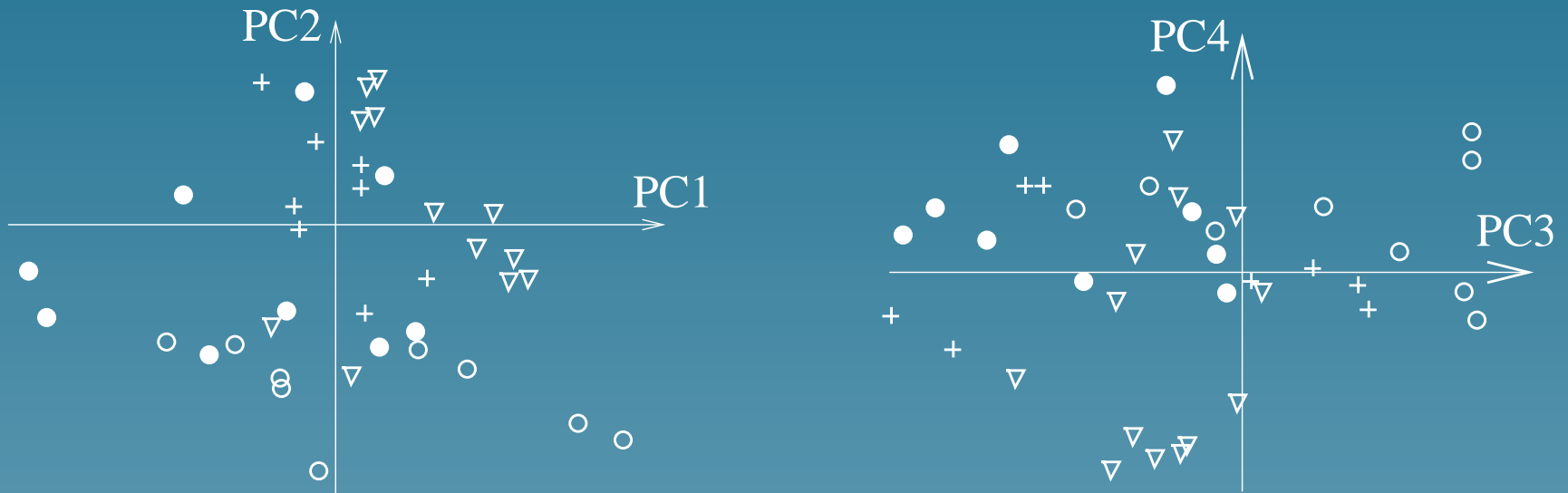
## Results (ROC 50)

Functional class	Naive kernel	Tree kernel	Difference
Amino-acid transporters	0.74	0.81	+ 9%
Fermentation	0.68	0.73	+ 7%
ABC transporters	0.64	0.87	+ 36%
C-compound transport	0.59	0.68	+ 15%
Amino-acid biosynthesis	0.37	0.46	+ 24%
Amino-acid metabolism	0.35	0.32	- 9%
Tricarboxylic-acid pathway	0.33	0.48	+ 45%
Transport Facilitation	0.33	0.28	- 15%

## A insight into the feature space

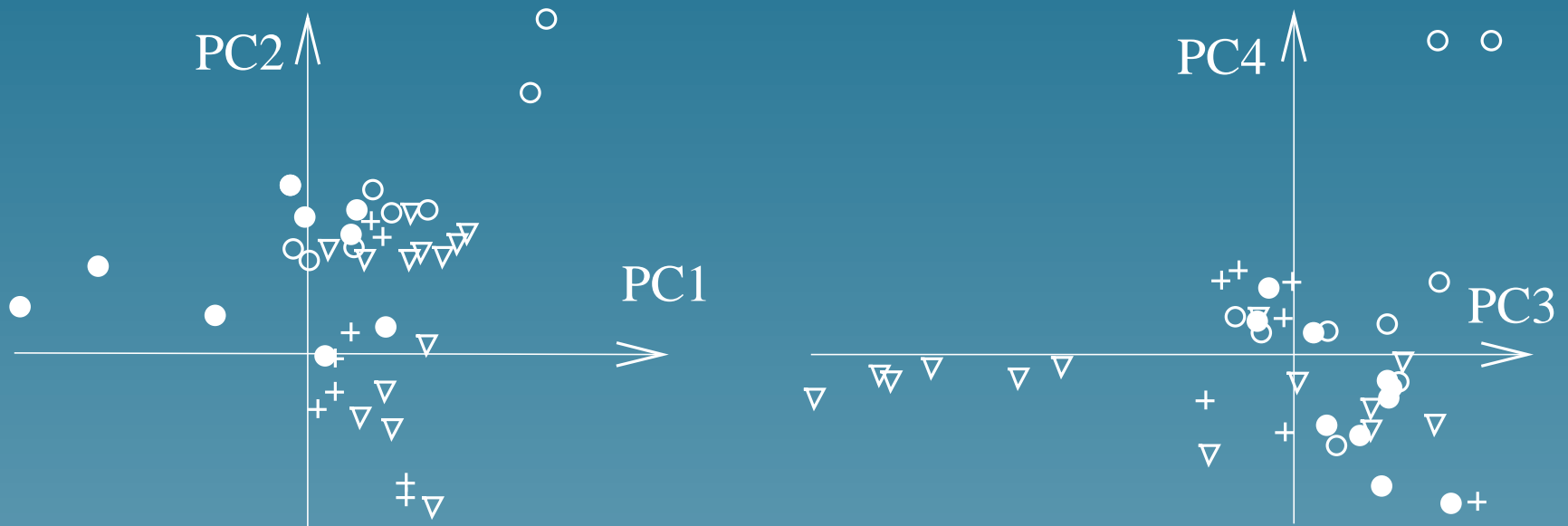
- PCA can be performed implicitly in the feature space with a kernel function: **kernel-PCA** (Scholkopf et al. 1999)
- Projecting the genes on the first principal components gives an idea of the shape of the features space

# Naive kernel PCA



- Amino-acid transporters
- Fermentation
- ▽ ABC transporters
- + C-compound, carbohydrate transport

# Tree kernel PCA



- Amino-acid transporters
- Fermentation
- ▽ ABC transporters
- + C-compound, carbohydrate transport

# Extensions

- $X_1, \dots, X_n$  discrete r.v.
- $I_1, \dots, I_v \subset \{1, \dots, n\}$  a family of subsets
- Interpolated kernel:

$$K(x, y) = \frac{1}{v} \sum_{i=1}^v p(x_{I_i}) p(y_{I_i}) \times p(x_{I_i^c}) \delta(x_{I_i^c}, y_{I_i^c})$$

# Property 1

This kernel **interpolates** between the **diagonal kernel**:

$$K_{diag}(x, y) = p(x)\delta(x, y)$$

and the **product kernel**:

$$K_{prod}(x, y) = p(x)p(y).$$

## Property 2

Two objects  $x$  and  $y$  get closer in the feature space when they share **rare common subparts**:

$$K(x, y) = K_{prod}(x, y) \times \frac{1}{v} \sum_{i=1}^v \frac{\delta(x_{I_i}, y_{I_i})}{p(x_{I_i})}$$

# Linear-time implementations

- iid r.v., all possible subsets (*PSB 02*):

$X_1$

$X_2$

$X_3$

$X_4$

$X_5$

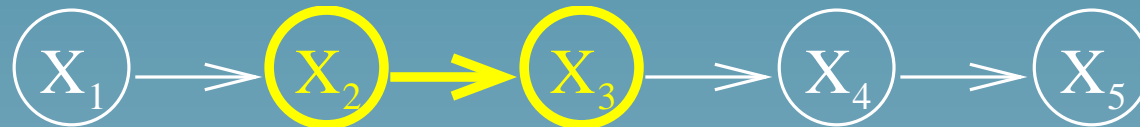


# Linear-time implementations

- iid r.v., all possible subsets (*PSB 02*):

