

Support vector machines, Kernel methods, and Applications in bioinformatics

Jean-Philippe.Vert@mines.org

Ecole des Mines de Paris
Computational Biology group

Machine Learning in Bioinformatics conference,
October 17th, 2003, Brussels, Belgium

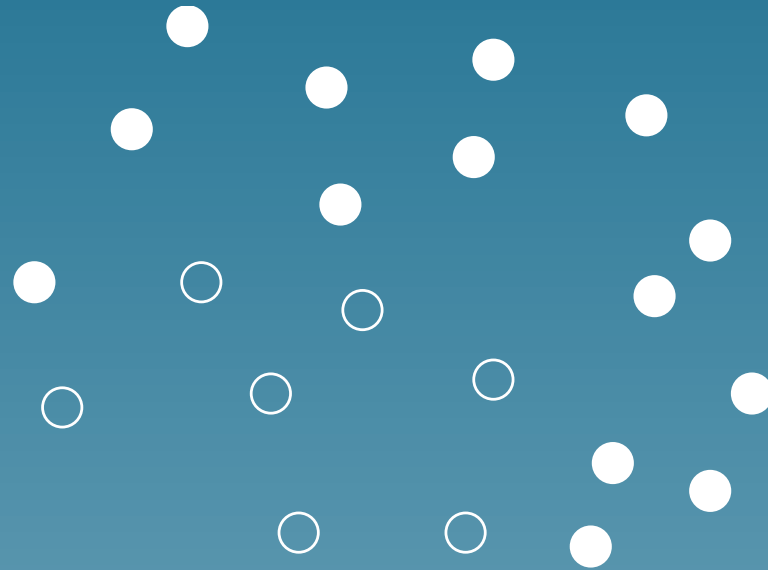
Overview

1. Support Vector Machines and kernel methods
2. Application: Protein remote homology detection
3. Application: Extracting pathway activity from gene expression data

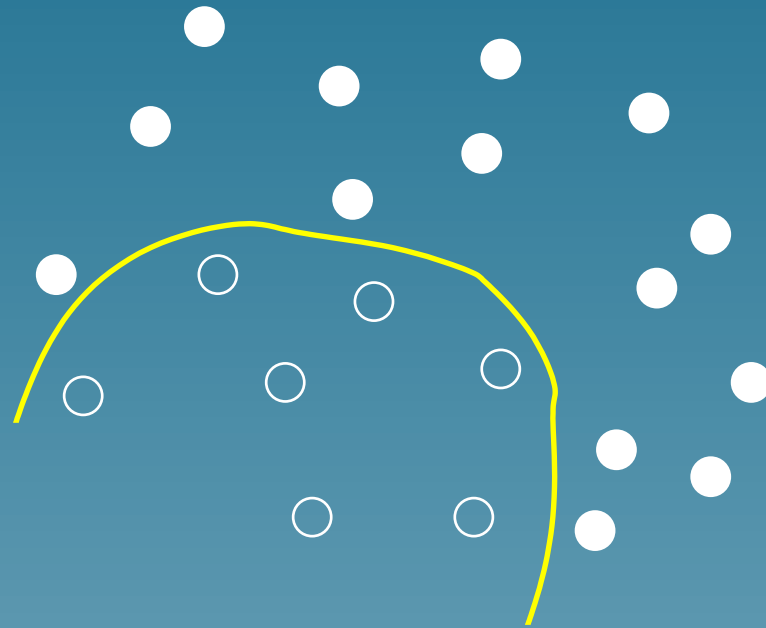
Partie 1

Support Vector Machines (SVM) and Kernel Methods

The pattern recognition problem

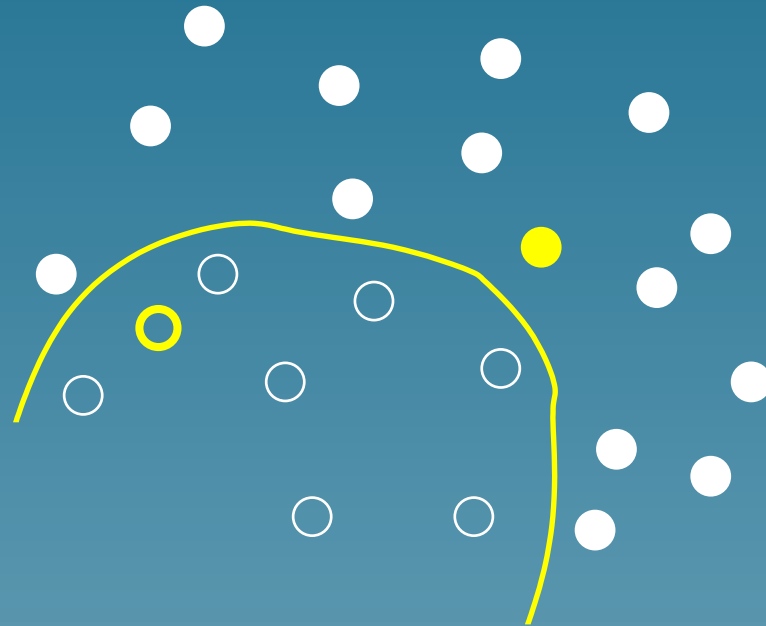


The pattern recognition problem



- Learn from labelled examples a discrimination rule

The pattern recognition problem

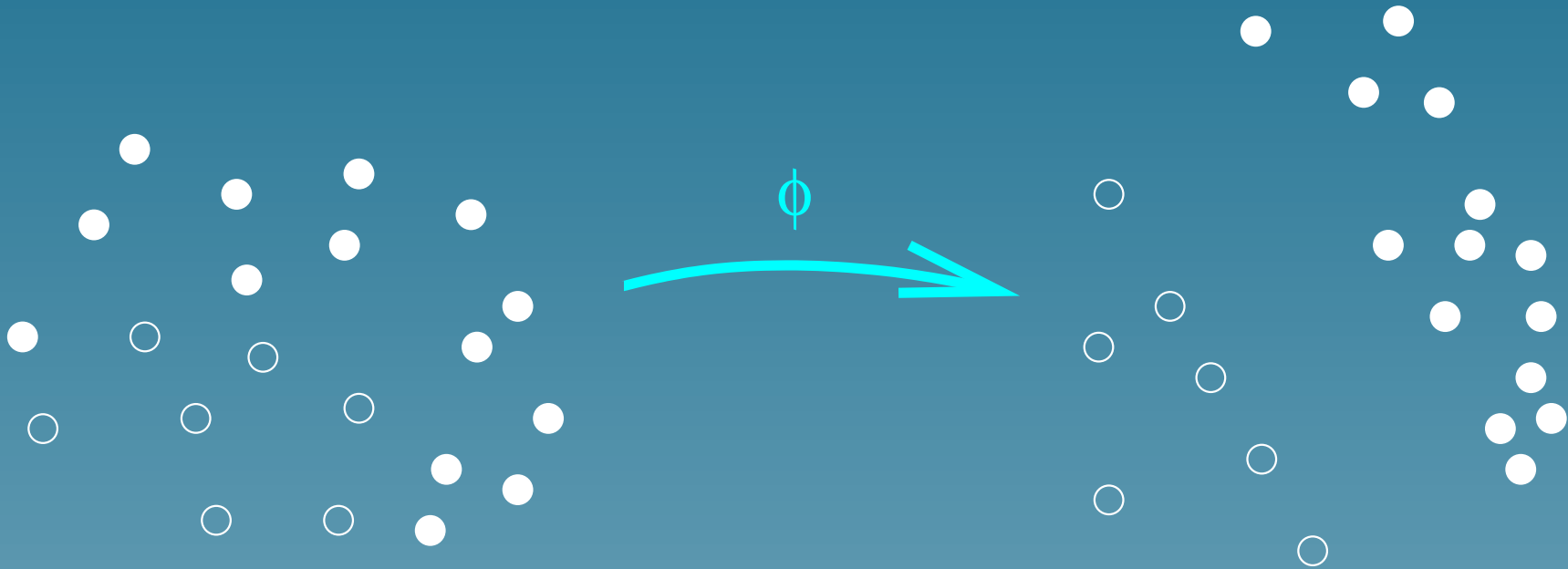


- Learn from labelled examples a **discrimination rule**
- Use it to **predict** the class of new points

Pattern recognition examples

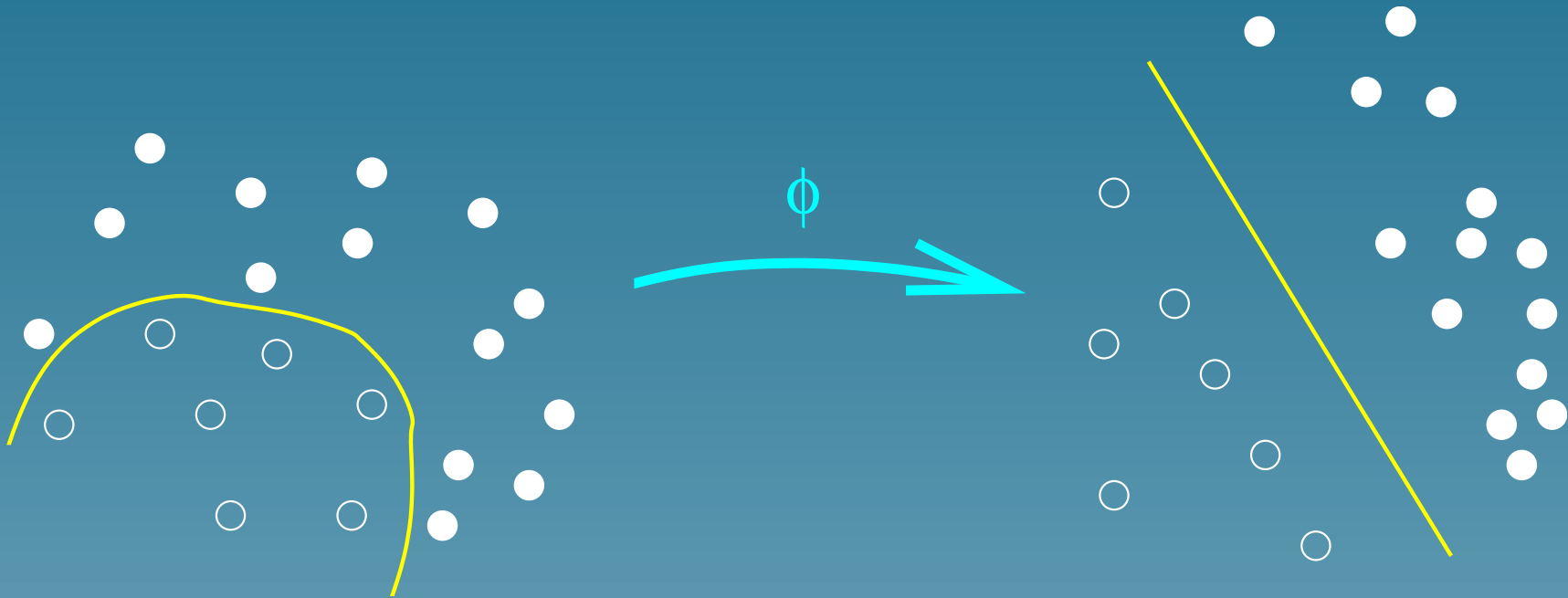
- Medical diagnosis (e.g., from microarrays)
- Drugability/activity of chemical compounds
- Gene function, structure, localization
- Protein interactions

Support Vector Machines for pattern recognition



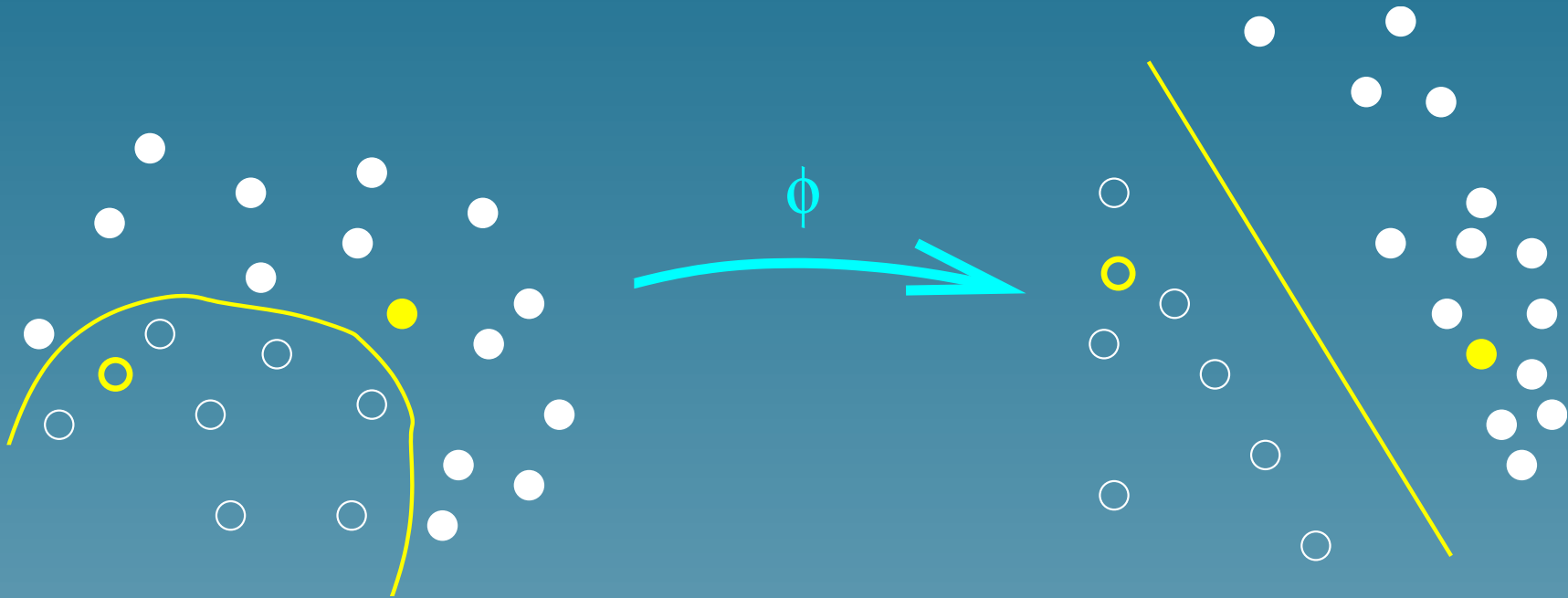
- Object x represented by the vector $\Phi(\vec{x})$ (feature space)

