Kernel methods in computational biology

Jean-Philippe Vert

Ecole des Mines de Paris, France
Jean-Philippe.Vert@mines.org

Universite Paul Sabatier, June 27, 2003

Outline

1. About kernels

2. What you can do with a kernel

3. Local alignment kernels for strings

4. *Analysis of microarray data with pathways information (if enough time)*

Part 1

Kernels

Definition

- Let $\mathcal{X}$ be a set (e.g., discrete)

- A kernel is a mapping $K : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ which is:

  ⋆ symetric : $K(x, y) = K(y, x)$,

  ⋆ 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 $x_i \in \mathcal{X}$

Example

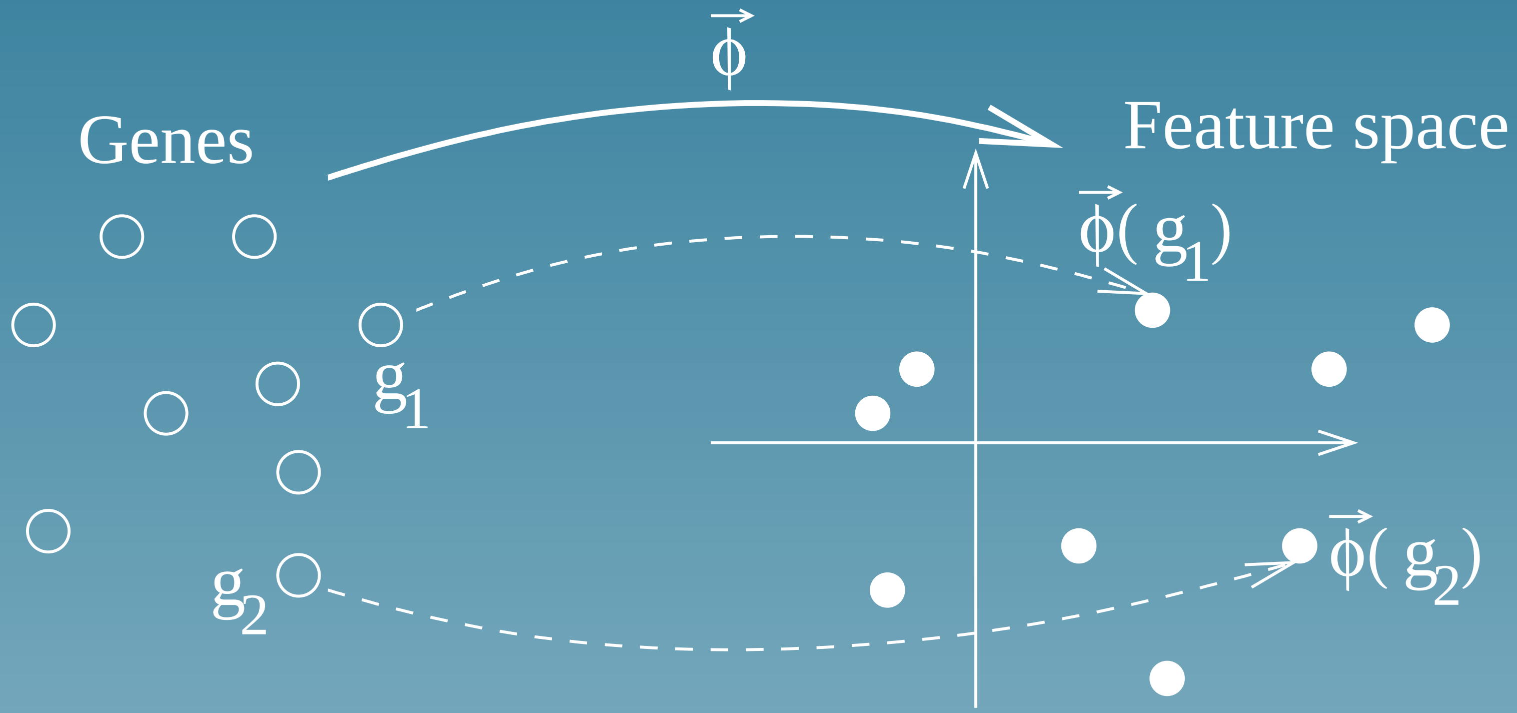
- Suppose $\mathcal{X} = \mathbb{R}^d$. Then the following is a valid kernel:

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

- Indeed:
  - ⋆ $\vec{x}.\vec{y} = \vec{y}.\vec{x}$
  - ⋆ $\sum_{i,j} a_i a_j \vec{x_i}.\vec{x_j} = ||\sum_i a_i \vec{x_i}||^2 \geq 0$

Example: kernel in feature space

$$K(g_i, g_j) \overset{def}{=} \vec{\Phi}(g_i).\vec{\Phi}(g_j)$$



All kernels are inner product

- If $K(.,.)$ is a kernel, then there exists a Hilbert space $\mathcal{H}$ and a mapping $\Phi : \mathcal{X} \to \mathcal{H}$ such that:

$$K(x,y) =< \Phi(x), \Phi(y) >_{\mathcal{H}} .$$

- Proof: by diagonalizing the kernel operator

- Second proof: by explicitly constructing such a $\mathcal{H}$

RKHS

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

$$span\left\{K(x,.), s \in \mathcal{X}\right\}.$$

- The inner product between two elements $f = \sum_i a_i K(x_i,.)$ and $g = \sum_i b_i K(x_i,.)$ is defined by:

$$< f, g >_{\mathcal{H}} = \sum_{i,j} a_i b_j K(x_i, x_i).$$

RKHS (2)

- Let $\Phi : \mathcal{X} \rightarrow \mathcal{H}$ defined by $\Phi(x) = K(x, .)$. Then:

$$K(x, y) =< \Phi(x), \Phi(y) >_{\mathcal{H}} =< K(x, .), K(y, .) >_{\mathcal{H}}$$

- For any $x \in \mathcal{X}$ and $f \in \mathcal{H}$, the following holds:

$$< f, K(x, .) >_{\mathcal{H}} = f(x).$$

RKHS (3)

- We have seen that a kernel K defines a Hilbert structure on (a subset of) $\mathcal{X}^{\mathbb{R}}$

- Conversely: let $\mathcal{H}$ be a Hilbert space, subset of $\mathcal{X}^{\mathbb{R}}$, such that for any $x \in \mathcal{X}$ the evaluation functional $f \in \mathcal{H} \rightarrow f(x)$ be continuous

- Then there exists a kernel K such that $\mathcal{H}$ be its associated RKHS.

Representer theorem (Wahba, 1971)

Let $\mathcal{H}$ be a RKHS with kernel K, and $(x_1, \ldots, x_N) \in \mathcal{X}^N$. Then the solution of:

$$\min_{f \in \mathcal{H}} \sum_{i=1}^{N} c(x_i, f(x_i)) + \lambda ||f||^2_{\mathcal{H}}$$

where $c : \mathcal{X} \times \mathbb{R} \to \mathbb{R}$, can always be written in the form:

$$f(x) = \sum_{i=1}^{n} a_i K(x_i, x).$$

Example

For a Gaussian kernel:

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

the norm in RKHS is:

$$||f||^2_{\mathcal{H}} = \frac{1}{2\pi\sigma^2}\int |\hat{f}(\omega)|^2 \exp\left(\frac{\sigma^2||\omega||^2}{2}\right) d\omega.$$