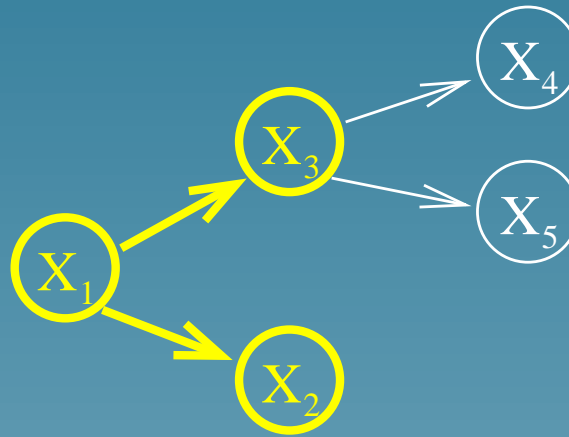


- Markov model, common blocks



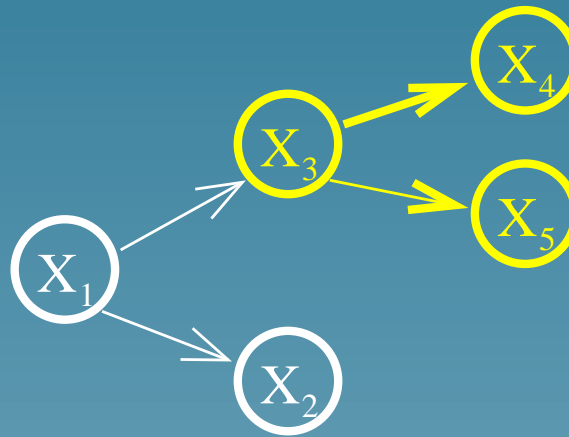
# Linear-time implementations

- Tree graphical model, common rooted subtrees



# Linear-time implementations

- Tree graphical model, common subtrees



## Part 2

# Remote protein homology detection

(with S. Hiroto, N. Ueda, T. Akutsu, preprint 2003)

# Motivations

- Develop a **kernel for strings** adapted to protein / DNA sequences
- Several methods have been adopted in bioinformatics to measure the similarity between sequences... but are not valid kernels
- How to mimic them?

## Local alignment

- For two strings  $x$  and  $y$ , a local alignment  $\pi$  with gaps is:

```

ABCD EF---G-HI JKL
      ||         ||  |
MNO  EFPORGS-I TUVWX
  
```

- The score is:

$$s(x, y, \pi) = s(E, E) + s(F, F) + s(G, G) + s(I, I) - s(\text{gaps})$$

## Smith-Waterman (SW) score

$$SW(x, y) = \max_{\pi \in \Pi(x, y)} s(x, y, \pi)$$

- Computed by dynamic programming
- Not a kernel in general

## Convolution kernels (Haussler 99)

- Let  $K_1$  and  $K_2$  be two kernels for strings
- Their **convolution** is the following valid kernel:

$$K_1 \star K_2(x, y) = \sum_{x_1 x_2 = x, y_1 y_2 = y} K_1(x_1, y_1) K_2(x_2, y_2)$$



## 3 basic kernels

- For the unaligned parts:  $K_0(x, y) = 1$ .

## 3 basic kernels

- For the unaligned parts:  $K_0(x, y) = 1$ .
- For aligned residues:

$$K_a^{(\beta)}(x, y) = \begin{cases} 0 & \text{if } |x| \neq 1 \text{ or } |y| \neq 1, \\ \exp(\beta s(x, y)) & \text{otherwise} \end{cases}$$

## 3 basic kernels

- For the unaligned parts:  $K_0(x, y) = 1$ .
- For aligned residues:

$$K_a^{(\beta)}(x, y) = \begin{cases} 0 & \text{if } |x| \neq 1 \text{ or } |y| \neq 1, \\ \exp(\beta s(x, y)) & \text{otherwise} \end{cases}$$

- For gaps:

$$K_g^{(\beta)}(x, y) = \exp[\beta (g(|x|) + g(|y|))]$$

## Combining the kernels

- Detecting local alignments of exactly  $n$  residues:

$$K_{(n)}^{(\beta)}(x, y) = K_0 \star \left( K_a^{(\beta)} \star K_g^{(\beta)} \right)^{(n-1)} \star K_a^{(\beta)} \star K_0.$$

## Combining the kernels

- Detecting local alignments of exactly  $n$  residues:

$$K_{(n)}^{(\beta)}(x, y) = K_0 \star \left( K_a^{(\beta)} \star K_g^{(\beta)} \right)^{(n-1)} \star K_a^{(\beta)} \star K_0.$$

- Considering all possible local alignments:

$$K_{LA}^{(\beta)} = \sum_{i=0}^{\infty} K_{(i)}^{(\beta)}.$$

# Properties

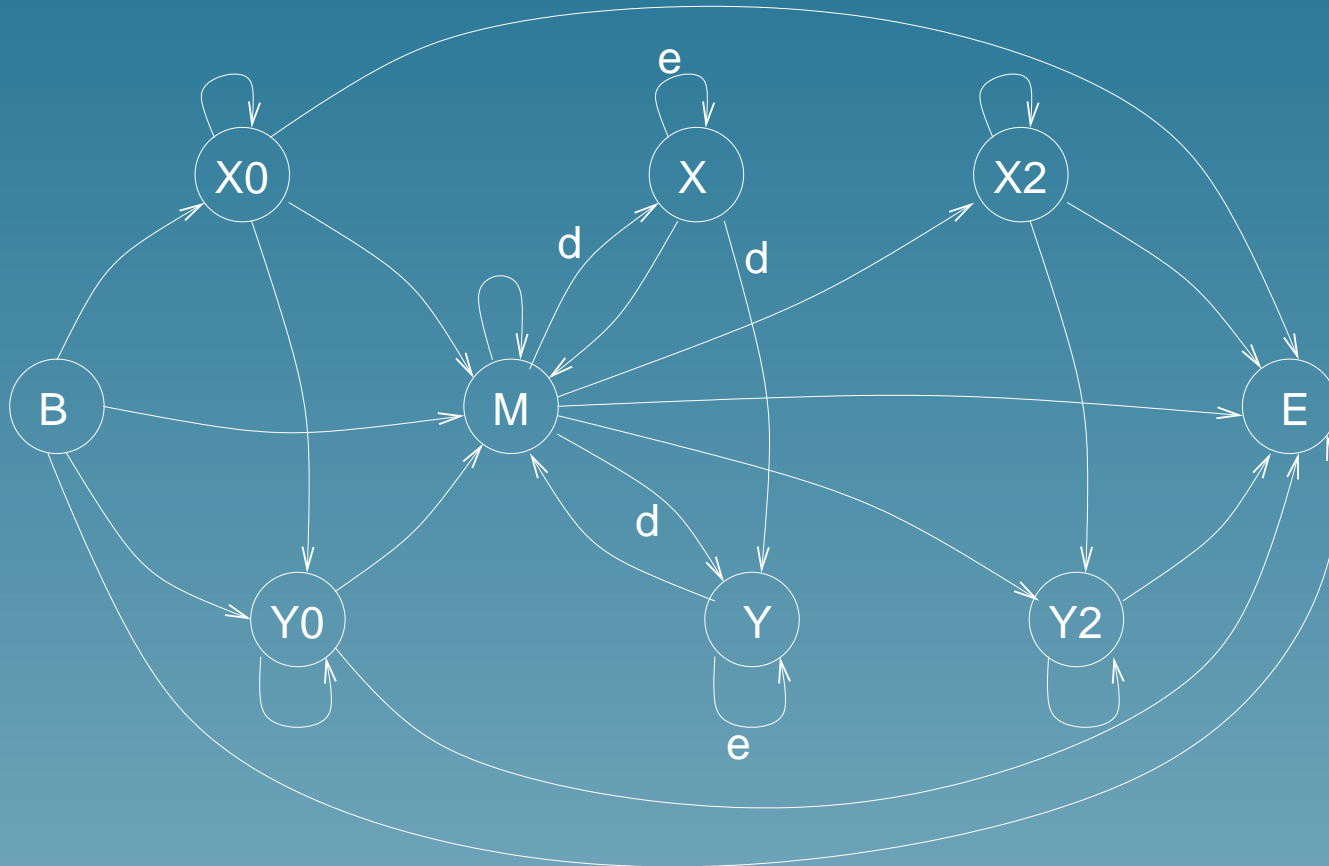
$$K_{LA}^{(\beta)}(x, y) = \sum_{\pi \in \Pi(x, y)} \exp(\beta s(x, y, \pi)),$$