Support Vector Machines for pattern recognition



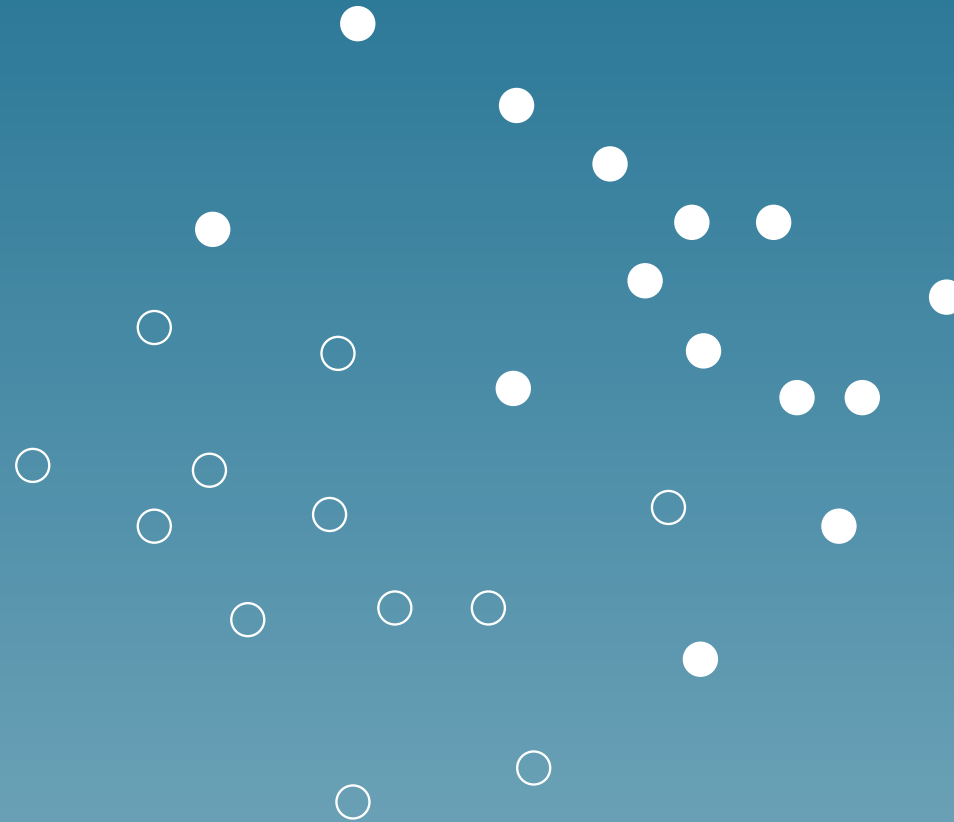
- Object x represented by the **vector** $\Phi(\vec{x})$ (**feature space**)
- **Linear separation** in the feature space

Support Vector Machines for pattern recognition

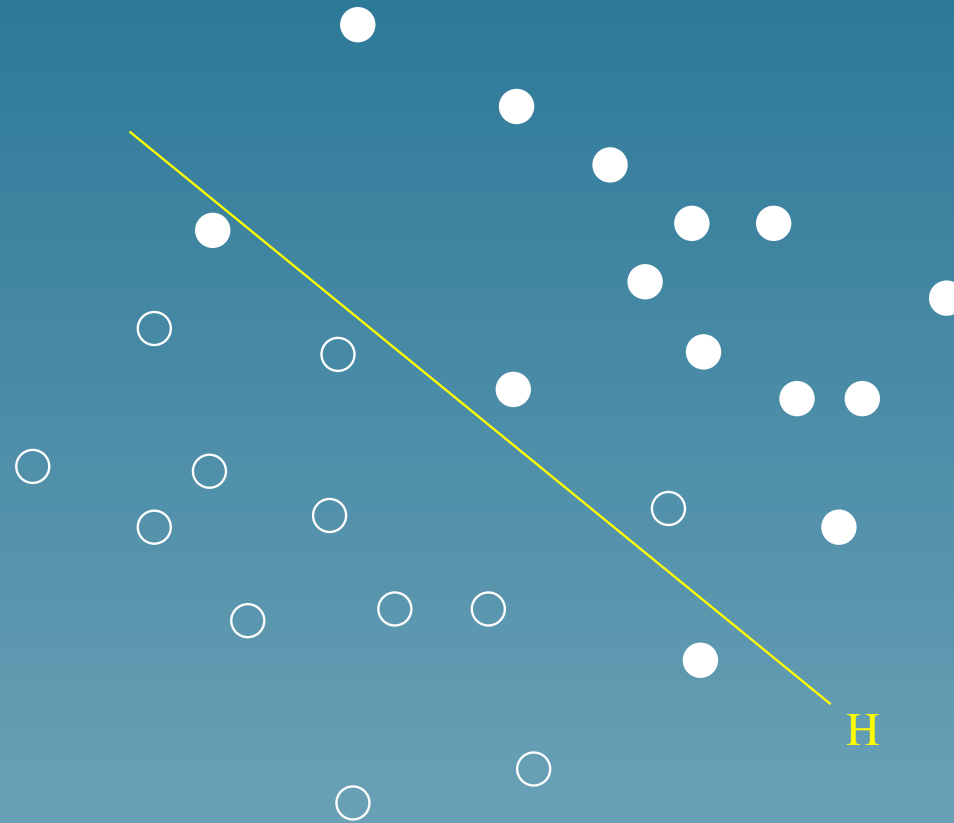


- Object x represented by the vector $\Phi(\vec{x})$ (feature space)
- Linear separation with large margin in the feature space

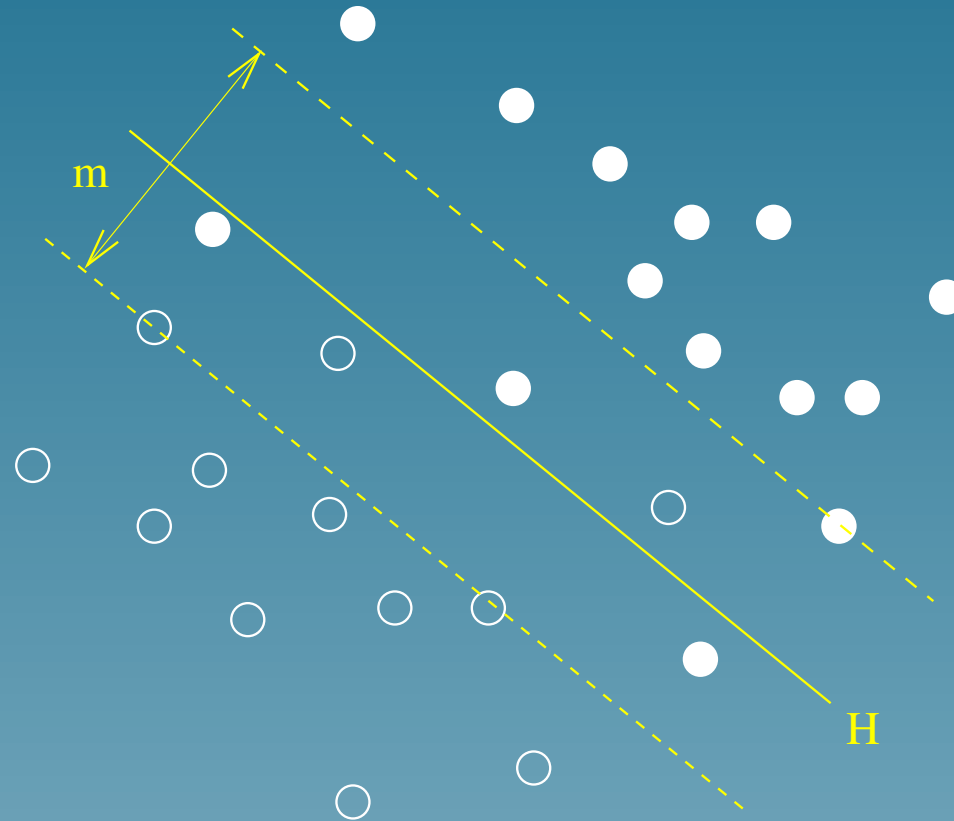
Large margin separation



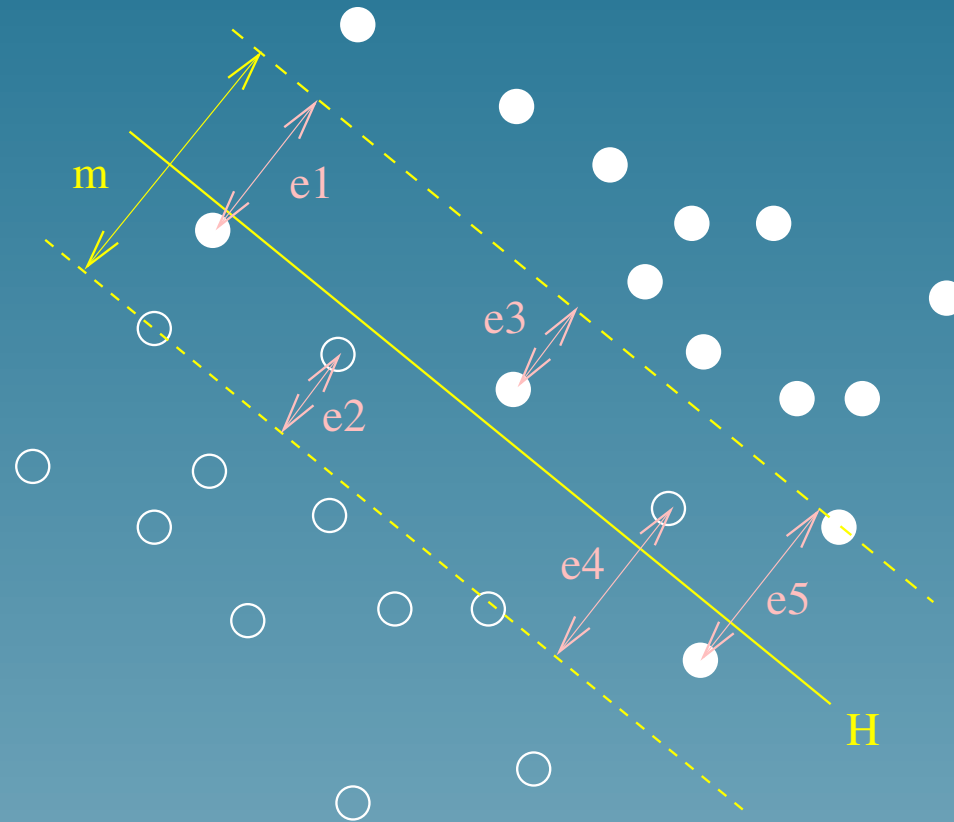
Large margin separation



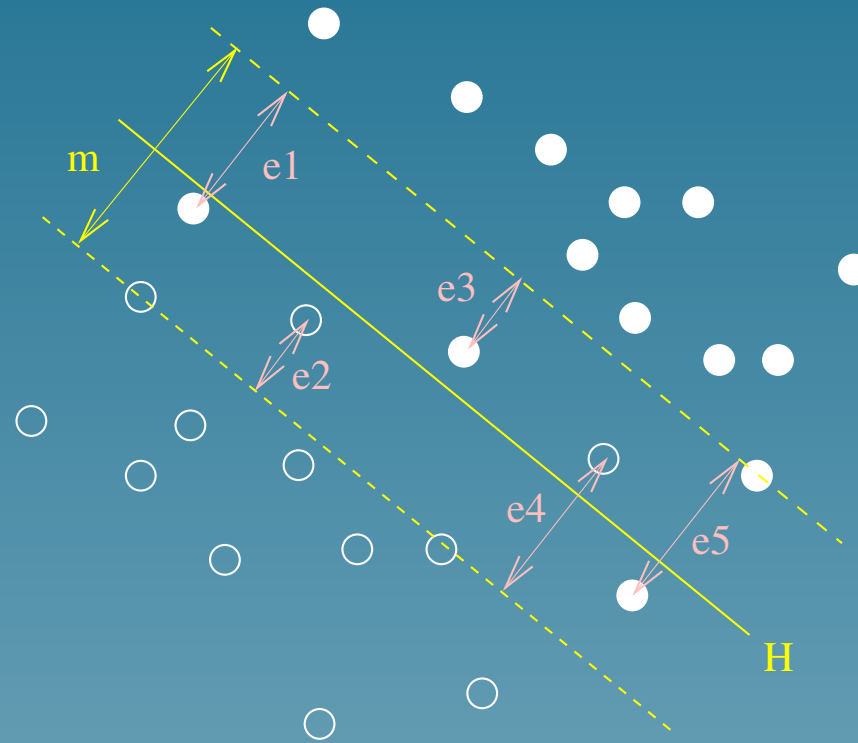
Large margin separation



Large margin separation



Large margin separation



$$\min_{H, m} \left\{ \frac{1}{m^2} + C \sum_i e_i \right\}$$

Dual formulation

The classification of a new point x is the sign of:

$$f(x) = \sum_i \alpha_i K(x, x_i) + b,$$

where α_i solves:

$$\begin{cases} \max_{\vec{\alpha}} \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\ \forall i = 1, \dots, n \quad 0 \leq \alpha_i \leq C \\ \sum_{i=1}^n \alpha_i y_i = 0 \end{cases}$$

with the notation:

$$K(x, x') = \Phi(\vec{x}) \cdot \Phi(\vec{x}')$$

The kernel trick for SVM

- The separation can be found without knowing $\Phi(x)$. Only the **kernel** matters:

$$K(x, y) = \Phi(\vec{x}) \cdot \Phi(\vec{y})$$

- Simple kernels $K(x, y)$ can correspond to complex $\vec{\Phi}$
- SVM work with **any sort of data** as soon as a kernel is defined