Partie 2

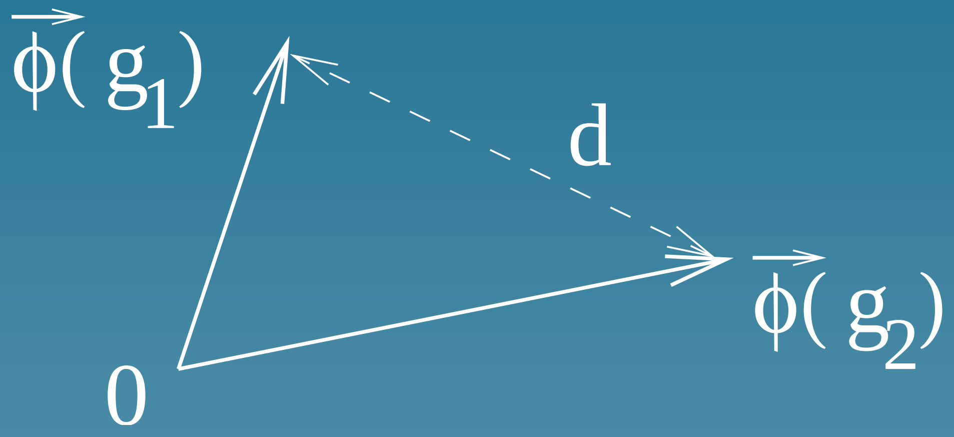
What can you do with a kernel

Overview

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

- Compute the distance between points
- Principal component analysis (PCA)
- Canonical correlation analysis (CCA)
- Classification by Support vector machines (SVM)

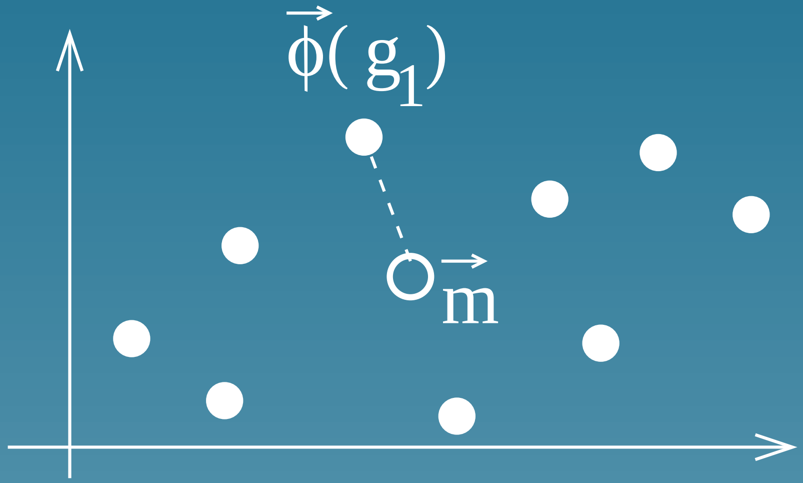
Compute the distance between objects



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

$$d(g_1,g_2)^2 = K(g_1,g_1) + K(g_2,g_2) - 2K(g_1,g_2)$$

Distance to the center of mass

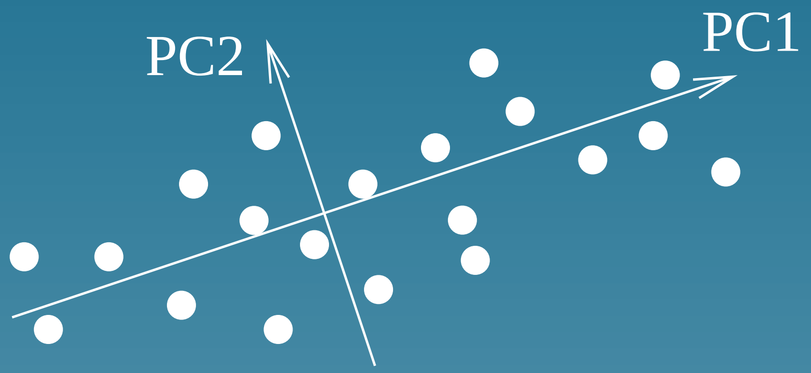


Center of mass: $\vec{m} = \frac{1}{N}\sum_{i=1}^{N}\vec{\Phi}(g_i)$, hence:

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

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

Principal component analysis

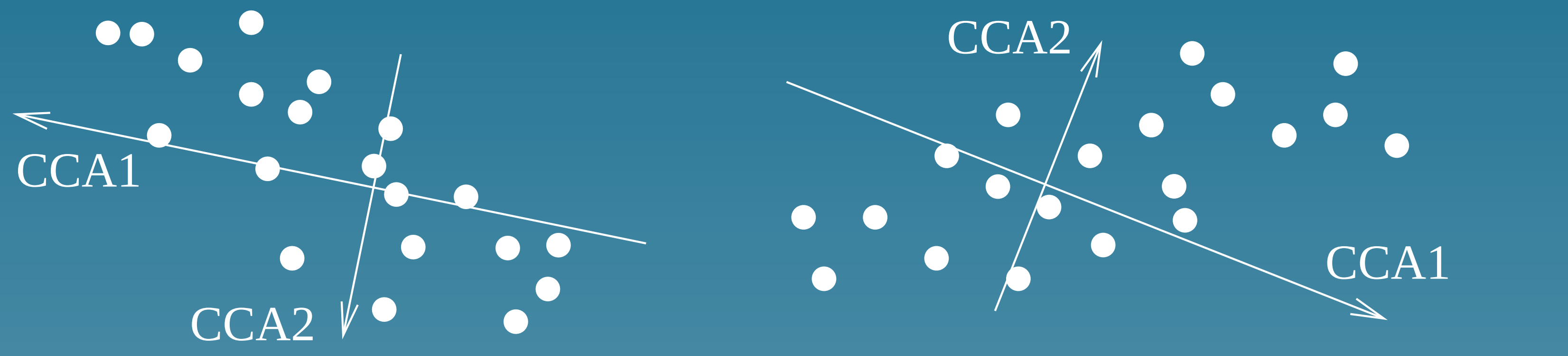


It is equivalent to find the eigenvectors of

$$K = \left(\vec{\Phi}(g_i).\vec{\Phi}(g_j)\right)_{i,j=1\ldots N}$$
$$= \left(K(g_i, g_j)\right)_{i,j=1\ldots N}$$

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

Canonical correlation analysis

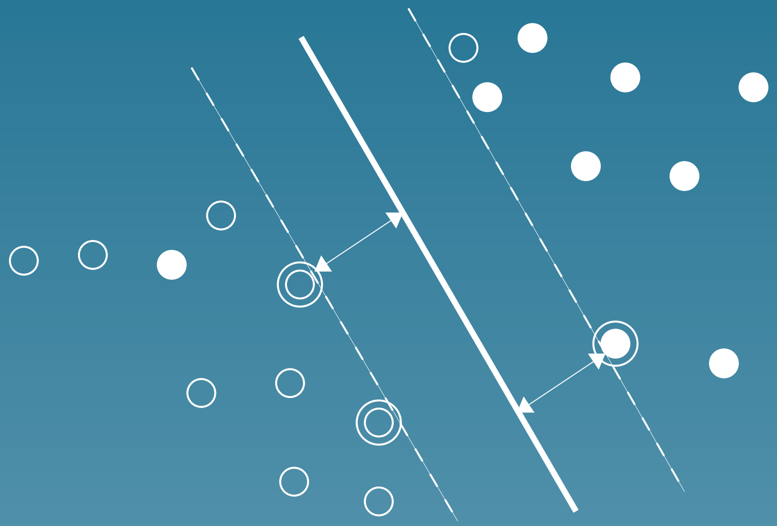


K_1 and K_2 are two kernels for the same objects. CCA can be performed by solving the following generalized eigenvalue problem:

$$\begin{pmatrix} 0 & K_1K_2 \\ K_2K_1 & 0 \end{pmatrix} \vec{\xi} = \rho \begin{pmatrix} K_1^2 & 0 \\ 0 & K_2^2 \end{pmatrix} \vec{\xi}$$