# Properties

$$K_{LA}^{(\beta)}(x, y) = \sum_{\pi \in \Pi(x, y)} \exp(\beta s(x, y, \pi)),$$

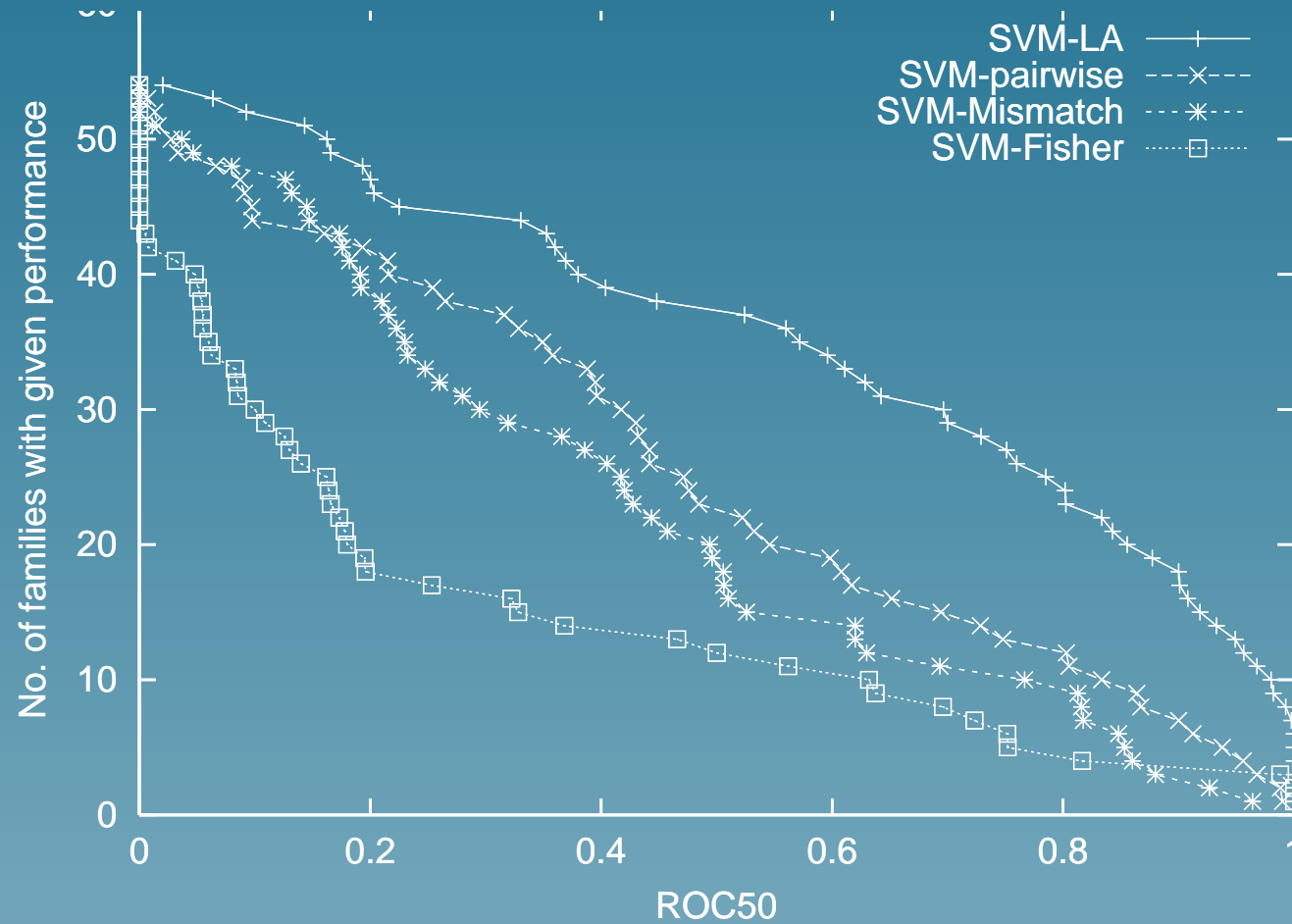
$$\lim_{\beta \rightarrow +\infty} \frac{1}{\beta} \ln K_{LA}^{(\beta)}(x, y) = SW(x, y).$$