Kernel examples

- Linear :

$$K(x, x') = x \cdot x'$$

- Polynomial :

$$K(x, x') = (x \cdot x' + c)^d$$

- Gaussian RBF :

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

Kernels

For any set \mathcal{X} , a function $K : \mathcal{X} \times \mathcal{X} \rightarrow \mathbb{R}$ is a kernel iff:

- it is **symmetric** :

$$K(x, y) = K(y, x),$$

- it is **positive semi-definite**:

$$\sum_{i,j} a_i a_j K(x_i, x_j) \geq 0$$

for all $a_i \in \mathbb{R}$ and $x_i \in \mathcal{X}$

Advantages of SVM

- Works well on real-world applications
- Large dimensions, noise OK (?)
- Can be applied to **any kind of data** as soon as a kernel is available

Examples: SVM in bioinformatics

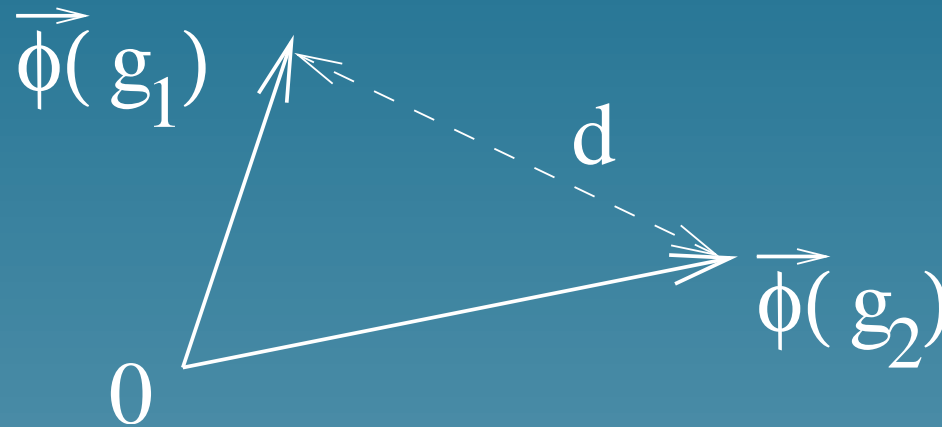
- Gene functional classification from microarray: Brown et al. (2000), Pavlidis et al. (2001)
- Tissue classification from microarray: Mukherje et al. (1999), Furey et al. (2000), Guyon et al. (2001)
- Protein family prediction from sequence: Jaakkola et al. (1998)
- Protein secondary structure prediction: Hua et al. (2001)
- Protein subcellular localization prediction from sequence: Hua et al. (2001)

Kernel methods

Let $K(x, y)$ be a given kernel. Then is it possible to perform other linear algorithms **implicitly** in the feature space such as:

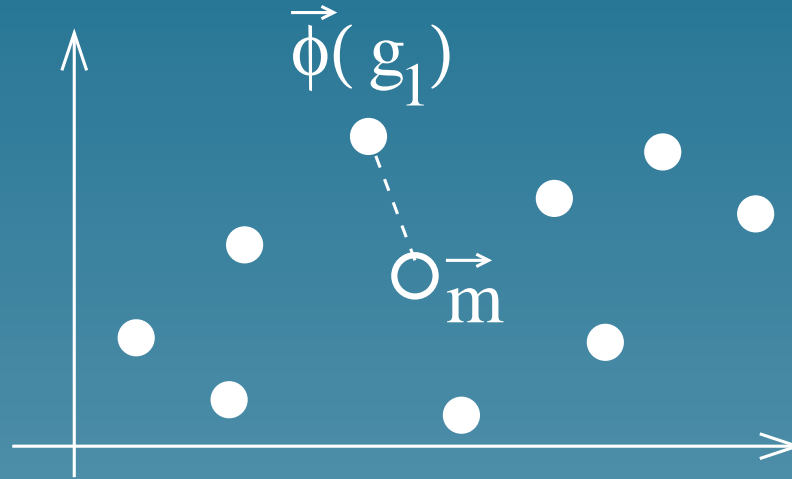
- Compute the distance between points
- Principal component analysis (PCA)
- Canonical correlation analysis (CCA)

Compute the distance between objects



$$\begin{aligned}
 d(g_1, g_2)^2 &= \|\vec{\Phi}(g_1) - \vec{\Phi}(g_2)\|^2 \\
 &= \left(\vec{\Phi}(g_1) - \vec{\Phi}(g_2)\right) \cdot \left(\vec{\Phi}(g_1) - \vec{\Phi}(g_2)\right) \\
 &= \vec{\Phi}(g_1) \cdot \vec{\Phi}(g_1) + \vec{\Phi}(g_2) \cdot \vec{\Phi}(g_2) - 2\vec{\Phi}(g_1) \cdot \vec{\Phi}(g_2) \\
 d(g_1, g_2)^2 &= K(g_1, g_1) + K(g_2, g_2) - 2K(g_1, g_2)
 \end{aligned}$$

Distance to the center of mass

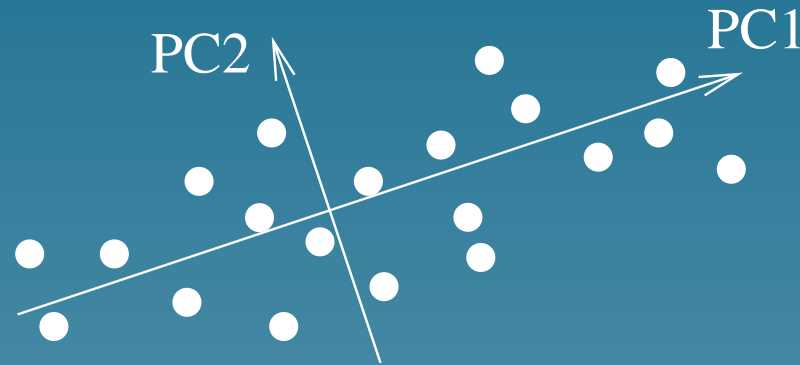


Center of mass: $\vec{m} = \frac{1}{N} \sum_{i=1}^N \vec{\Phi}(g_i)$, hence:

$$\|\vec{\Phi}(g_1) - \vec{m}\|^2 = \vec{\Phi}(g_1) \cdot \vec{\Phi}(g_1) - 2\vec{\Phi}(g_1) \cdot \vec{m} + \vec{m} \cdot \vec{m}$$

$$= K(g_1, g_1) - \frac{2}{N} \sum_{i=1}^N K(g_1, g_i) + \frac{1}{N^2} \sum_{i,j=1}^N K(g_i, g_j)$$

Principal component analysis

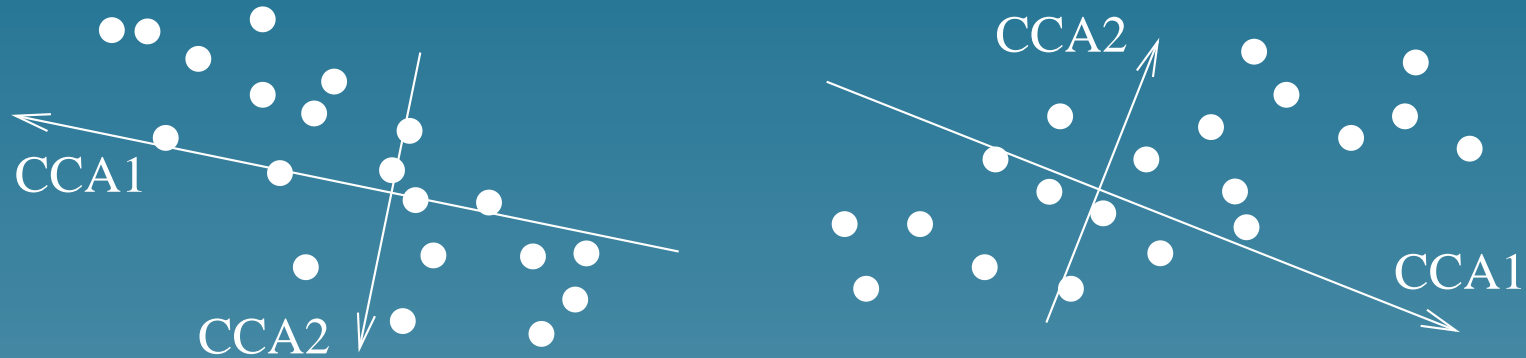


It is equivalent to find the eigenvectors of

$$\begin{aligned} K &= \left(\vec{\Phi}(g_i) \cdot \vec{\Phi}(g_j) \right)_{i,j=1\dots N} \\ &= \left(K(g_i, g_j) \right)_{i,j=1\dots N} \end{aligned}$$

Useful to project the objects on small-dimensional spaces (feature extraction).

Canonical correlation analysis



K_1 and K_2 are two kernels for the same objects. CCA can be performed by solving the following generalized eigenvalue problem:

$$\begin{pmatrix} 0 & K_1 K_2 \\ K_2 K_1 & 0 \end{pmatrix} \vec{\xi} = \rho \begin{pmatrix} K_1^2 & 0 \\ 0 & K_2^2 \end{pmatrix} \vec{\xi}$$

Useful to find correlations between different representations of the same objects (ex: genes, ...)

Part 2

Local alignment kernel for strings

(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?

Related work

- Spectrum kernel (Leslie et al.):

$$K(x_1 \dots x_m, y_1 \dots y_n) = \sum_{i=1}^{m-k} \sum_{j=1}^{n-k} \delta(x_i \dots x_{i+k}, y_j \dots y_{j+k}).$$

Related work

- **Spectrum kernel** (Leslie et al.):

$$K(x_1 \dots x_m, y_1 \dots y_n) = \sum_{i=1}^{m-k} \sum_{j=1}^{n-k} \delta(x_i \dots x_{i+k}, y_j \dots y_{j+k}).$$

- **Fisher kernel** (Jaakkola et al.): given a statistical model $(p_\theta, \theta \in \Theta \subset \mathbb{R}^d)$:

$$\phi(x) = \nabla_\theta \log p_\theta(x)$$

and use the Fisher information matrix.