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

Classification: support vector machines (SVM)



Find a linear boundary with large margin and few errors

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

Examples: SVM in bioinformatics

- Gene functional classification from microarry: 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: Jaakkoola et al. (1998)
- Protein secondary structure prediction: Hua et al. (2001)
- Protein subcellular localization prediction from sequence: Hua et al. (2001)

Summary

- Once a kernel $K(x, y)$ is given, several analysis can be performed implicitly in the feature space

- These methods are considered currently among the most powerful on many real-world problems

- Modularity: each kernel can work with each method

Part 3

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.):

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

Related work

- Spectrum kernel (Leslie et al.):

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

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

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

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:

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

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:

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

3 basic kernels

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

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

- For gaps:

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

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

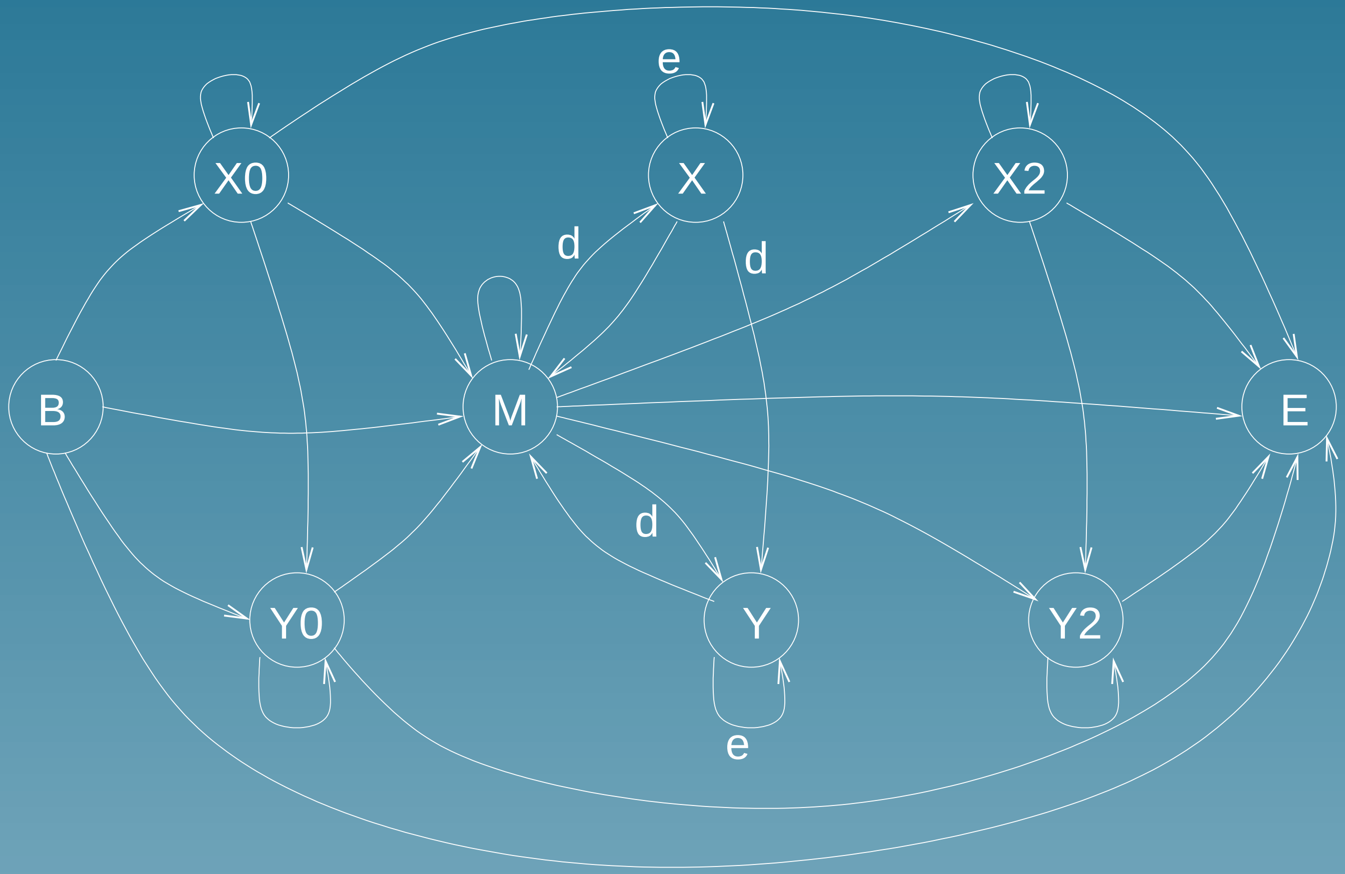
$$K_{LA}^{(\beta)}(x,y) = \sum_{\pi \in \Pi(x,y)} \exp\left(\beta s(x,y,\pi)\right),$$

Properties

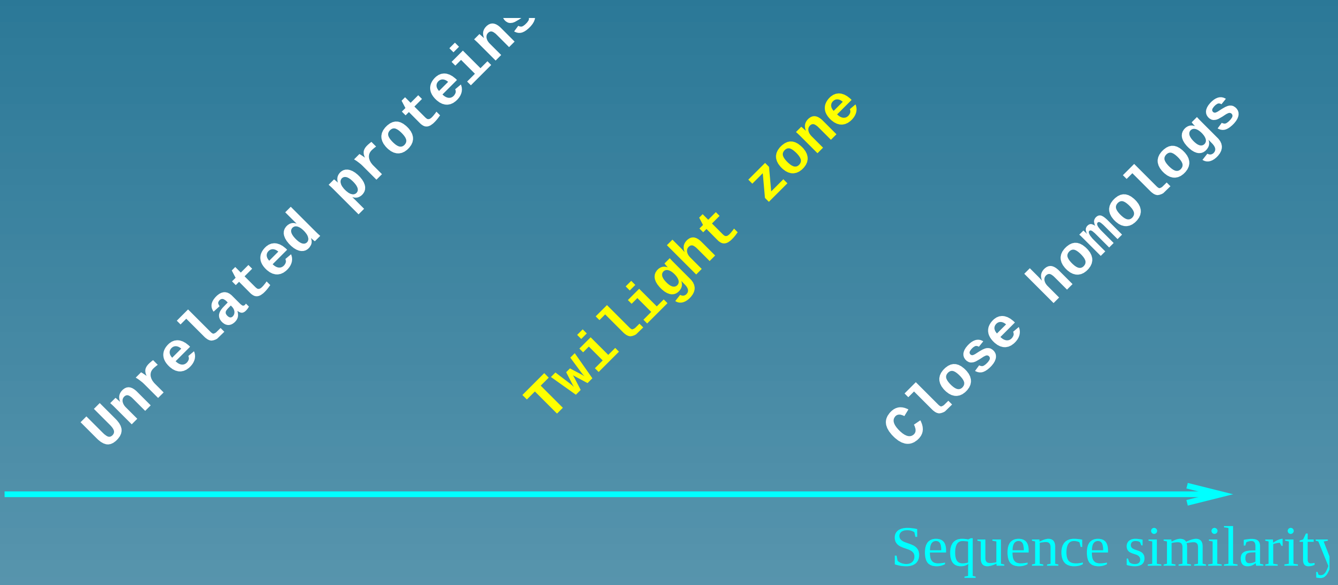
$$K_{LA}^{(\beta)}(x,y) = \sum_{\pi \in \Pi(x,y)} \exp\left(\beta s(x,y,\pi)\right),$$