# Kernel computation





# SCOP superfamily recognition benchmark

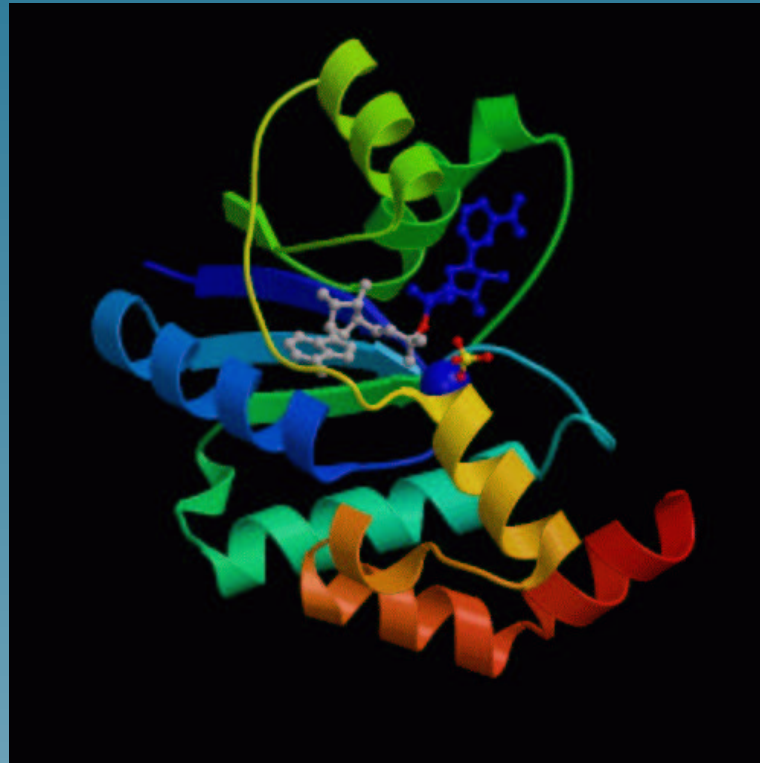


## Part 3

# Detection of active metabolic pathways from gene expression data

*(NIPS 02)*

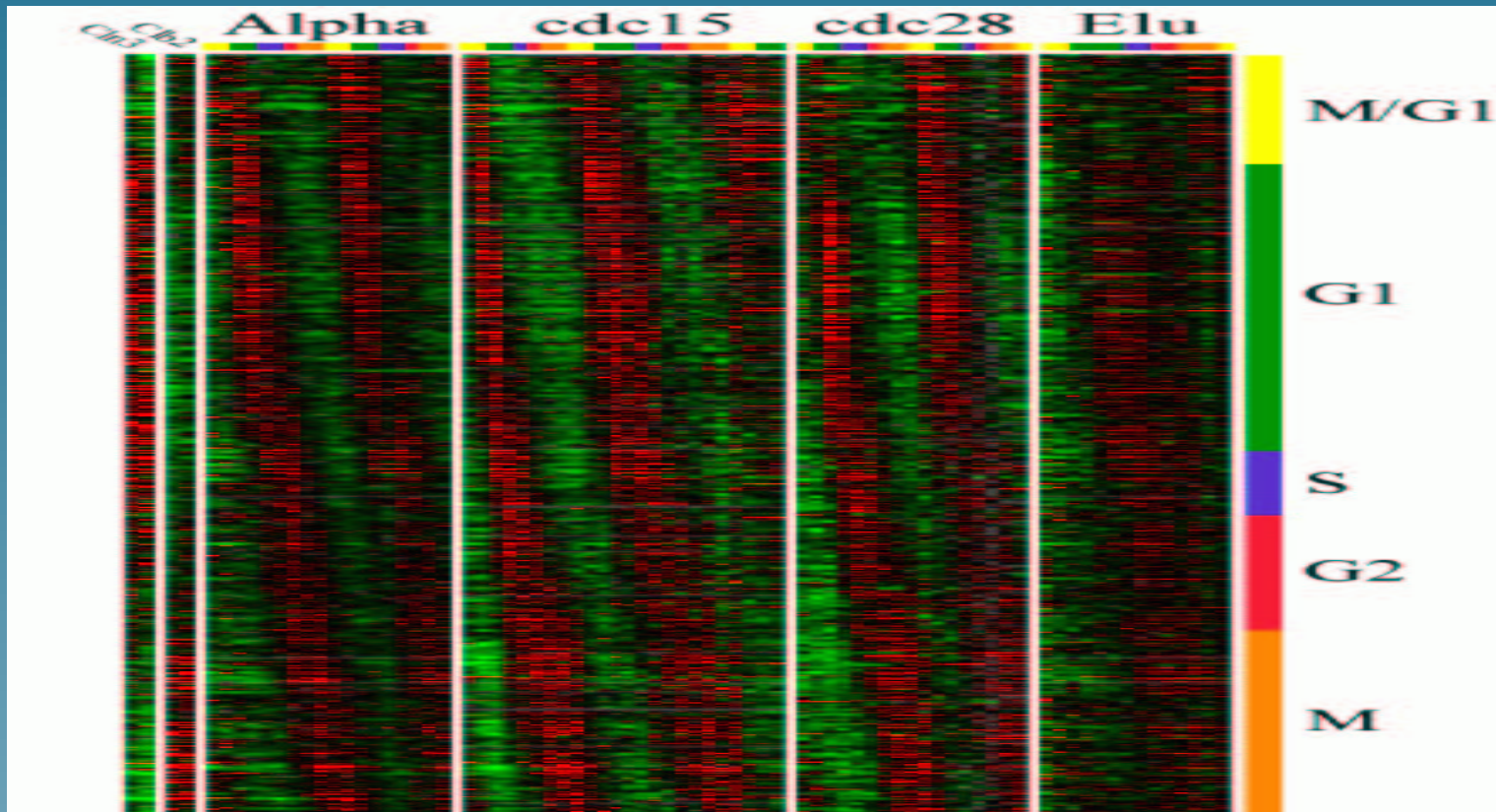
# Genes encode proteins which can catalyse chemical reactions



Nicotinamide Mononucleotide Adenylyltransferase With Bound Nad<sup>+</sup>

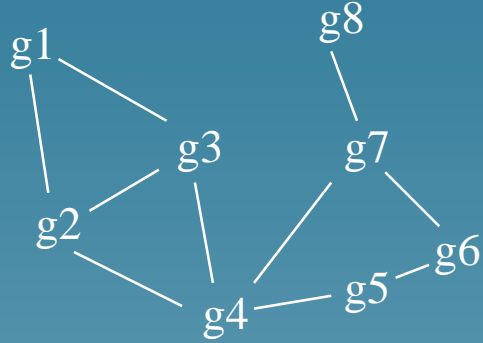


# Microarray technology monitors RNA quantity



(From Spellman et al., 1998)