Local alignment

- For two strings x and y , a local alignment π with gaps is:

```

ABCD EF---G-HI JKL
      ||         ||  |
MNO  EEPORGS-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

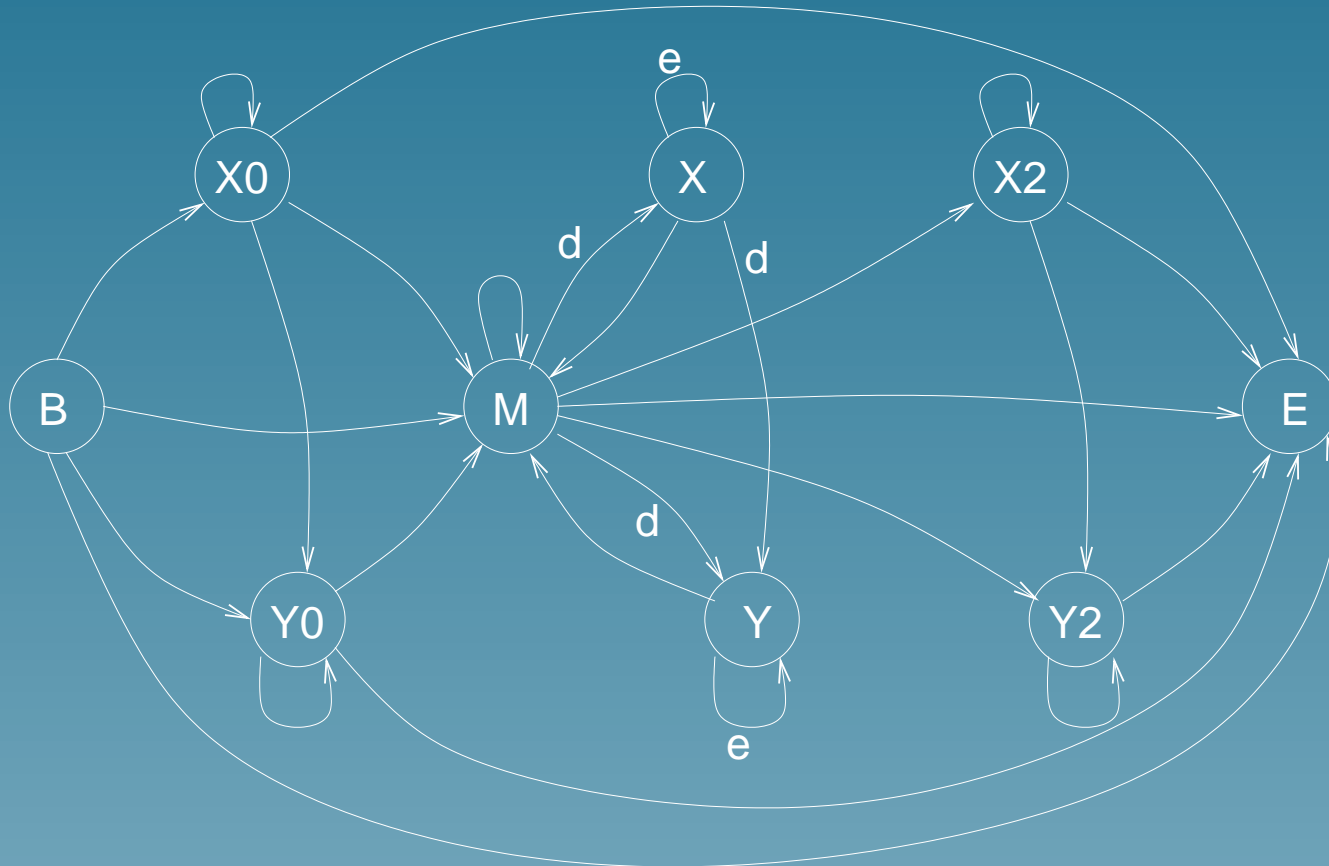
$$K_{LA}^{(\beta)}(x, y) = \sum_{\pi \in \Pi(x, y)} \exp(\beta s(x, y, \pi)),$$

Properties

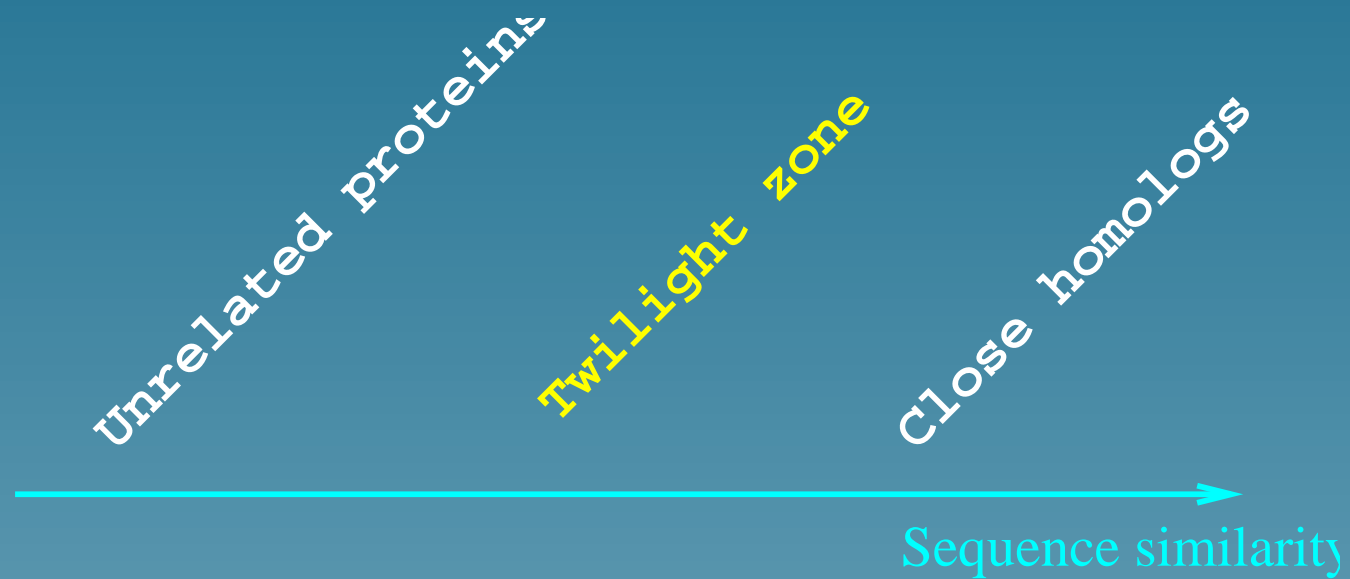
$$K_{LA}^{(\beta)}(x, y) = \sum_{\pi \in \Pi(x, y)} \exp(\beta s(x, y, \pi)),$$

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

Kernel computation

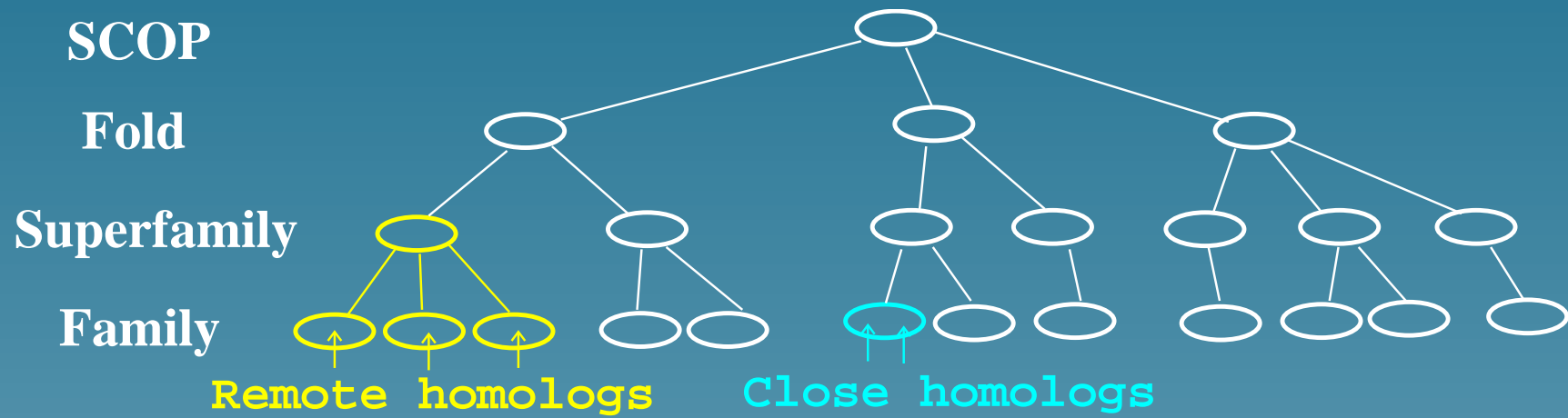


Application: remote homology detection



- Same structure/function but sequence diverged
- Remote homology can not be found by direct sequence similarity

SCOP database



A benchmark experiment

- Can we predict the **superfamily** of a domain if we have not seen any member of its **family** before?

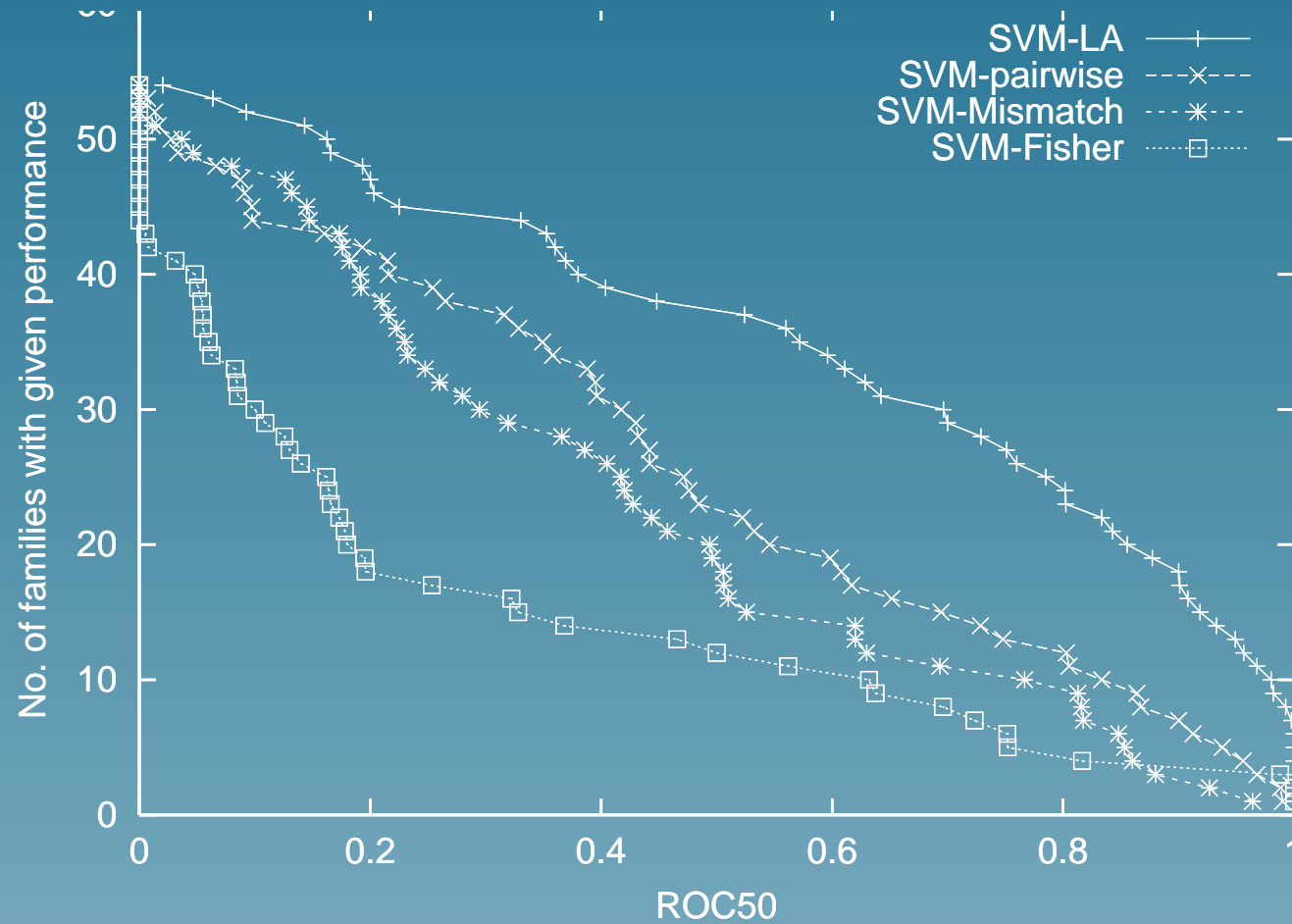
A benchmark experiment

- Can we predict the **superfamily** of a domain if we have not seen any member of its **family** before?