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

Kernel computation



Application: remote homology detection



Unrelated proteins

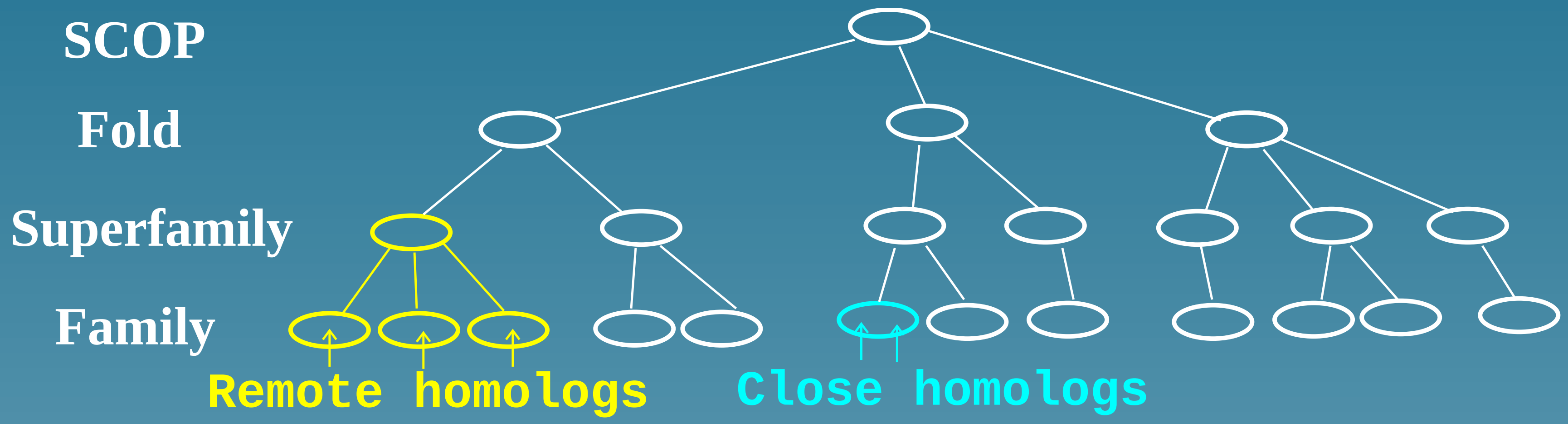
Twilight zone

Close homologs

Sequence similarity

- 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

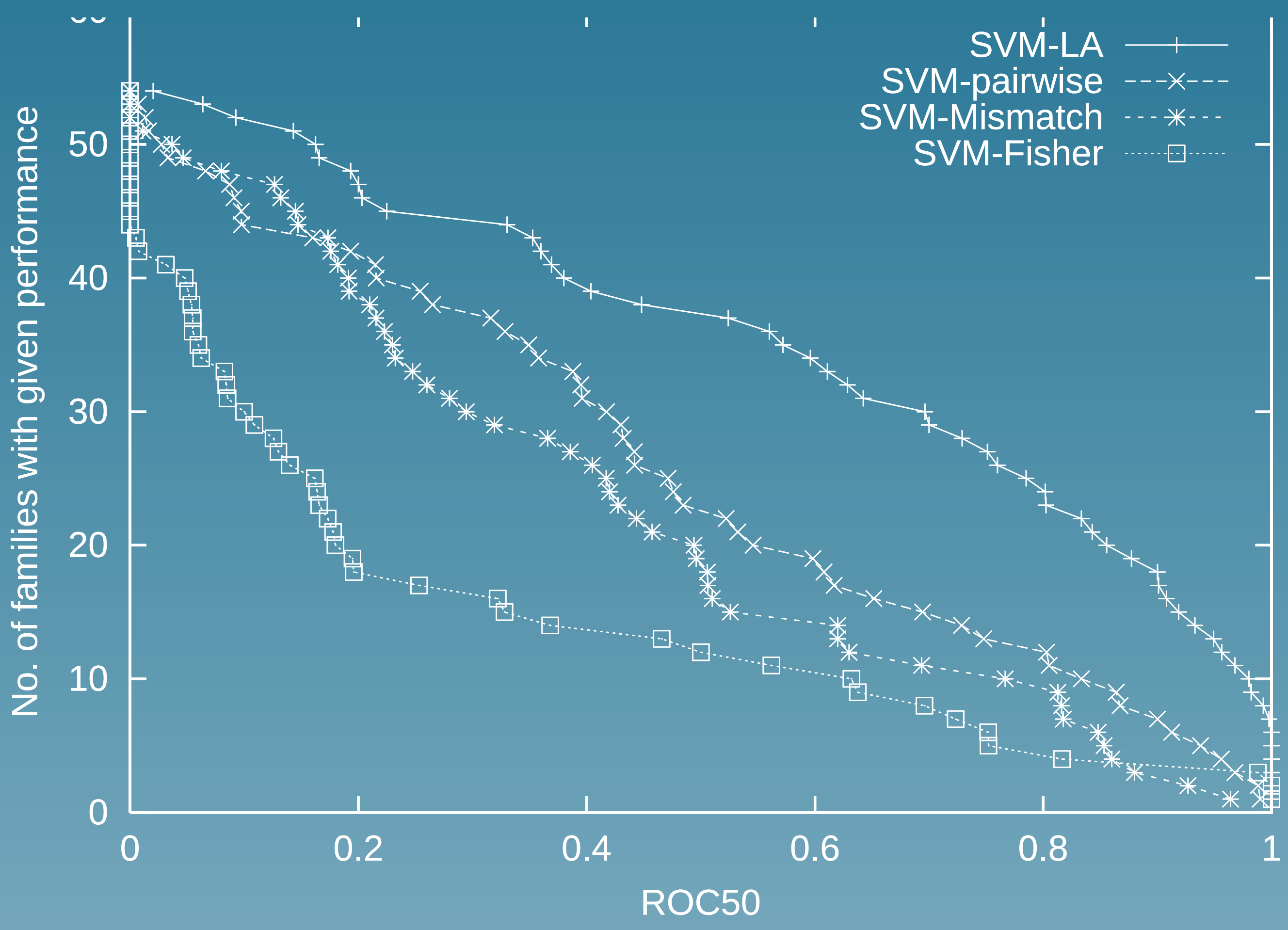
- Can we predict the superfamily of a domain if we have not seen any member of its family before?