# Comparing gene expression and protein network



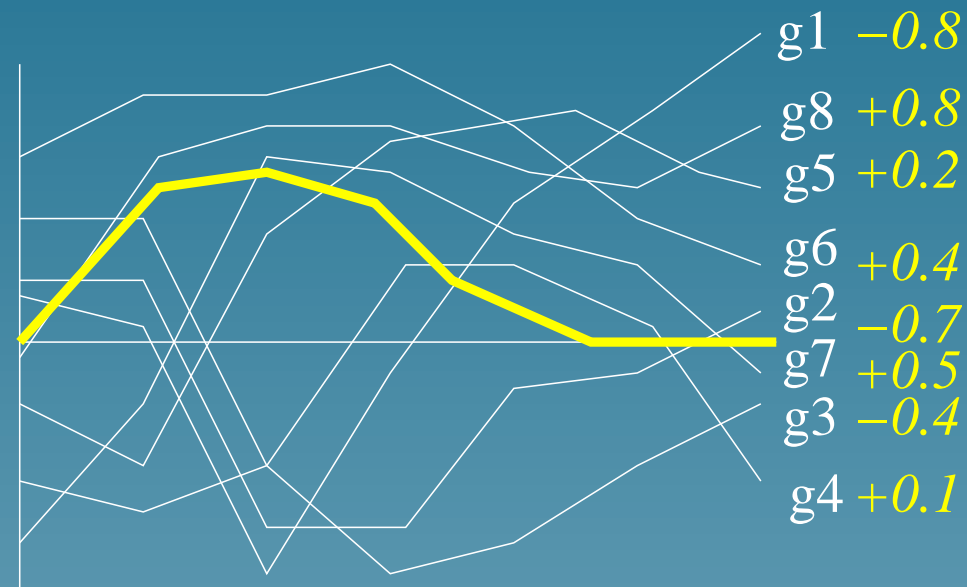
Gene network



Expression profiles

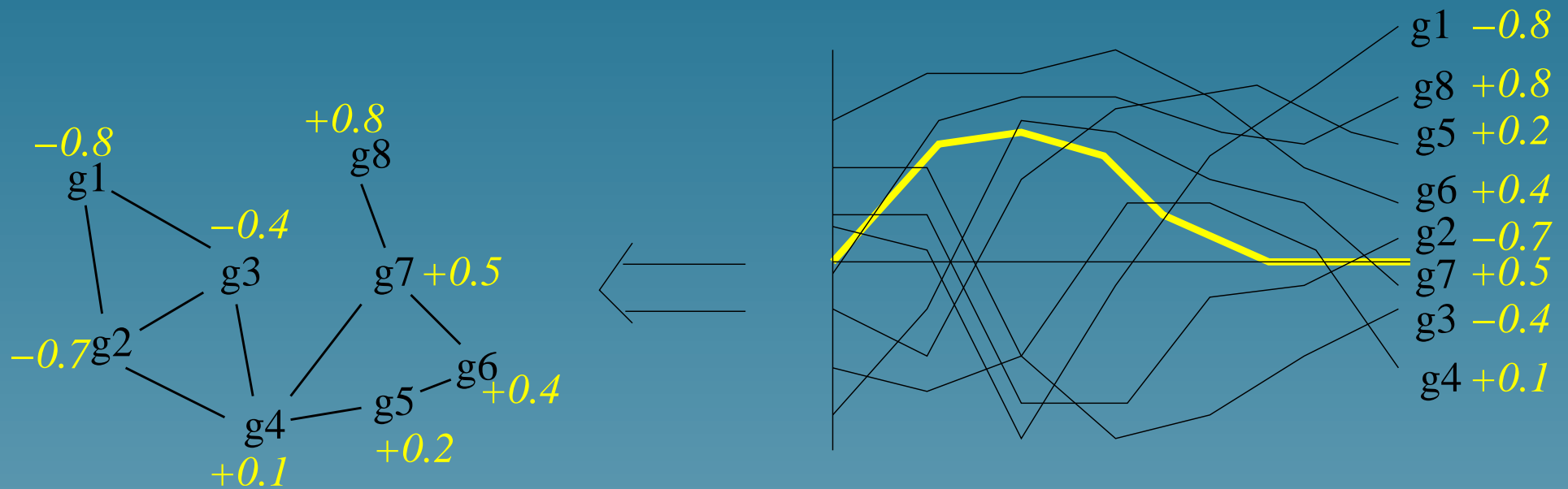
Are there “correlations”?

## Pattern of expression



- In yellow: a candidate **pattern** , and the **correlation coefficient** with each gene profile

# Pattern smoothness



- The correlation function with **interesting patterns** should vary **smoothly** on the graph



# Pattern relevance

- Interesting patterns involve many genes
- The projection of profiles onto an interesting pattern should capture a lot of variations among profiles
- Relevant patterns can be found by PCA

# Problem

Find patterns of expression which are **simultaneously**

- smooth
- relevant

## From kernels to RKHS

- To each kernel  $K$  is associated a **reproducing kernel Hilbert space (RKHS)**, subset of  $\mathbb{R}^{\mathcal{X}}$ , defined as the **completion** of:

$$\text{span} \{K(x, \cdot), x \in \mathcal{X}\}.$$

- The norm of a function  $f = \sum_i a_i K(x_i, \cdot)$  in the RKHS is:

$$\|f\|_{\mathcal{H}} = \sum_{i,j} a_i a_j K(x_i, x_j).$$

## Pattern relevance

- Let  $e(x)$  the profile of gene  $x$
- Let  $K_1(x, y) = e(x).e(y)$  be the **linear kernel**, with RKHS  $H_1$ .
- The norm  $\|\cdot\|_{H_1}$  is a **relevance functional**: the relevance of  $f \in H_1$  increases when the following decreases:

$$\frac{\|f\|_{H_1}}{\|f\|_{L_2}}$$

## Pattern smoothness

- Let  $K_2(x, y)$  be the **diffusion kernel** obtained from the gene network, with RKHS  $H_2$ .
- It can be considered as a discretized version of a Gaussian kernel (solving the heat equation with the graph Laplacian)
- The norm  $\|\cdot\|_{H_2}$  is a **smoothness functional**: the smoother a function  $f : \mathcal{X} \rightarrow \mathbb{R}$ , the larger the function:

$$\frac{\|f\|_{H_1}}{\|f\|_{L_2}}$$

## Problem reformulation

Find a linear function  $f_1$  and a function  $f_2$  such that:

- $f_1$  be relevant :  $\|f_1\|_{L^2}/\|f_1\|_{H_1}$  be large
- $f_2$  be smooth :  $\|f_2\|_{L^2}/\|f_2\|_{H_2}$  be large
- $f_1$  and  $f_2$  be correlated :

$$\frac{f_1 \cdot f_2}{\|f_1\|_{L^2} \|f_2\|_{L^2}}$$

be large

## Problem reformulation (2)

The three goals can be combined in the following problem:

$$\max_{f_1, f_2} \frac{f_1 \cdot f_2}{\left( \|f_1\|_{L^2}^2 + \delta \|f_1\|_{H_1}^2 \right)^{\frac{1}{2}} \left( \|f_2\|_{L^2}^2 + \delta \|f_2\|_{H_2}^2 \right)^{\frac{1}{2}}}$$

where the parameter  $\delta$  controls the trade-off between relevance/smoothness on the one hand, correlation on the other hand.

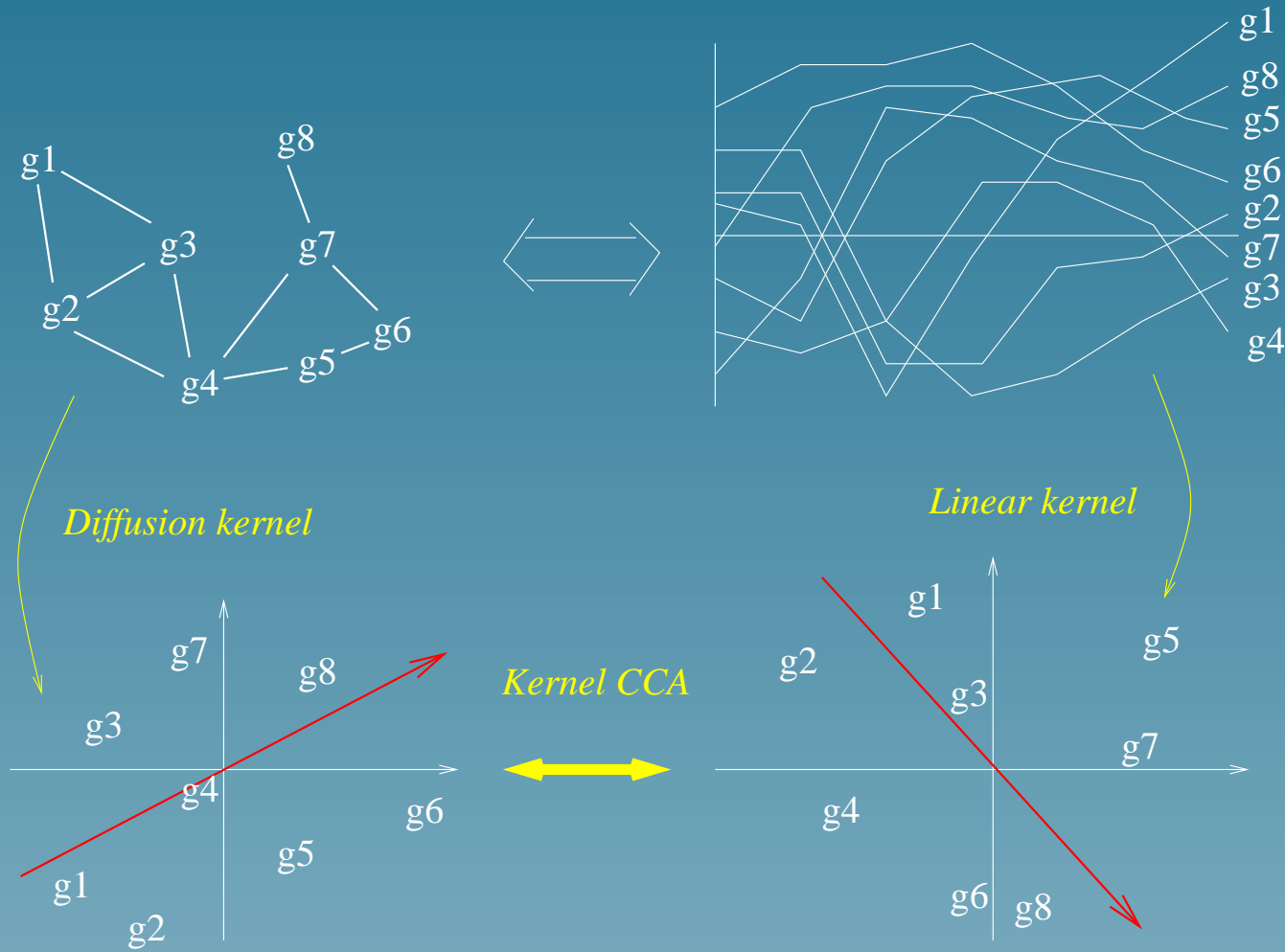
## Solving the problem

This formulation is equivalent to a generalized form of CCA (**Kernel-CCA**, Bach and Jordan, 2002), which is equivalent to the following generalized eigenvector problem