- During **learning**: remove a family and learn the difference between the superfamily and the rest

A benchmark experiment

- Can we predict the **superfamily** of a domain if we have not seen any member of its **family** before?
- During **learning**: remove a family and learn the difference between the superfamily and the rest
- Then, use the model to **test** each domain of the family removed

SCOP superfamily recognition benchmark

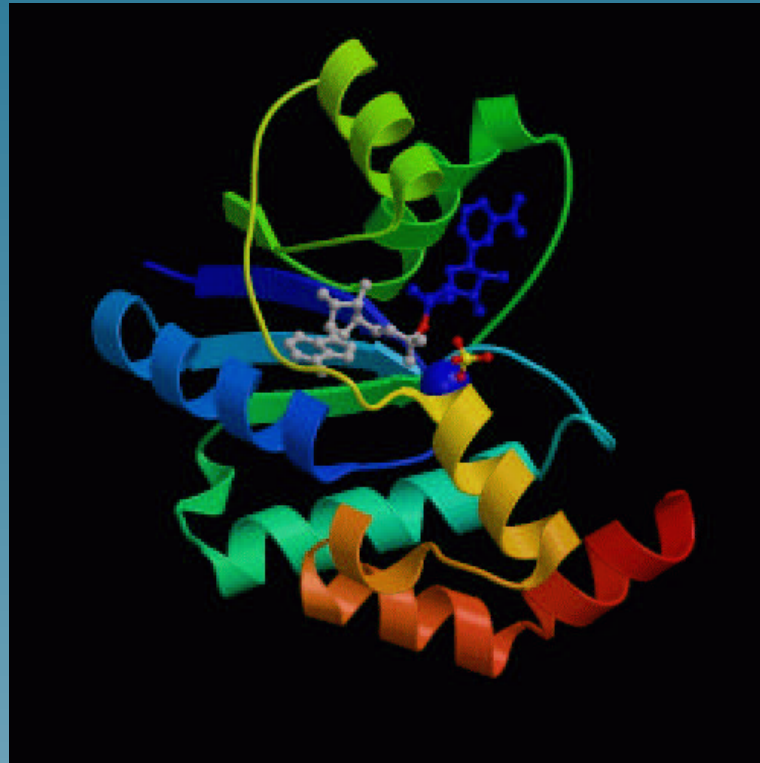


Part 3

Detecting pathway activity from microarray data

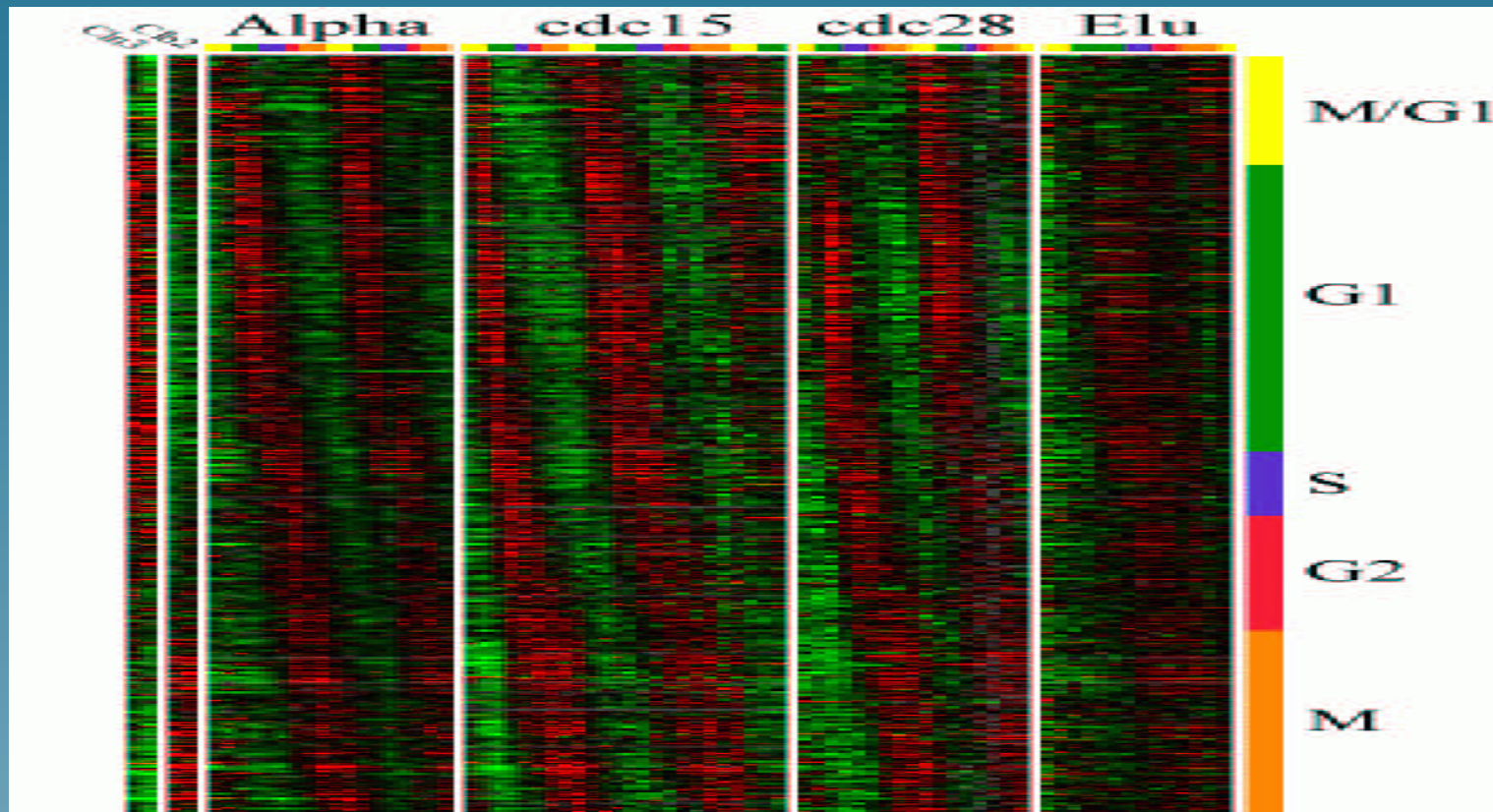
(with M. Kanehisa, *ECCB 2003*)

Genes encode proteins which can catalyse chemical reactions



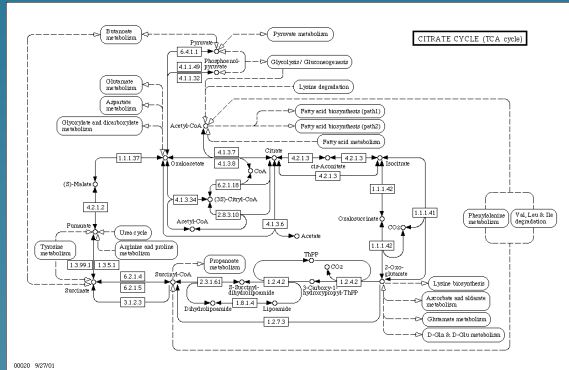
Nicotinamide Mononucleotide Adenylyltransferase With Bound Nad⁺

Microarray technology monitors mRNA quantity

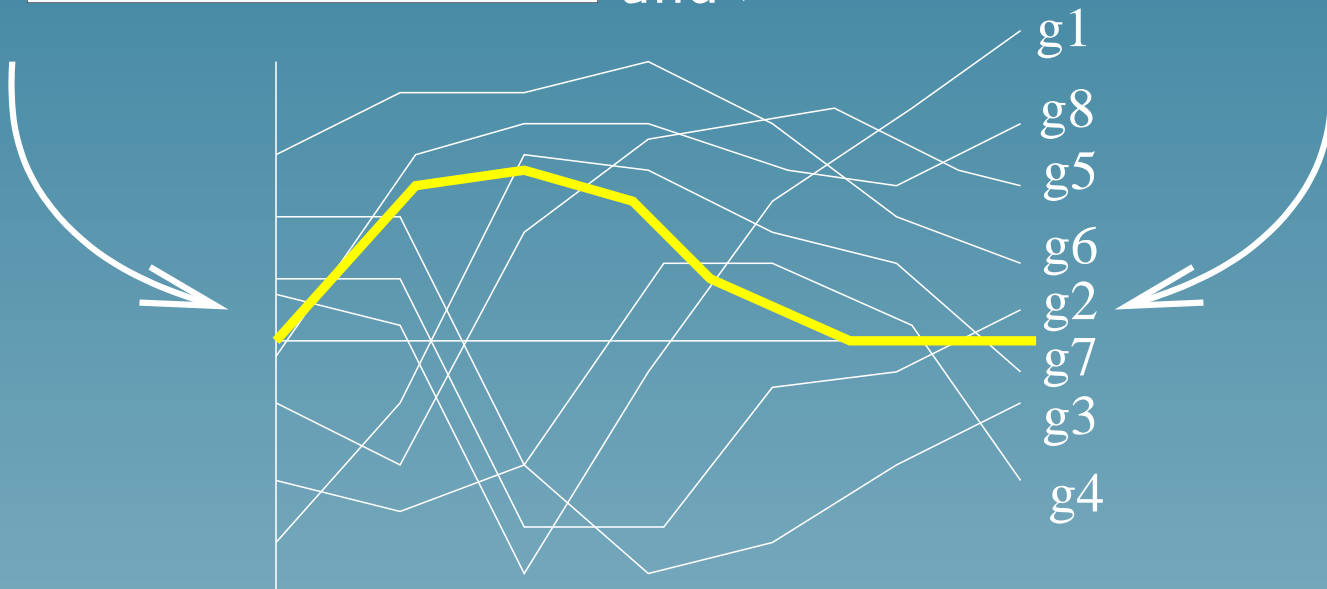


(From Spellman et al., 1998)

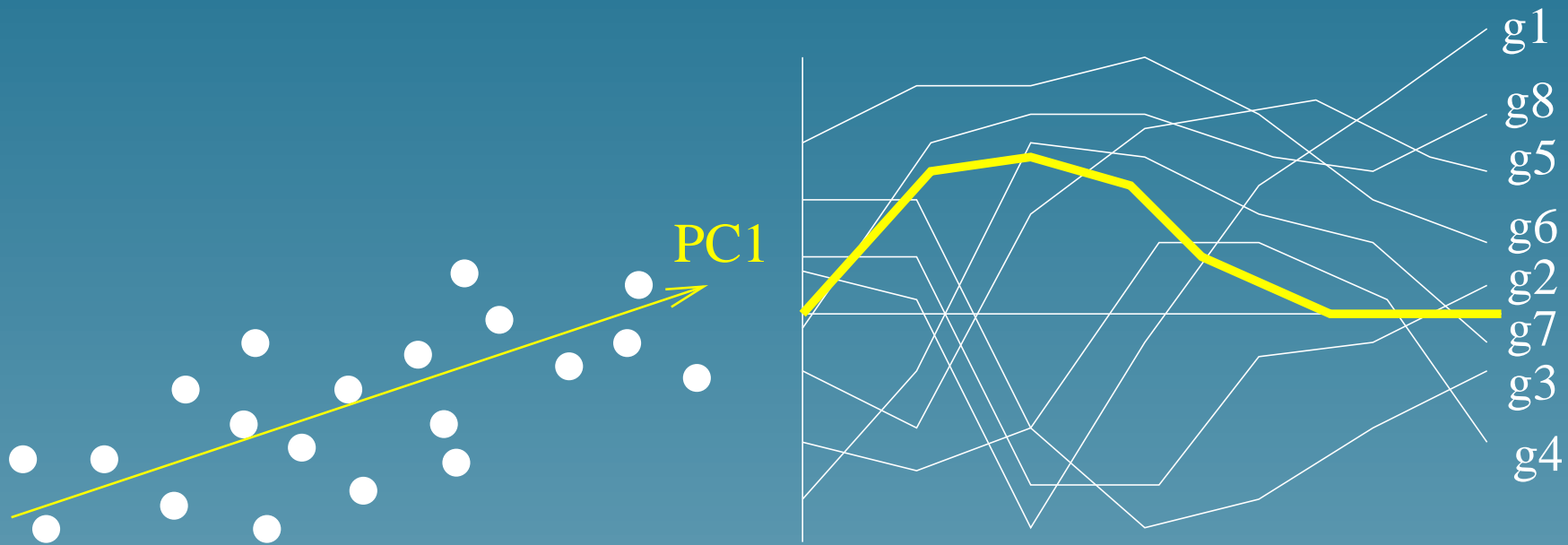
A useful first step



and



Using microarray only



PCA finds the directions (*profiles*) explaining the **largest amount of variations** among expression profiles.

PCA formulation