- 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

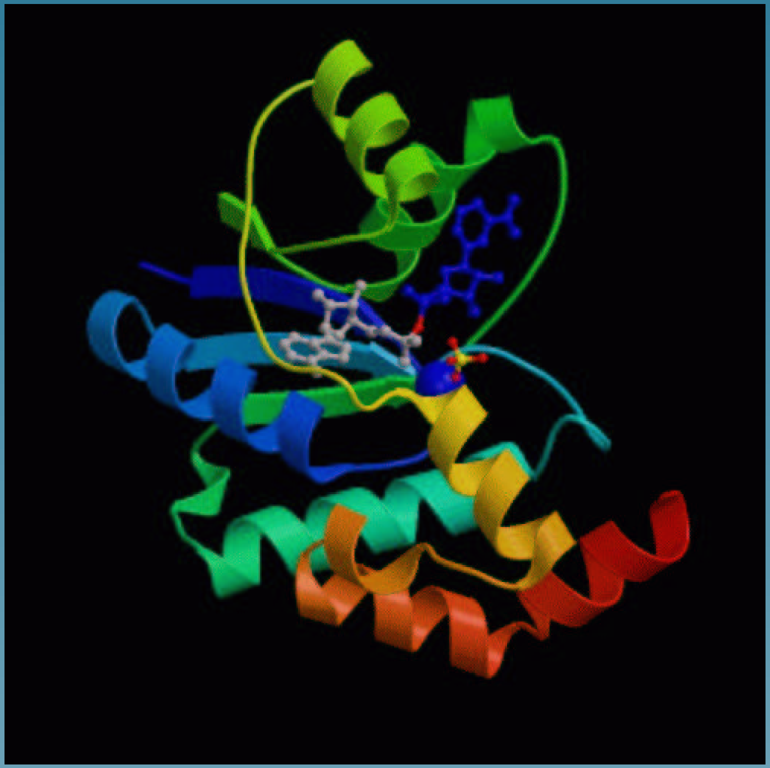
SCOP superfamily recognition benchmark



Part 4

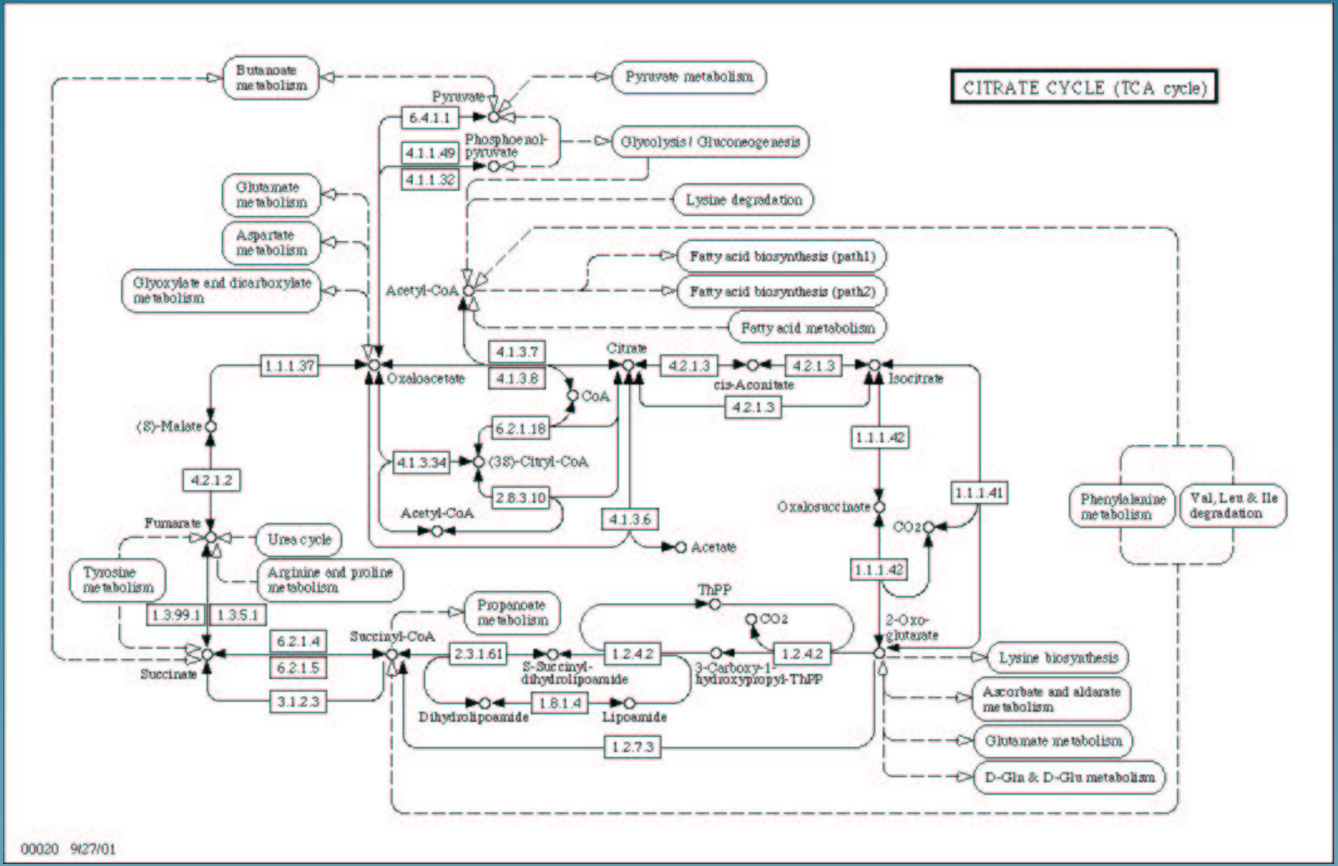
Analysis of microarray data with pathways information

Genes encode proteins which can catalyse chemical reations



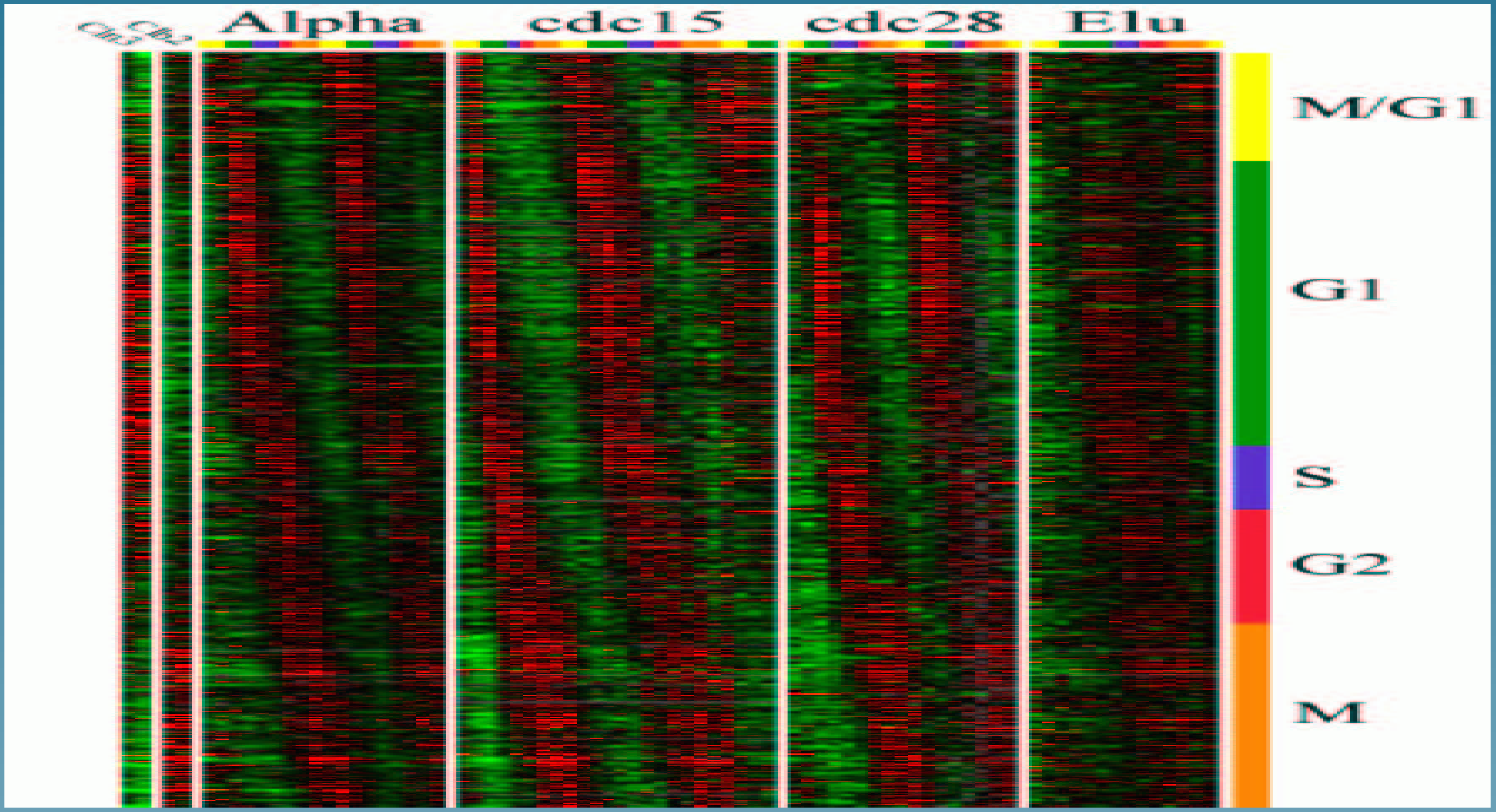
Nicotinamide Mononucleotide Adenylyltransferase With Bound Nad+