$$\begin{pmatrix} 0 & K_1 K_2 \\ K_2 K_1 & 0 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \rho \begin{pmatrix} K_1^2 + \delta K_1 & 0 \\ 0 & K_2^2 + \delta K_2 \end{pmatrix} \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$



# Summary



# Data

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

# First pattern of expression

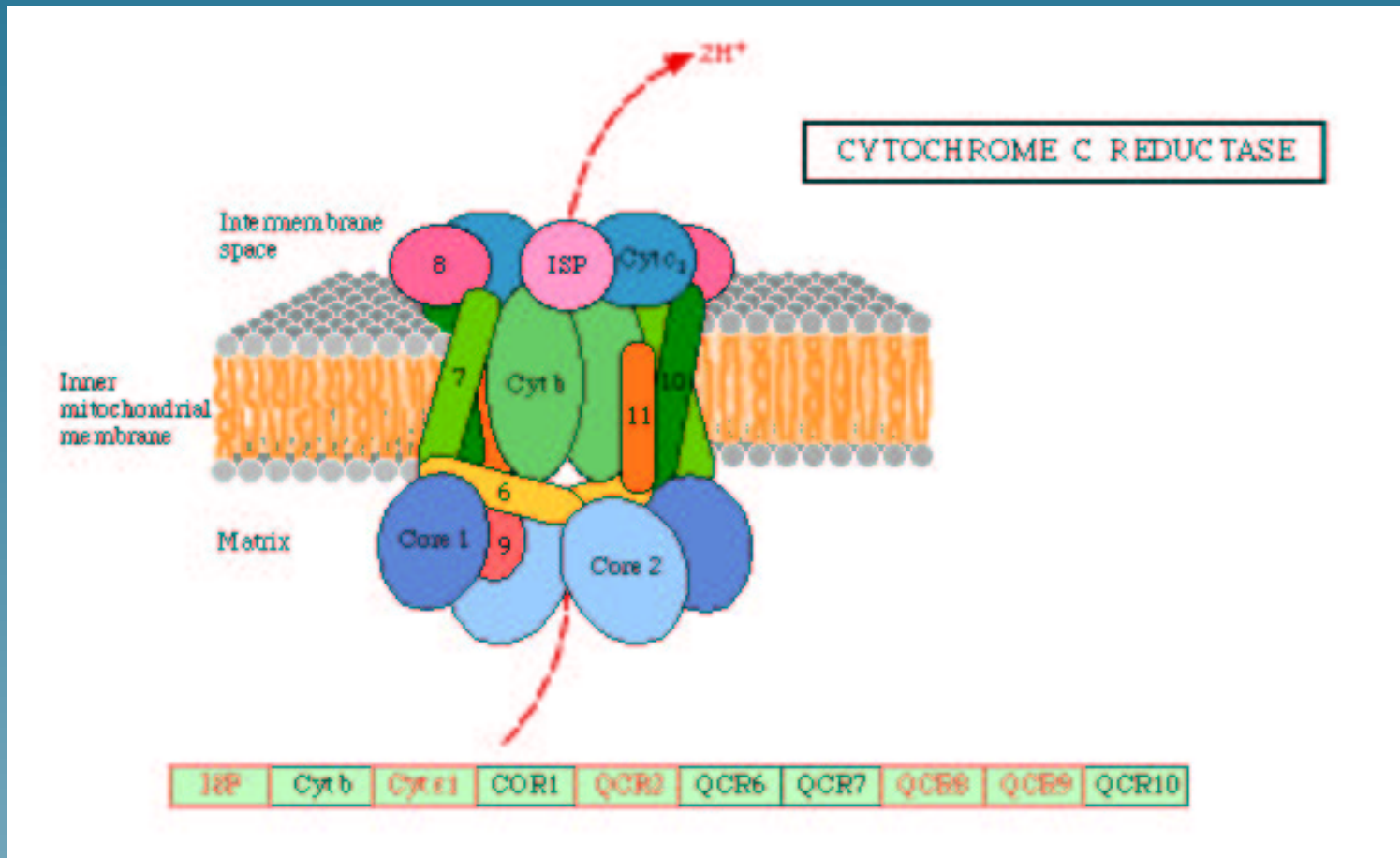


## Related metabolic pathways

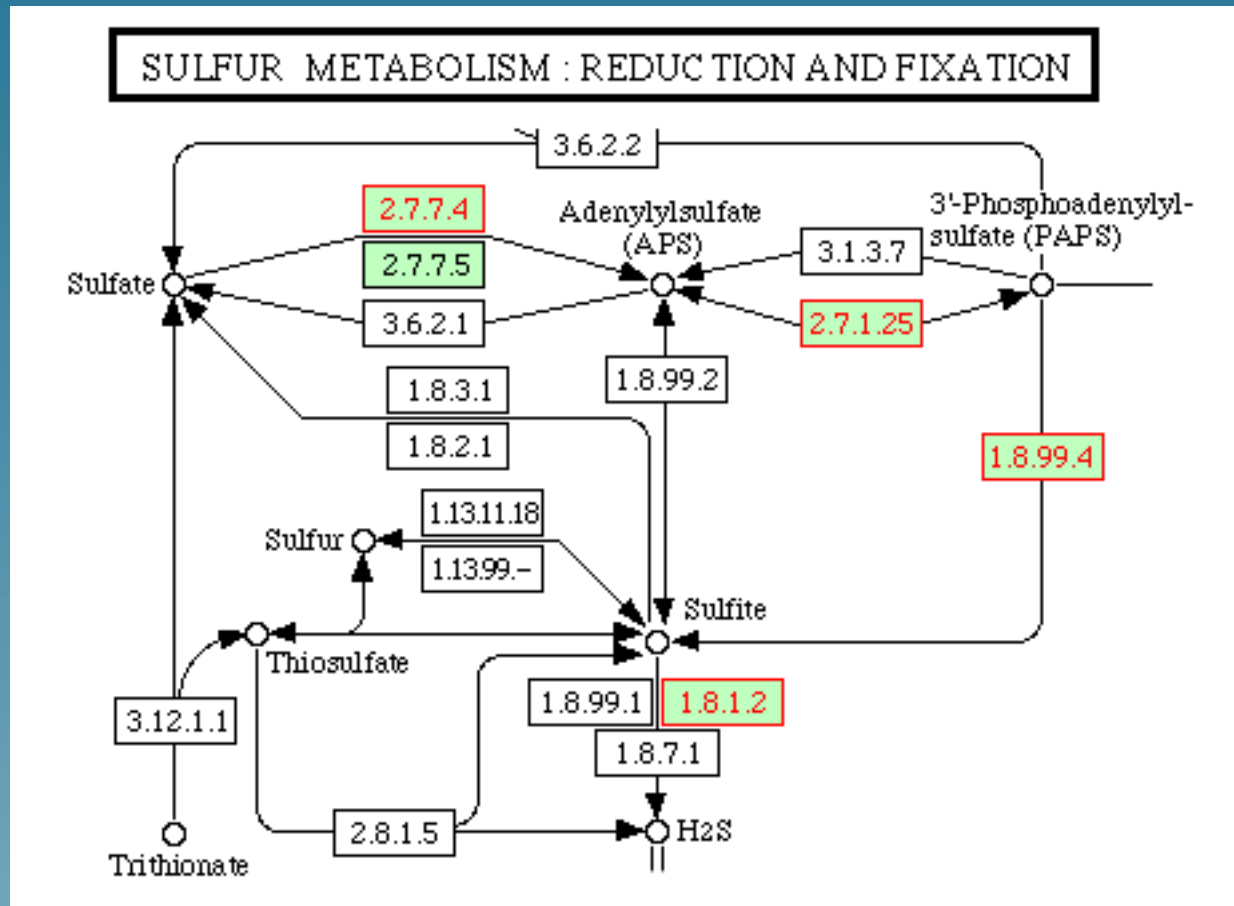
50 genes with highest  $s_2 - s_1$  belong to:

- Oxidative phosphorylation (10 genes)
- Citrate cycle (7)
- Purine metabolism (6)
- Glycerolipid metabolism (6)
- Sulfur metabolism (5)
- Selenoaminoacid metabolism (4) , etc...

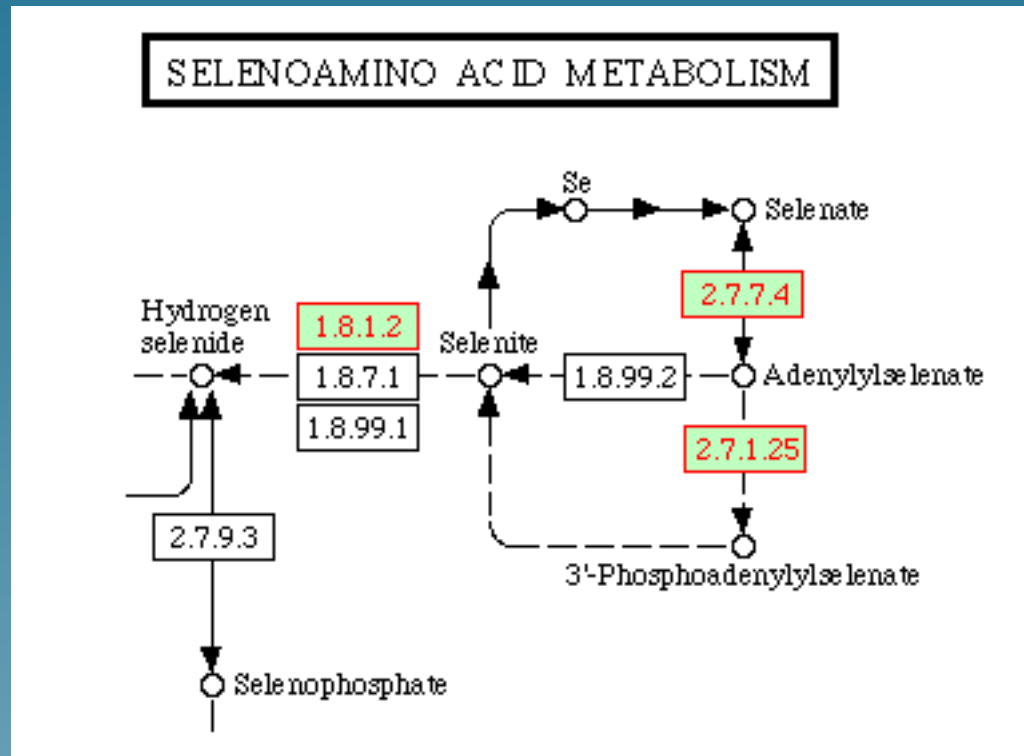
# Related genes



# Related genes



# Related genes



# Opposite pattern

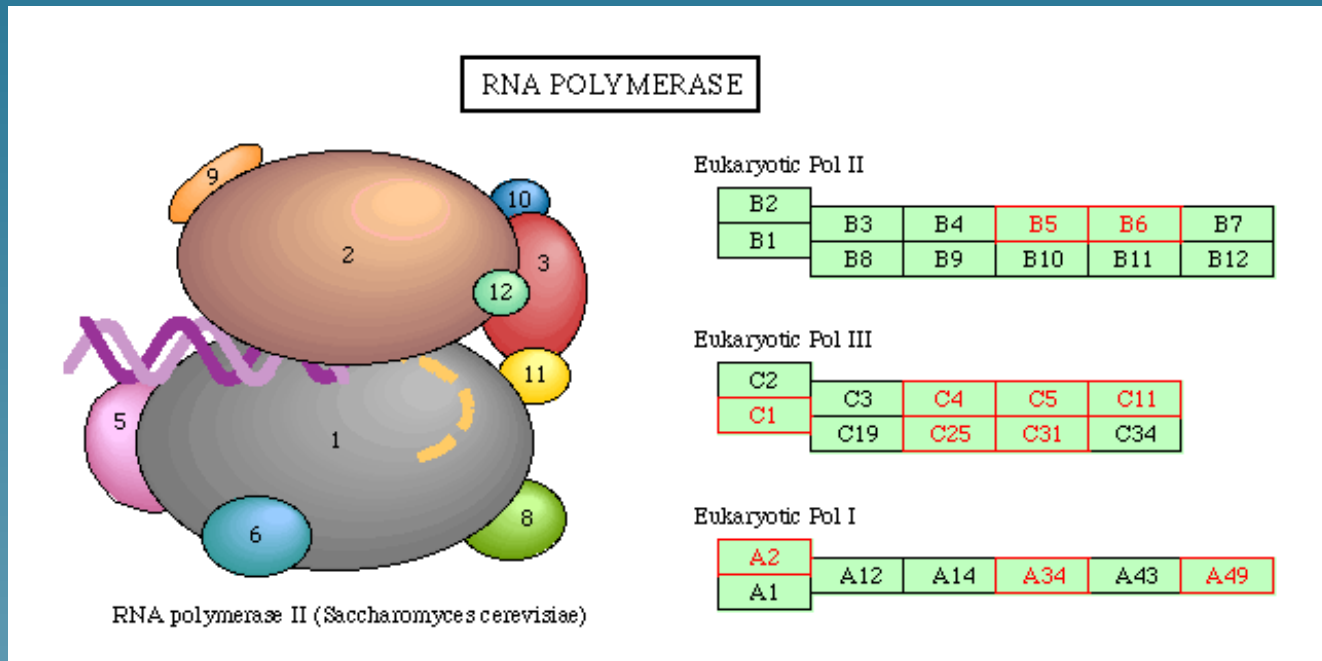




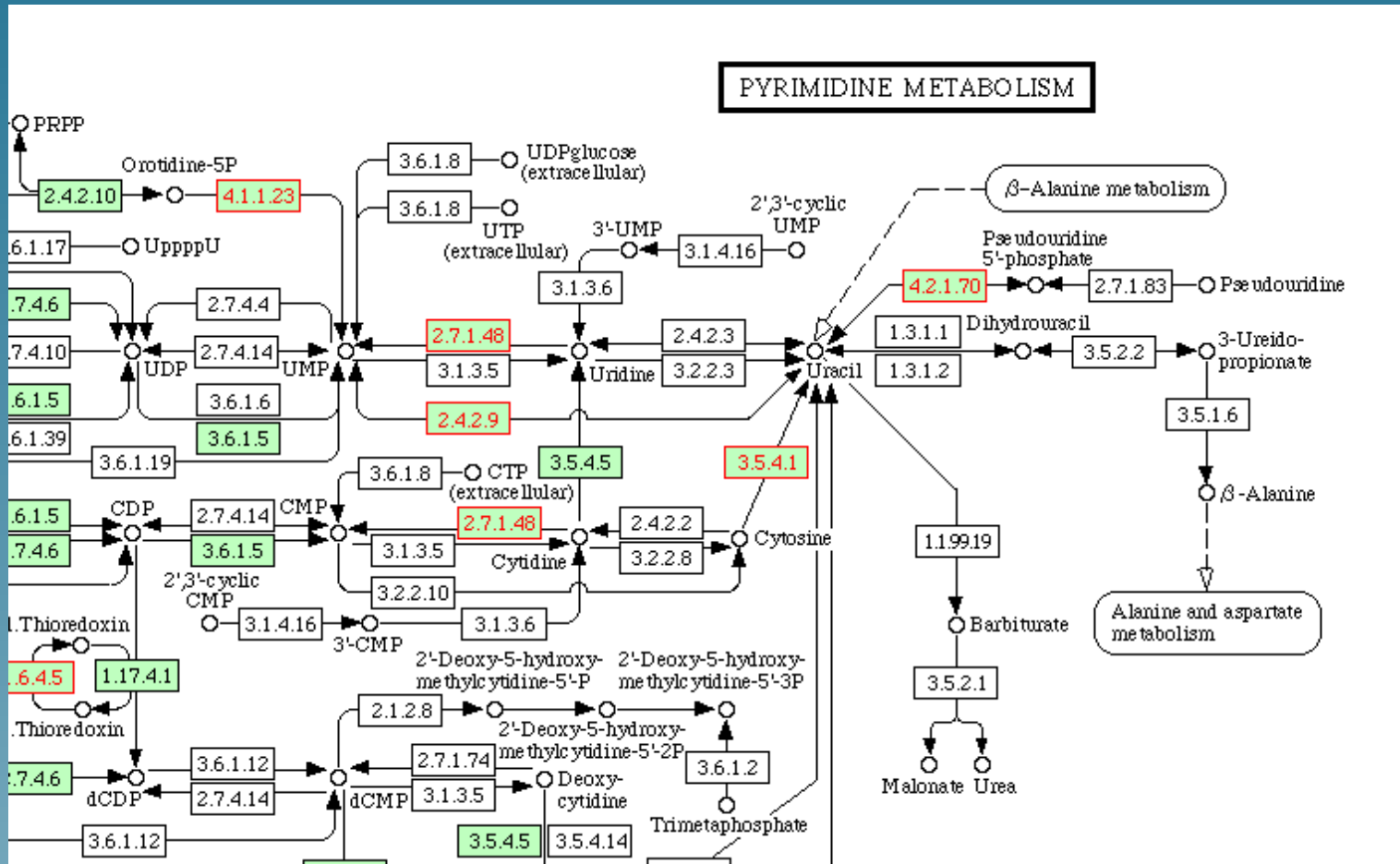
## Related genes

- 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...

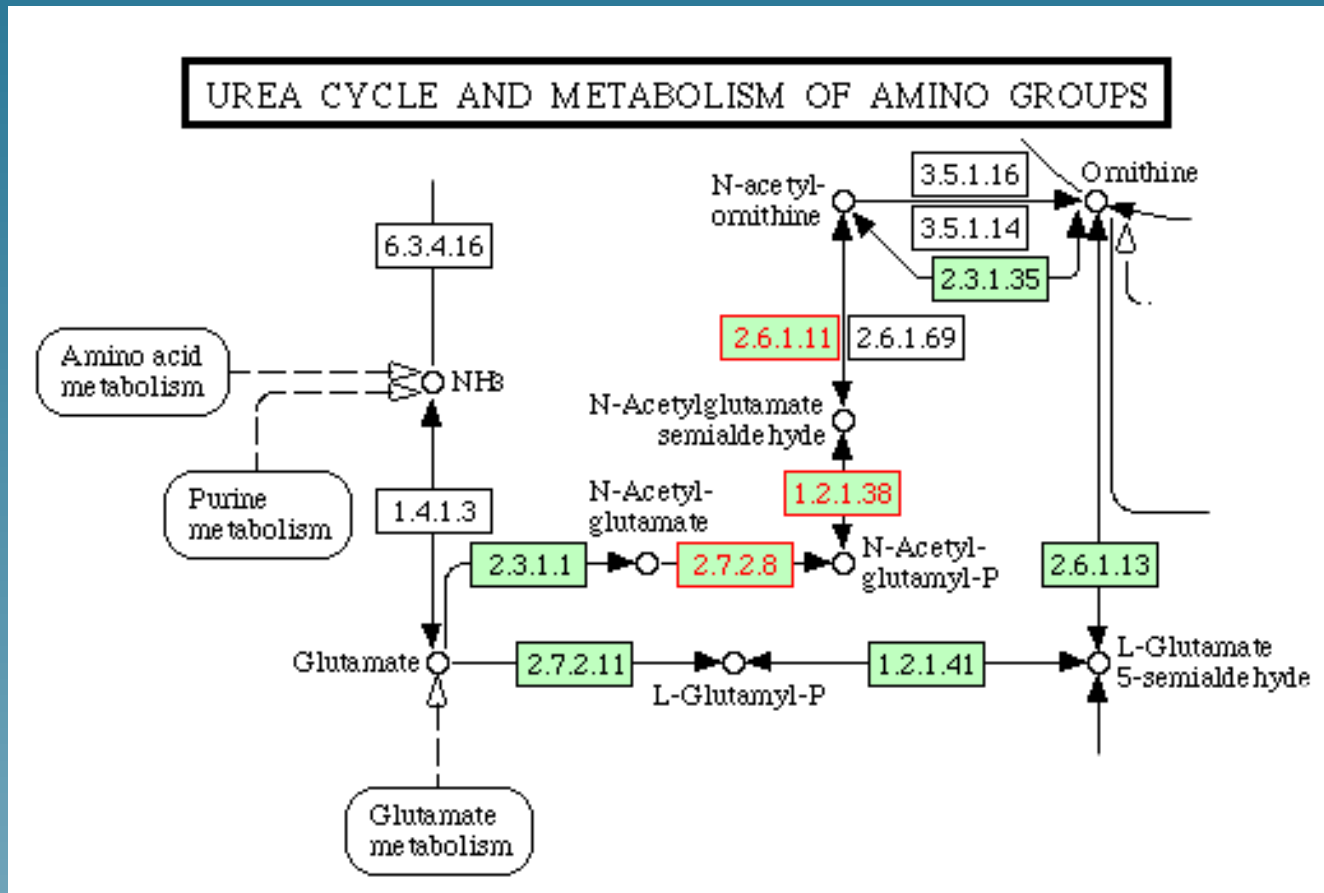
# Related genes



# Related genes



# Related genes



## Extensions

- Can be used to **extract features** from expression profiles (preprint 2002)
- Can be generalized to **more than 2 datasets** and other kernels
- Can be used to extract **clusters of genes** (e.g., operon detection, *ISMB 03* with Y. Yamanishi, A. Nakaya and M. Kanehisa)

# Conclusion

# Conclusion

- Kernels offer a versatile framework to represent biological data
- Increasing library of kernels and kernel methods
- Encouraging results on real-world applications
- Candidate to play an important role in learning from heterogeneous data