- Let $f_v(i)$ be the **projection** of the i -th profile onto v .
- The **amount of variation** captured by f_v is:

$$h_1(v) = \sum_{i=1}^N f_v(i)^2$$

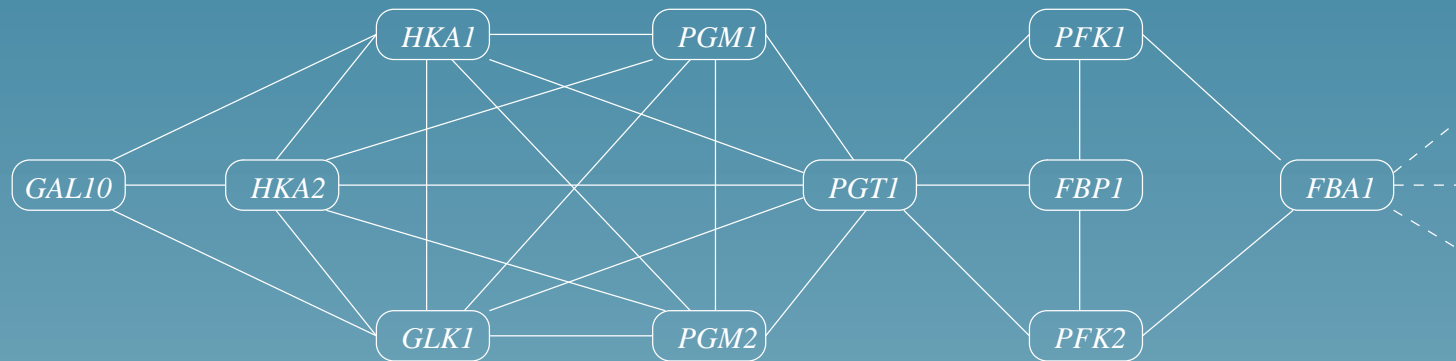
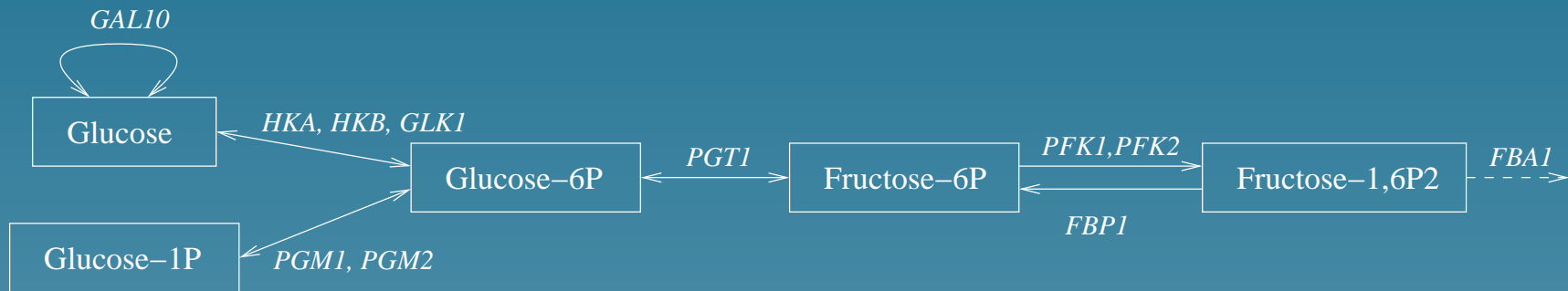
- PCA finds an orthonormal basis by solving successively:

$$\max_v h_1(v)$$

Issues with PCA

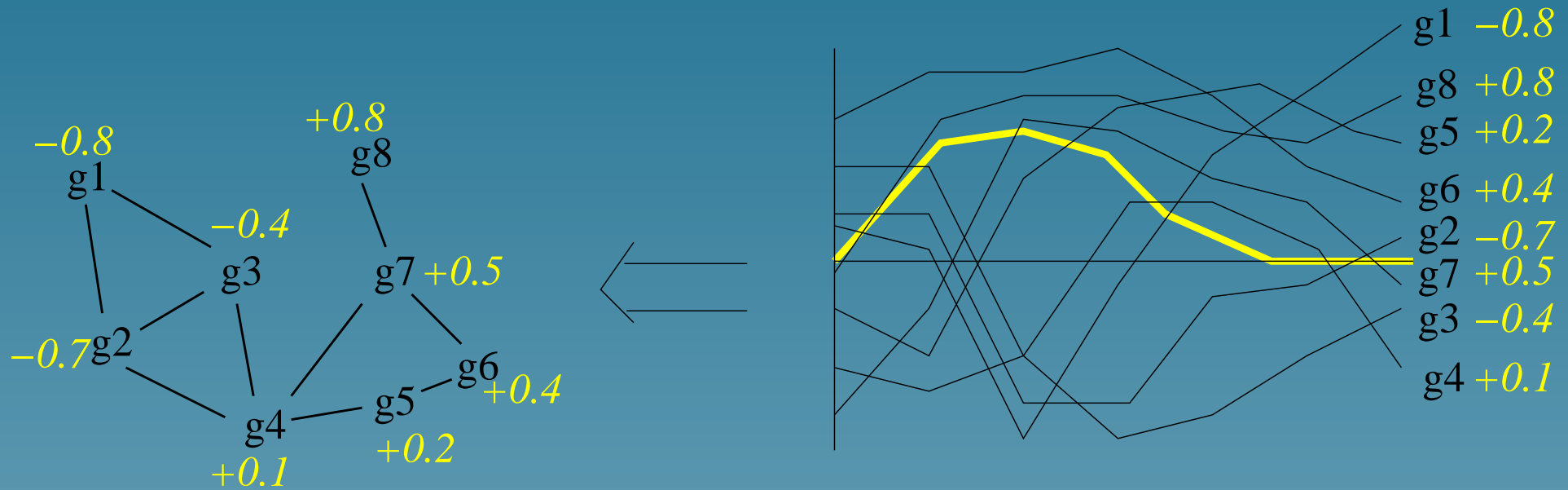
- PCA is useful if there is a small number of strong signal
- In concrete applications, we observe a **noisy superposition** of many events
- Using a prior knowledge of metabolic networks can help denoising the information detected by PCA

The metabolic gene network



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

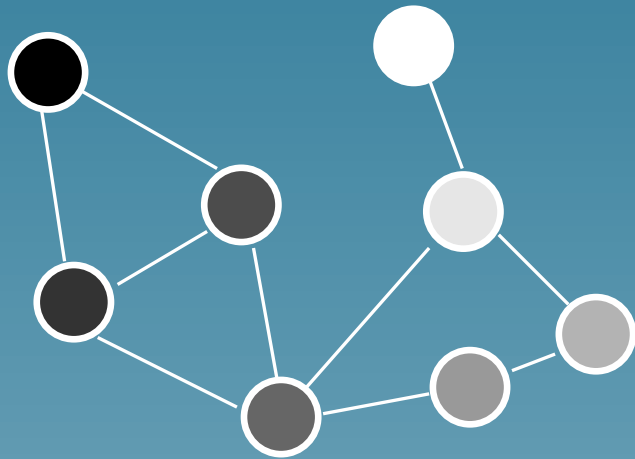
Mapping f_v to the metabolic gene network



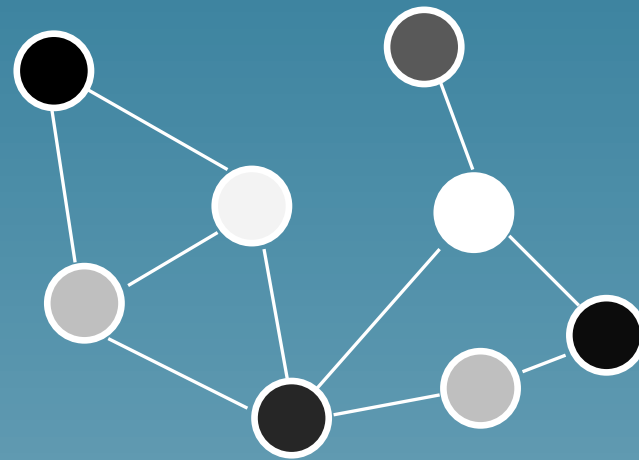
Does it look interesting or not?

Important hypothesis

If v is related to a metabolic activity, then f_v should **vary** "smoothly" on the graph

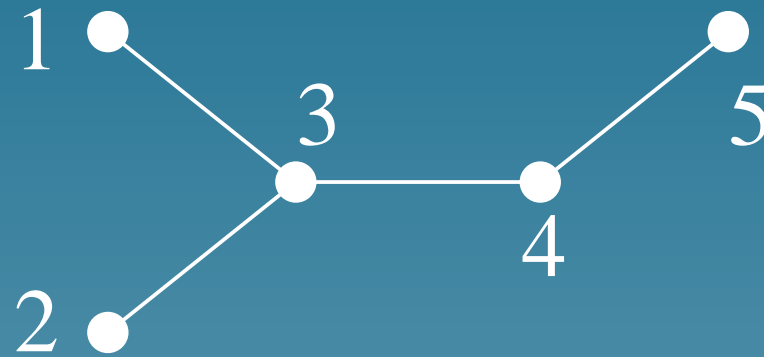


Smooth



Rugged

Graph Laplacian $L = D - A$

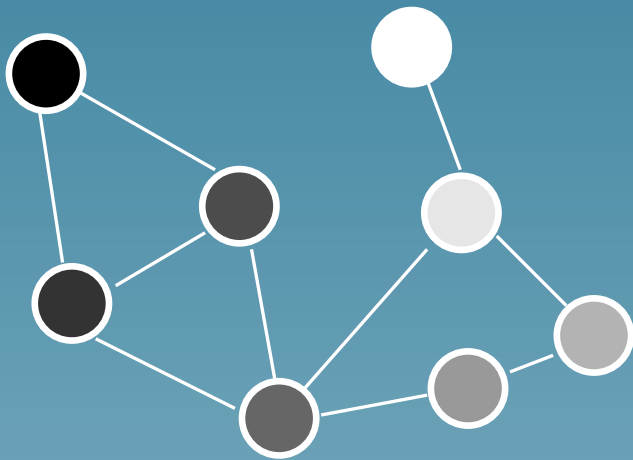