Chemical reactions are often parts of pathways



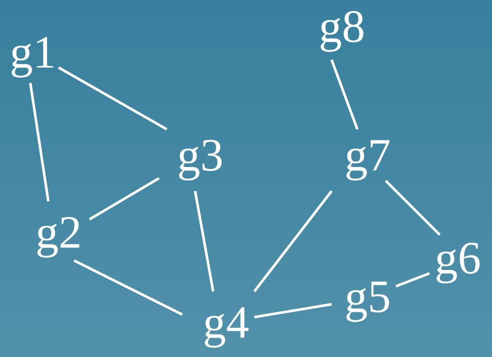
From http://www.genome.ad.jp/kegg/pathway

Microarray technology monitors RNA quantity



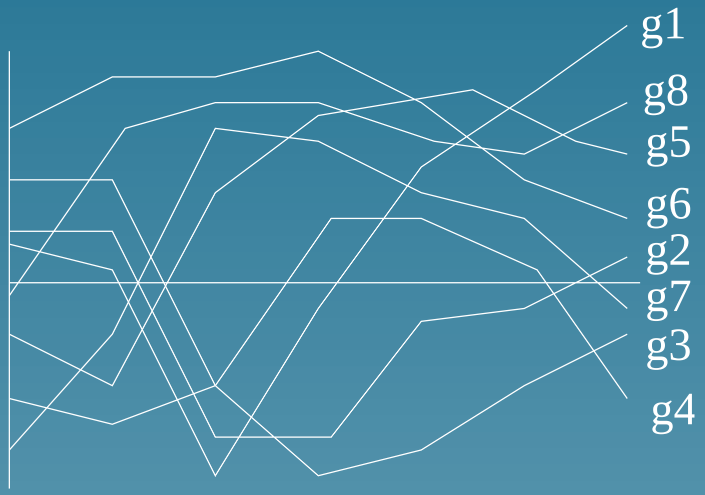
(From Spellman et al., 1998)

Comparing gene expression and protein network



Gene network

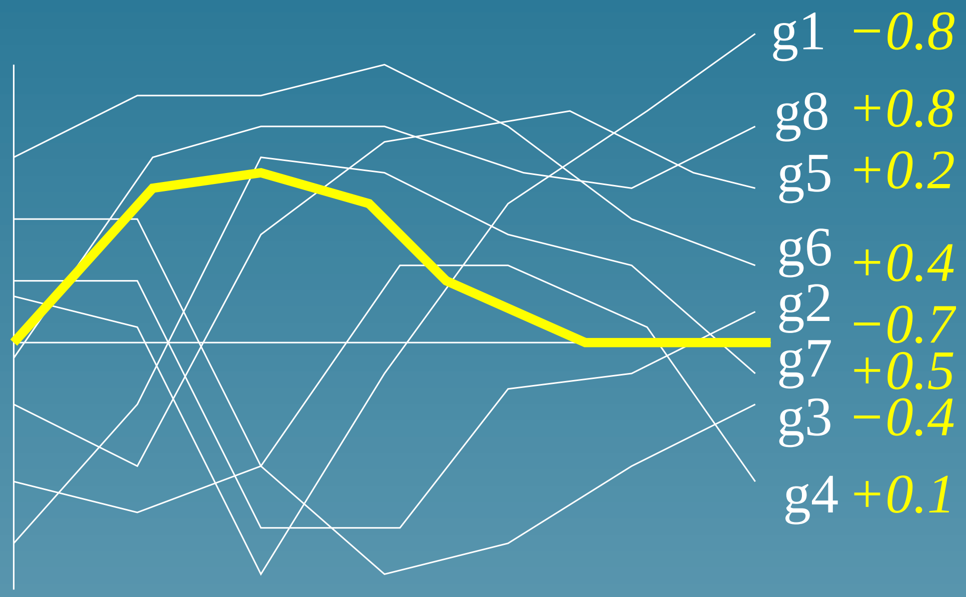
⟺



Expression profiles

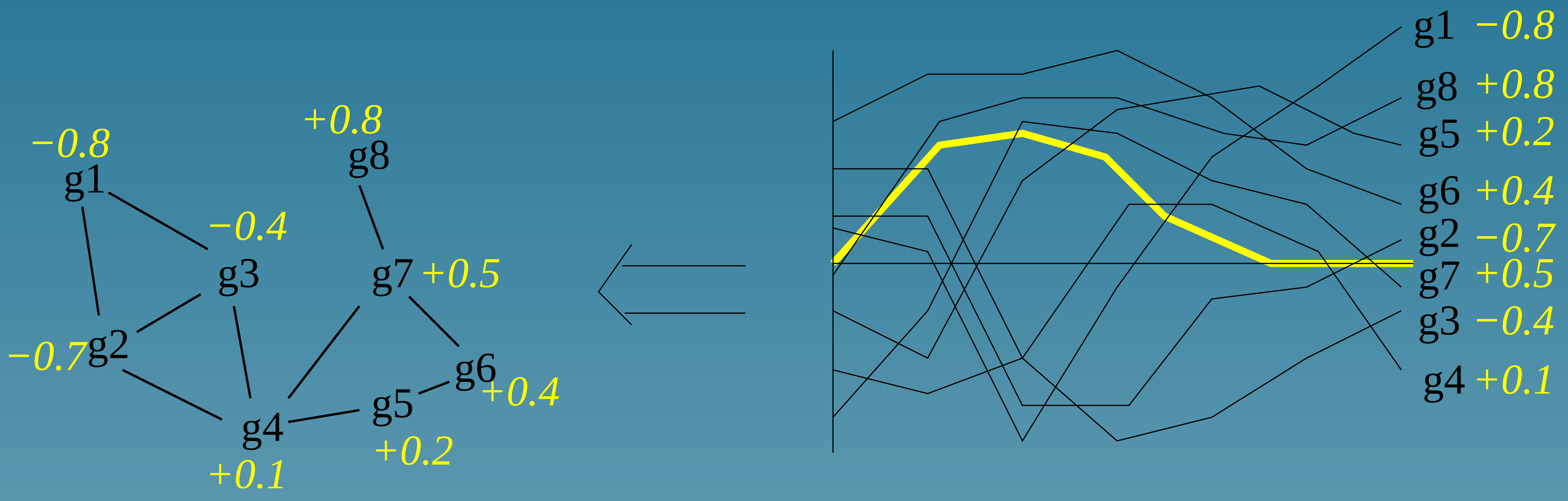
Are there “correlations”?