$$L = \begin{pmatrix} -1 & 0 & 1 & 0 & 0 \\ 0 & -1 & 1 & 0 & 0 \\ 1 & 1 & -3 & 1 & 0 \\ 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 1 & -1 \end{pmatrix}$$

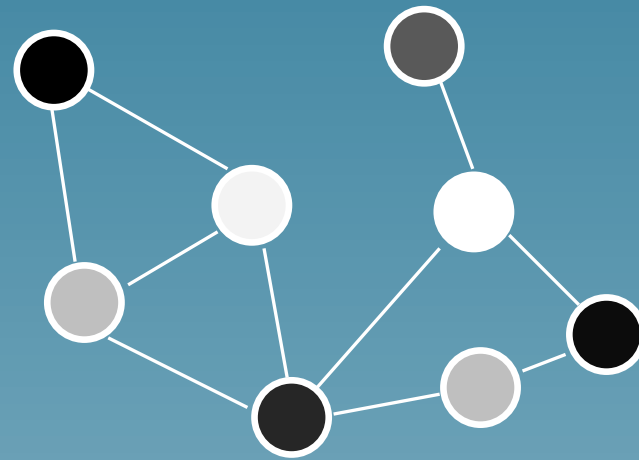
Smoothness quantification

$$h_2(f) = \frac{f^\top \exp(-\beta L) f}{f^\top f}$$

is large when f is smooth



$$h(f) = 2.5$$



$$h(f) = 34.2$$

Motivation

For a candidate profile v ,

- $h_1(f_v)$ is large when v captures a lot of natural variation among profiles
- $h_2(f_v)$ is large when f_v is smooth on the graph

Try to maximize both terms in the same time

Problem reformulation

Find a function f_v and a function f_2 such that:

- $h_1(f_v)$ be large
- $h_2(f_2)$ be large
- $\text{corr}(f_v, f_2)$ be large

by solving:

$$\max_{(f_v, f_2)} \text{corr}(f_v, f_2) \times \frac{h_1(f_v)}{h_1(f_v) + \delta} \times \frac{h_2(f_2)}{h_2(f_2) + \delta}$$

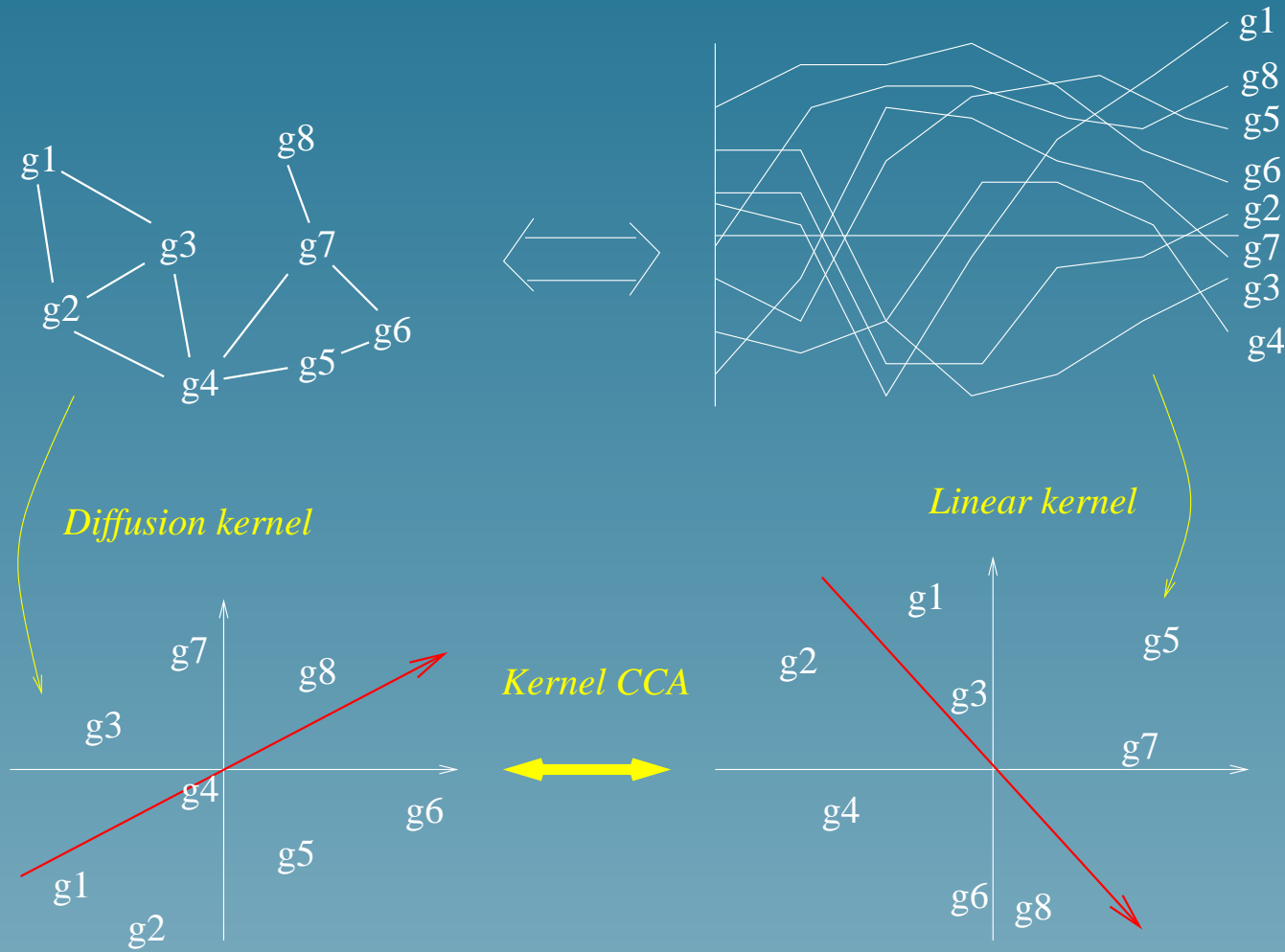
Solving the problem

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

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

where $[K_1]_{i,j} = e_i^\top e_j$ and $K_2 = \exp(-L)$.