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

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 $||.||_{H_1}$ 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 $||.||_{H_2}$ 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 :

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

be large

Problem reformulation (2)

The three goals can be combined in the following problem:

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

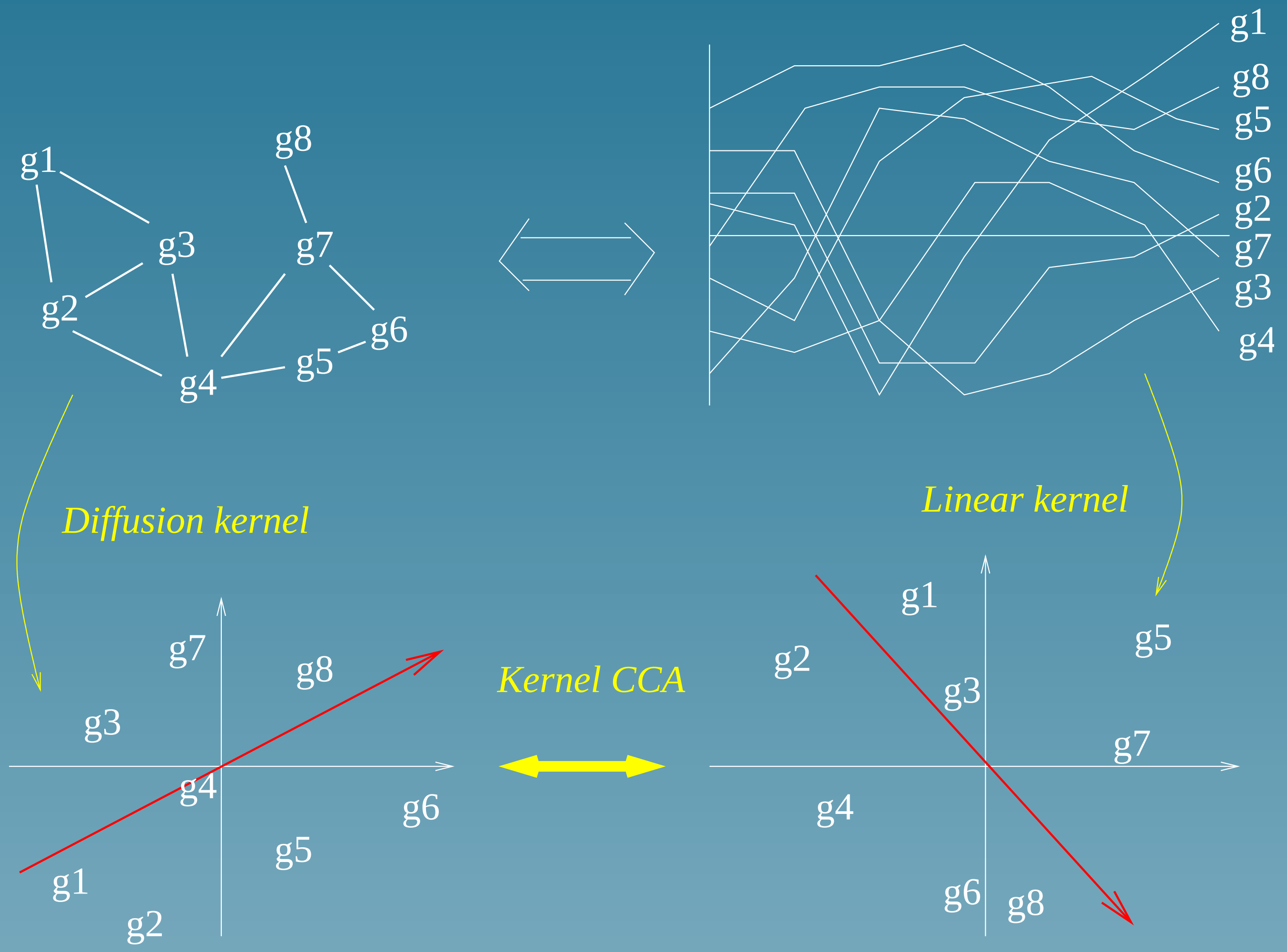
where the parameter δ controls the trade-off between relevance/smoothness on the one hand, correlation on the other hand.

Solving the problem

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

$$\begin{pmatrix} 0 & K_1K_2 \\ K_2K_1 & 0 \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix} = \rho\begin{pmatrix} K_1^2 + \delta K_1 & 0 \\ 0 & K_2^2 + \delta K_2 \end{pmatrix}\begin{pmatrix} \alpha \\ \beta \end{pmatrix}$$

Summary


Diffusion kernel

Linear kernel

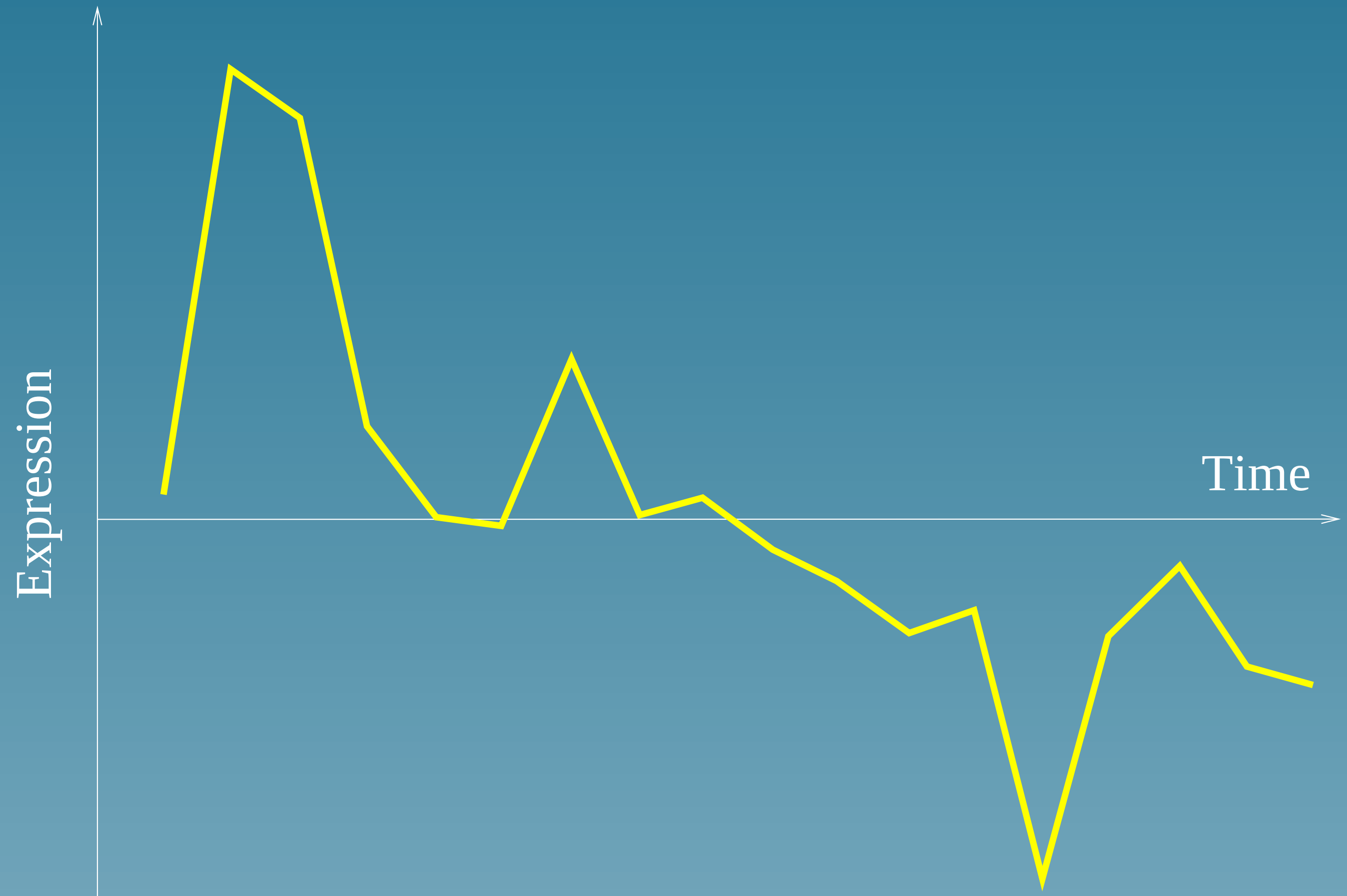
Kernel CCA

Data

- Gene network: two genes are linked if the 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