Then, $f_v = K_1 \alpha$ and $f_2 = K_2 \beta$.

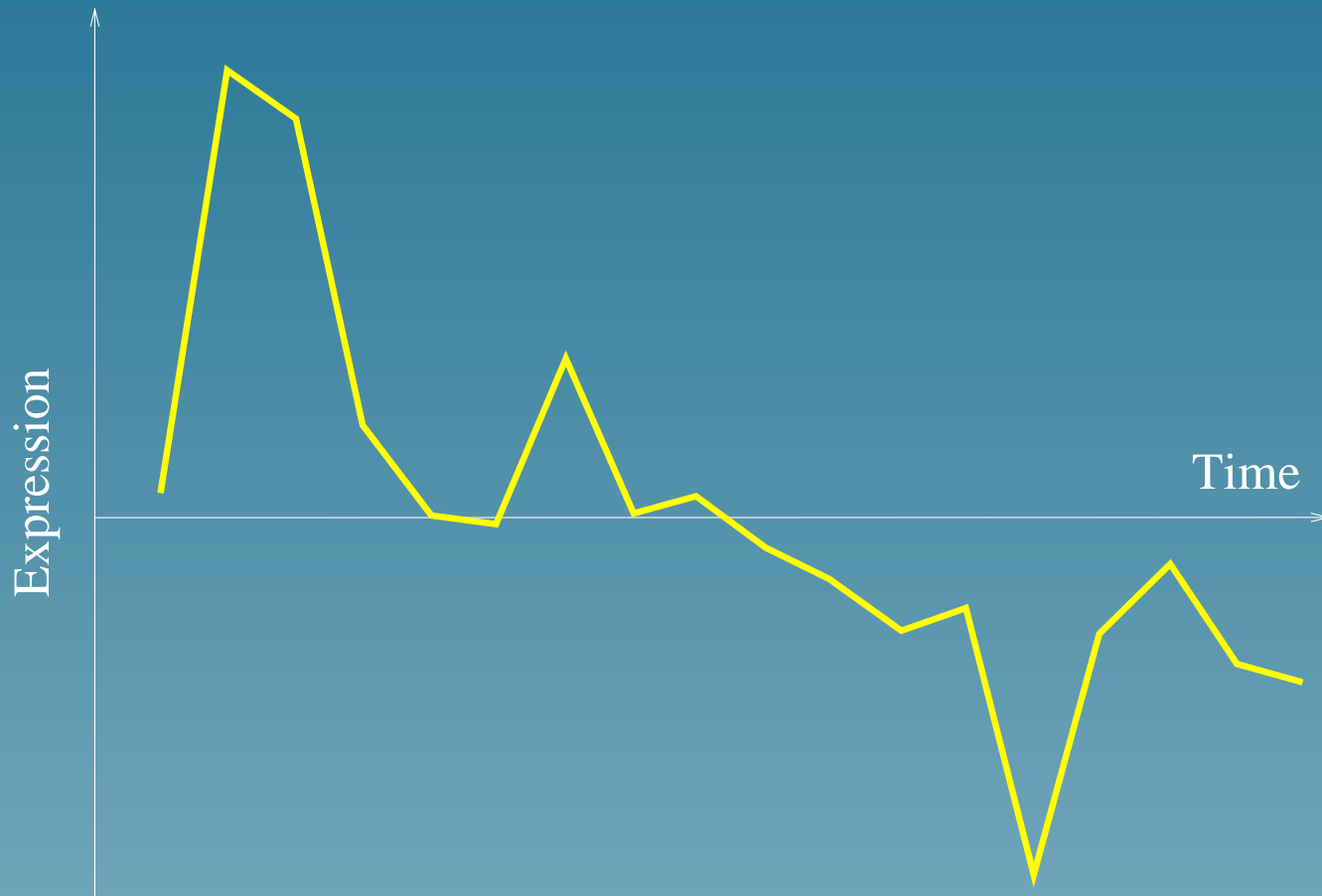
The kernel point of view...



Data

- **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

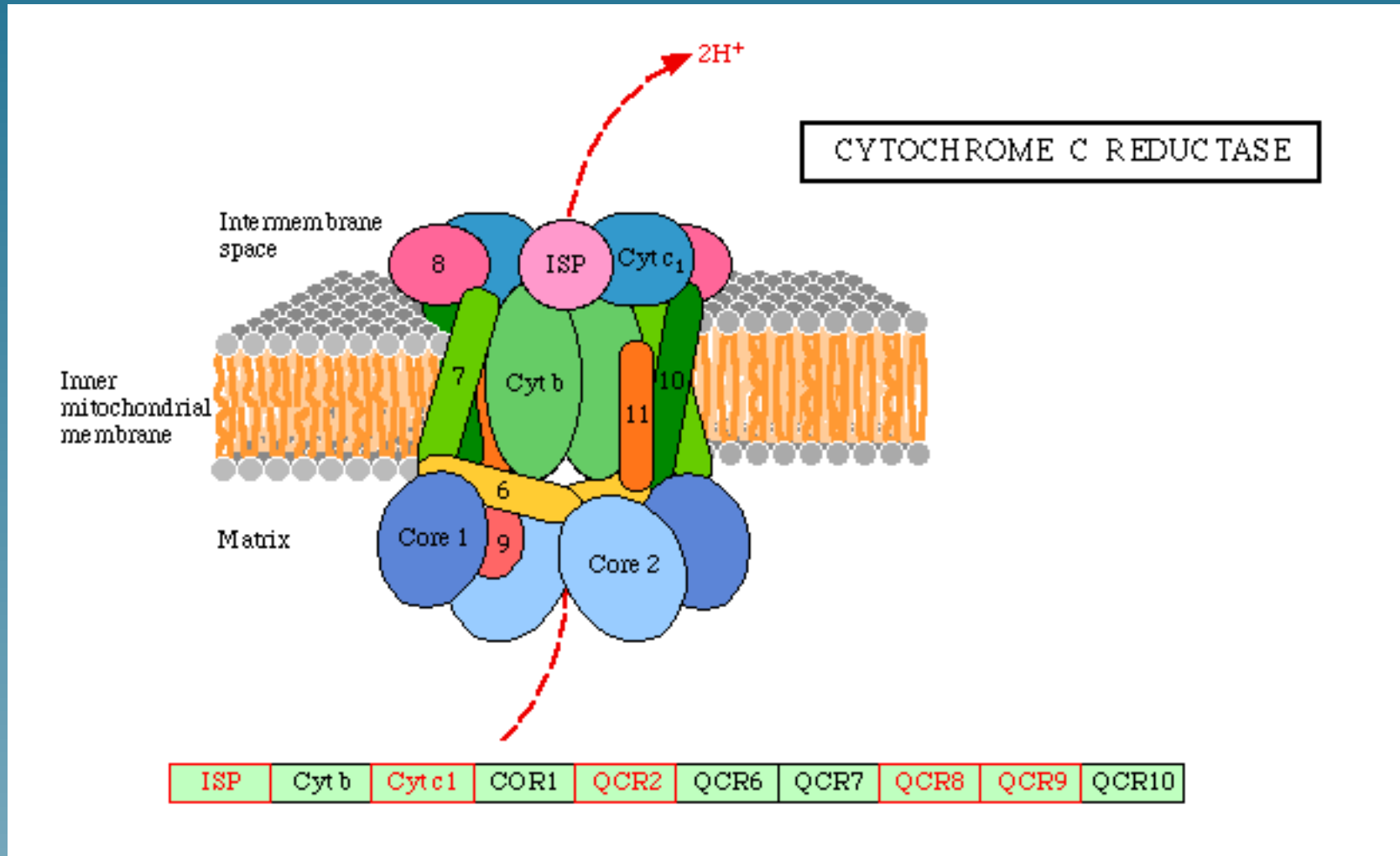


Related metabolic pathways

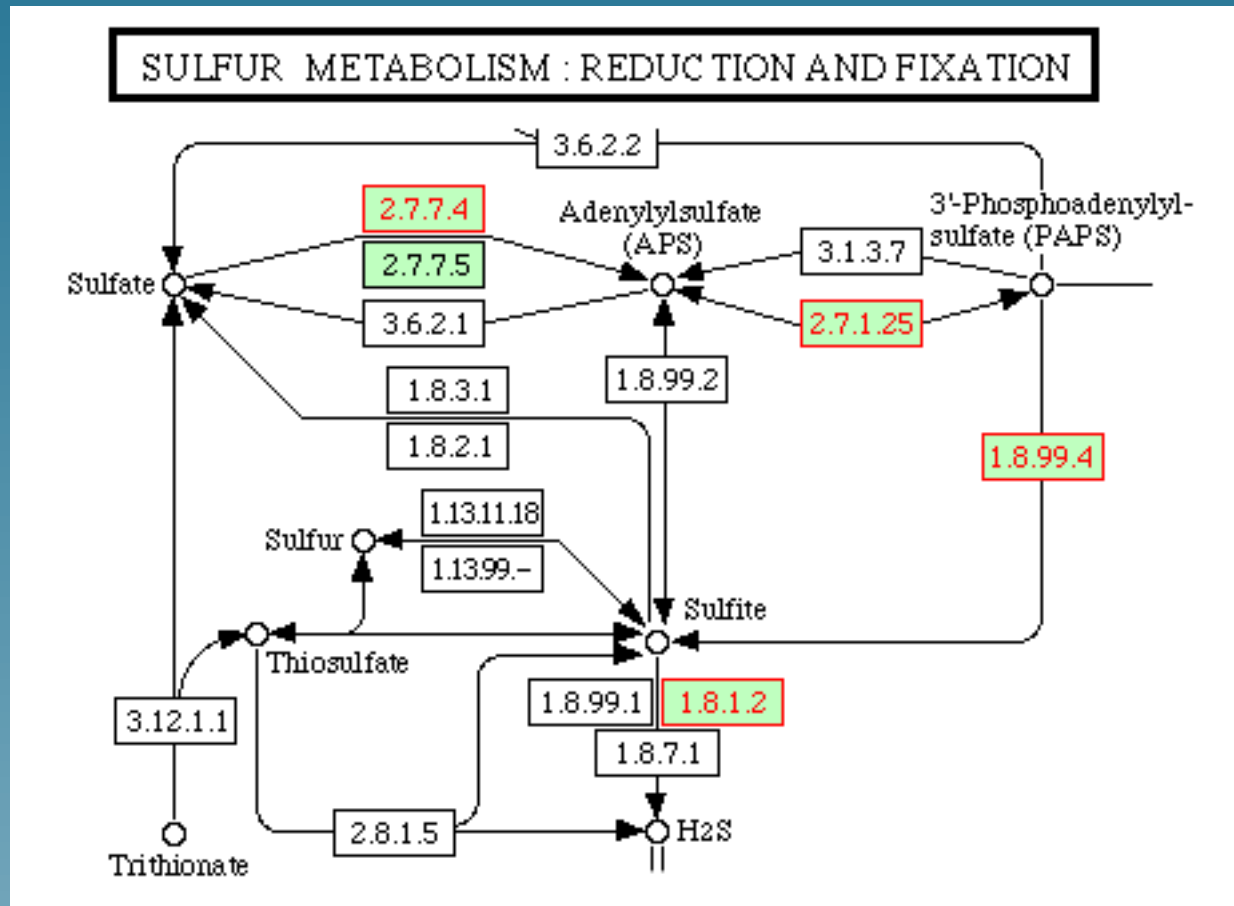
50 genes with highest $s_2 - s_1$ belong to:

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

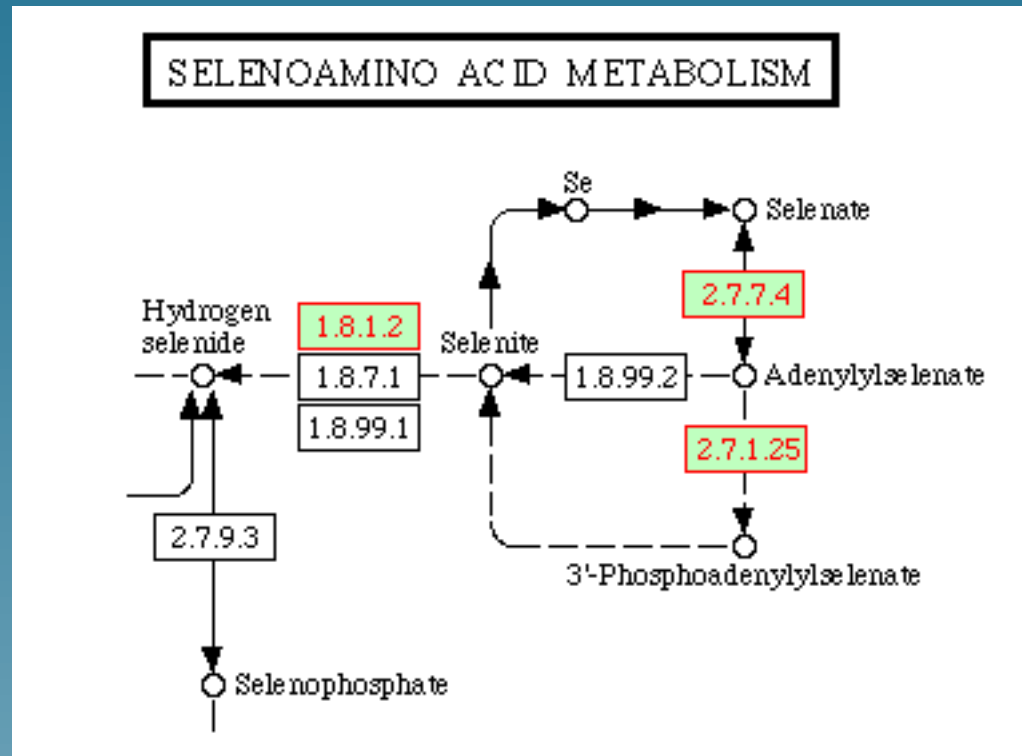
Related genes



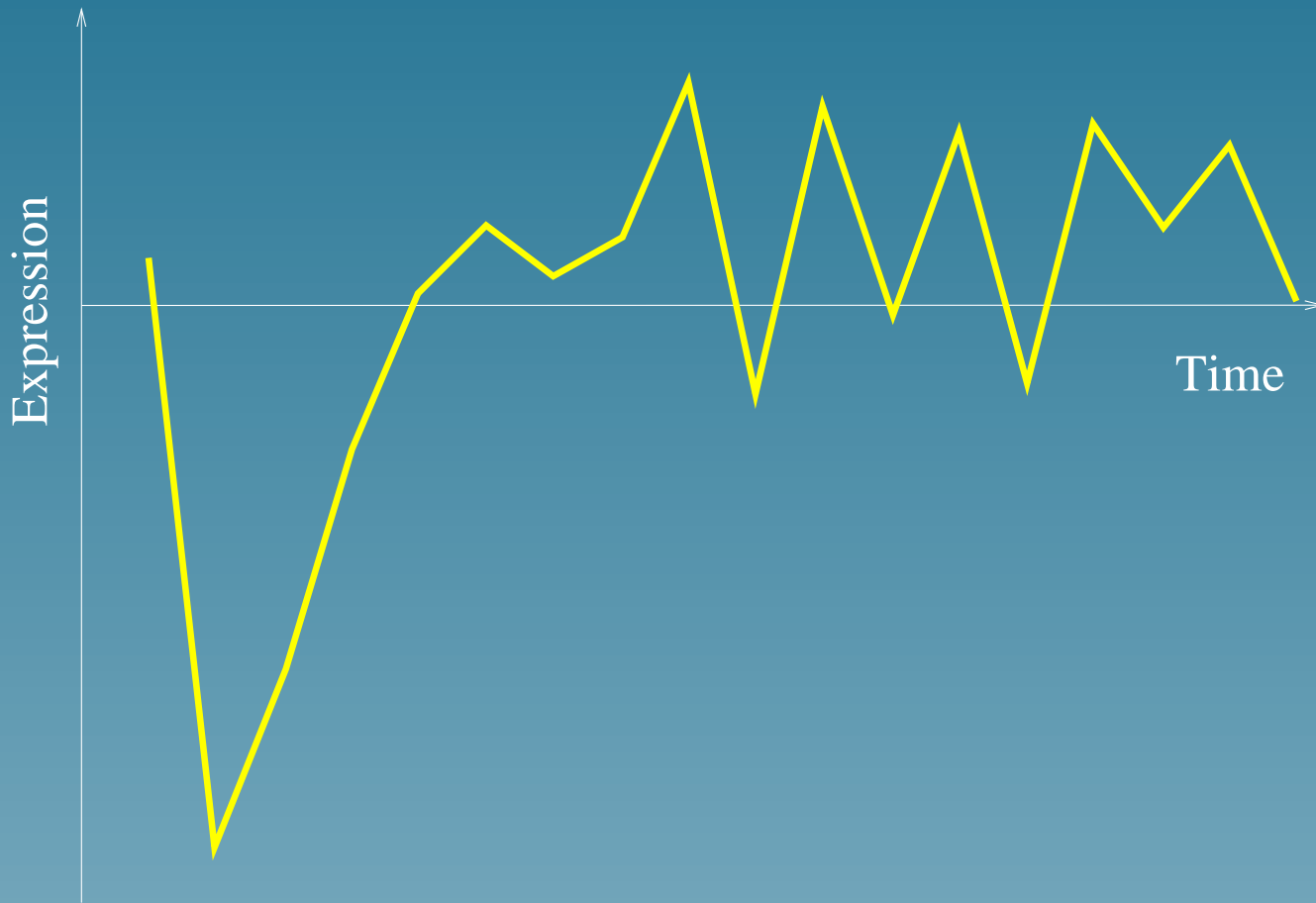
Related genes



Related genes



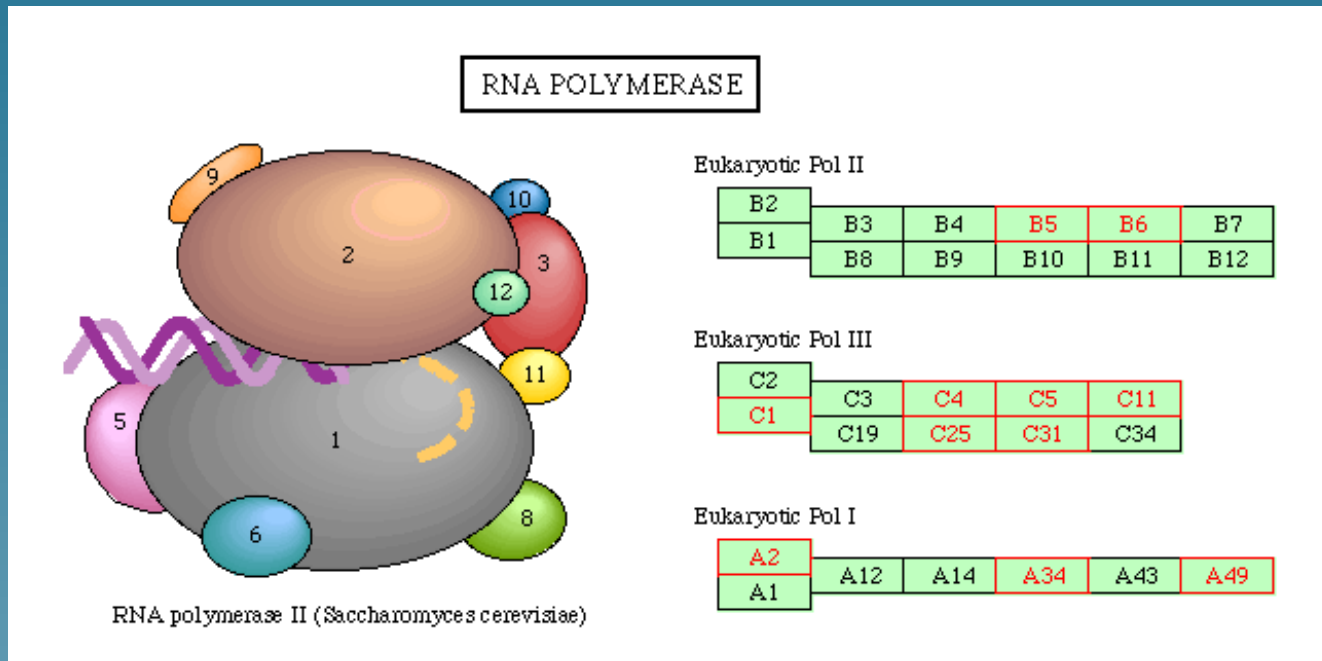
Opposite pattern



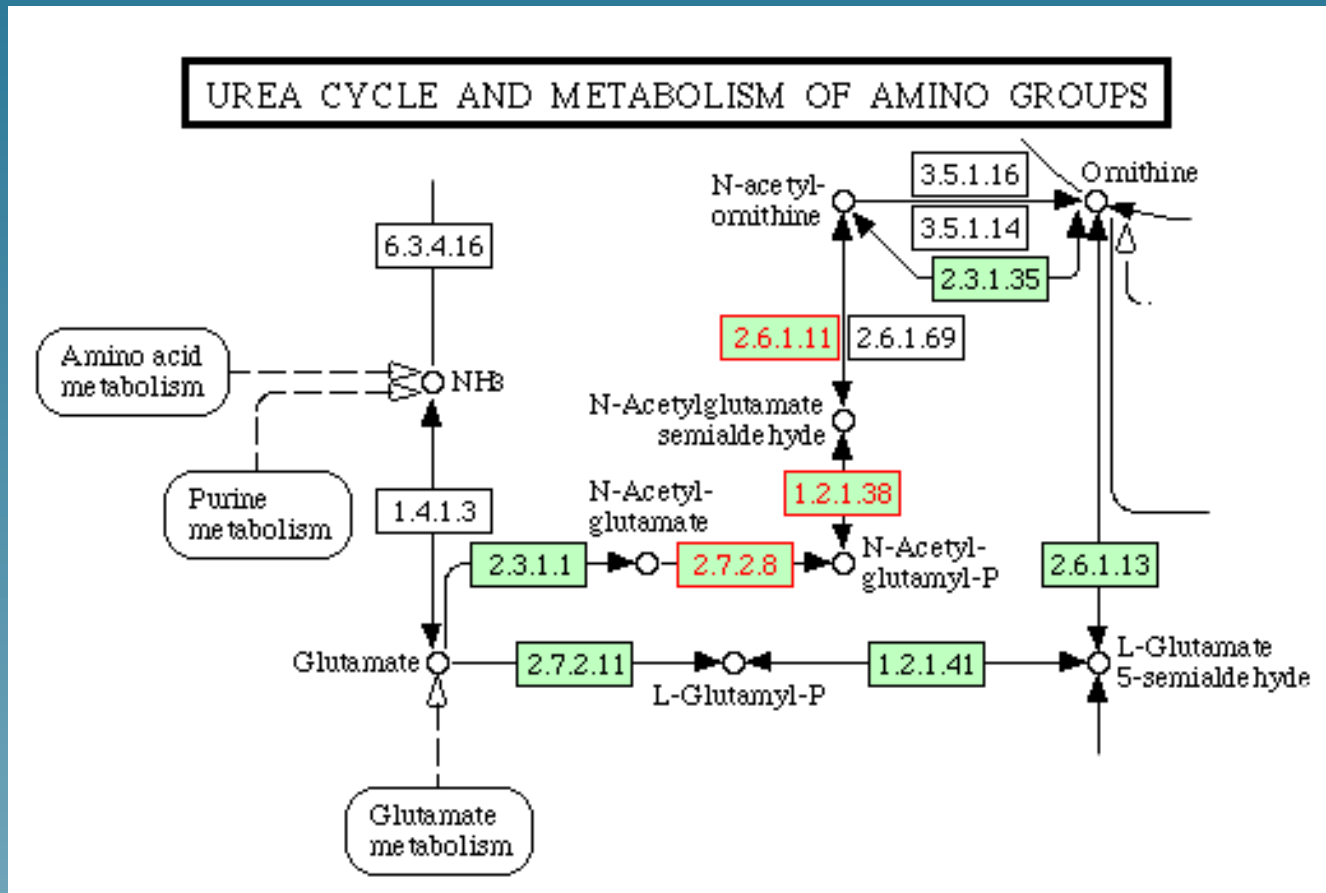
Related genes

- 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...

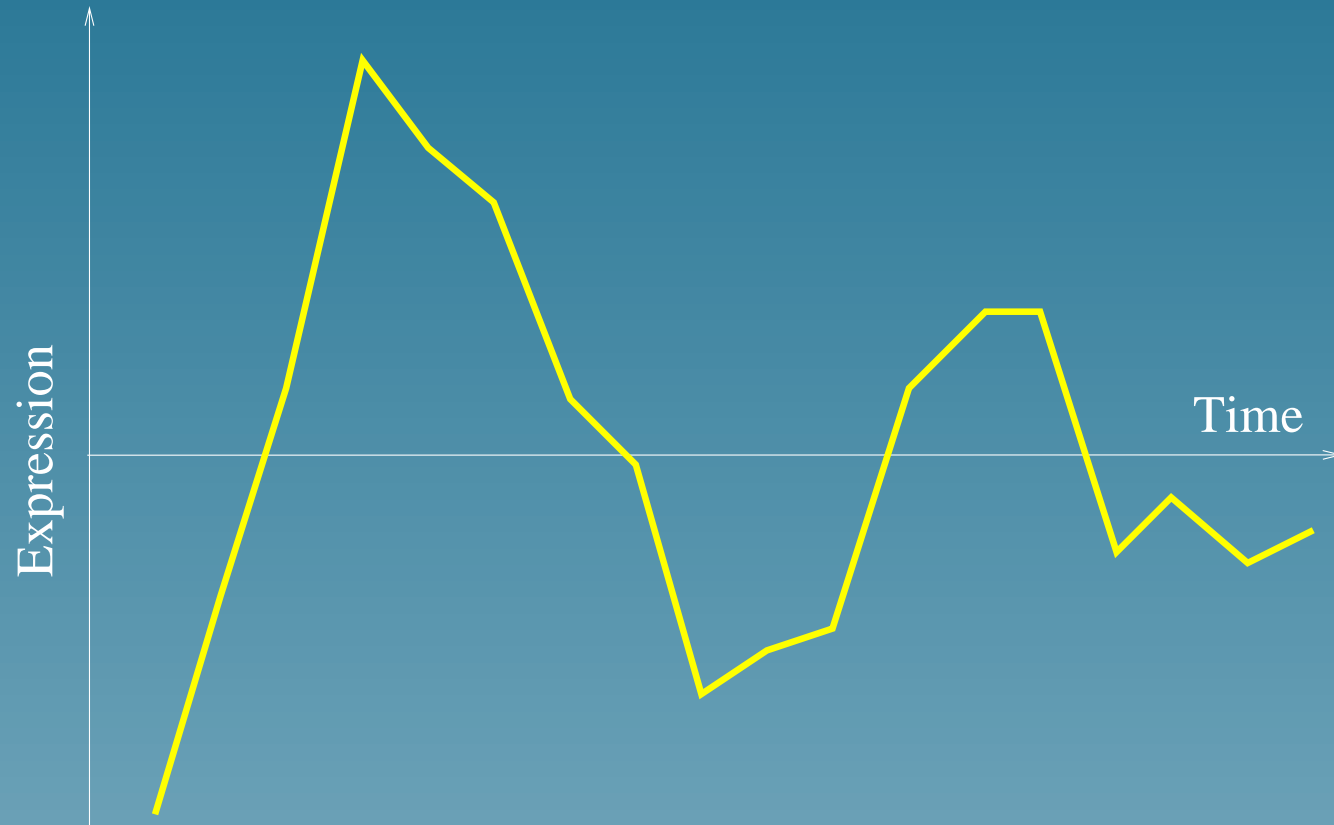
Related genes



Related genes



Second pattern



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**
- SVM and kernel methods **work well** on real-life problems, in particular in high dimension and with noise
- **Encouraging results** on real-world applications
- Many opportunities in **developing kernels for particular applications**