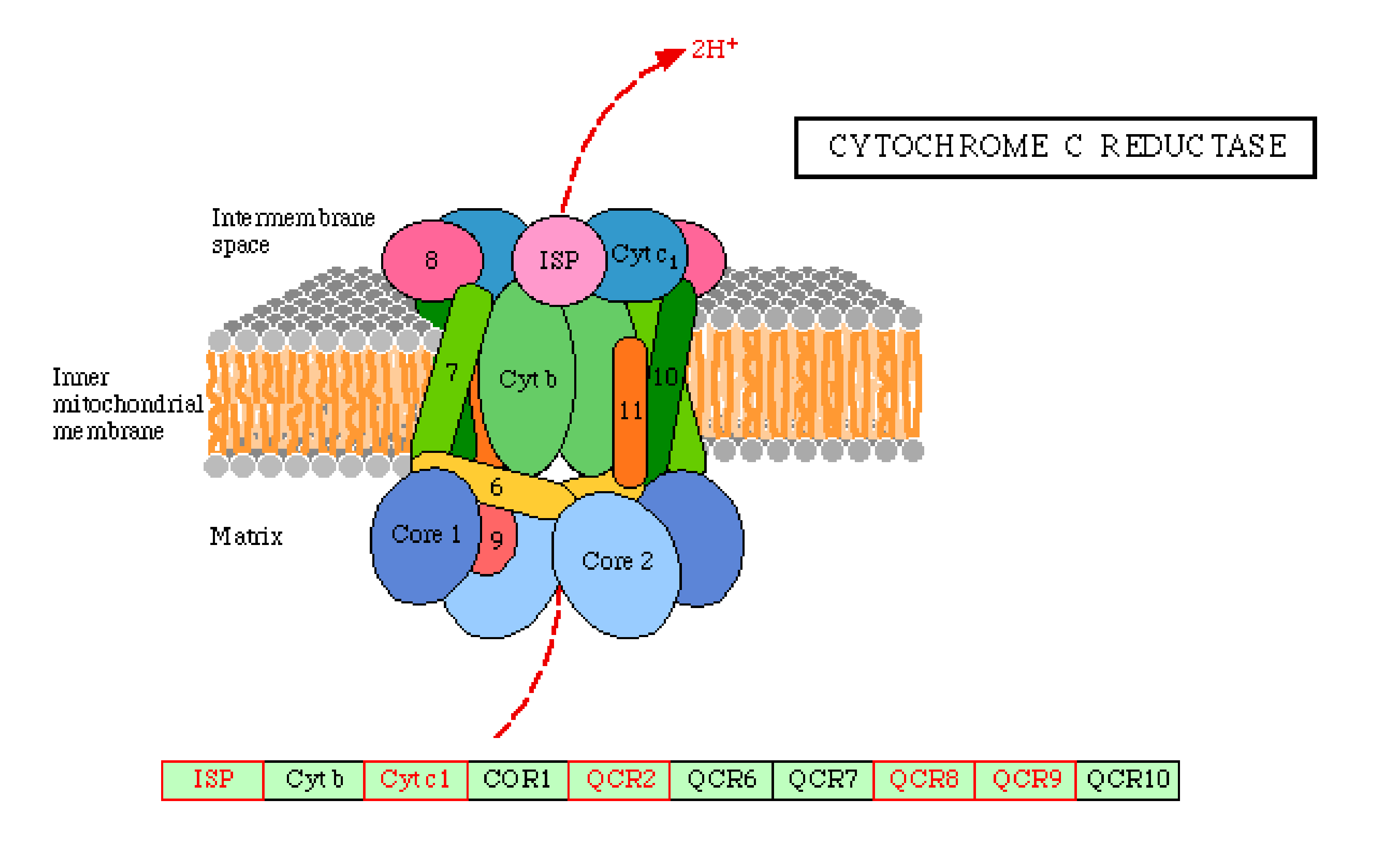


Related metabolic pathways

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

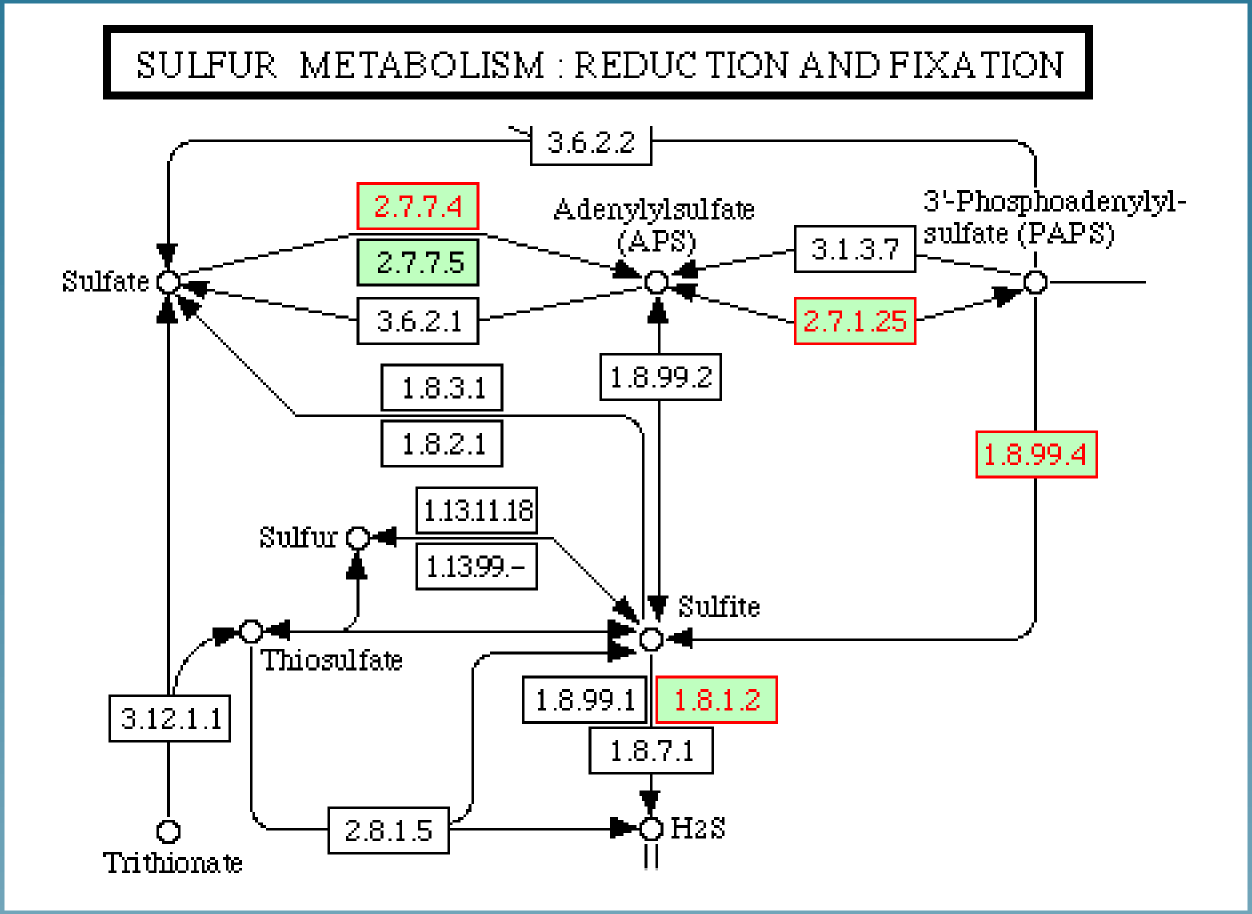
- Oxidative phosphorylation (10 genes)
- Citrate cycle (7)
- Purine metabolism (6)
- Glycerolipid metabolism (6)
- Sulfur metabolism (5)
- Selenoaminoacid metabolism (4) , etc...

Related genes

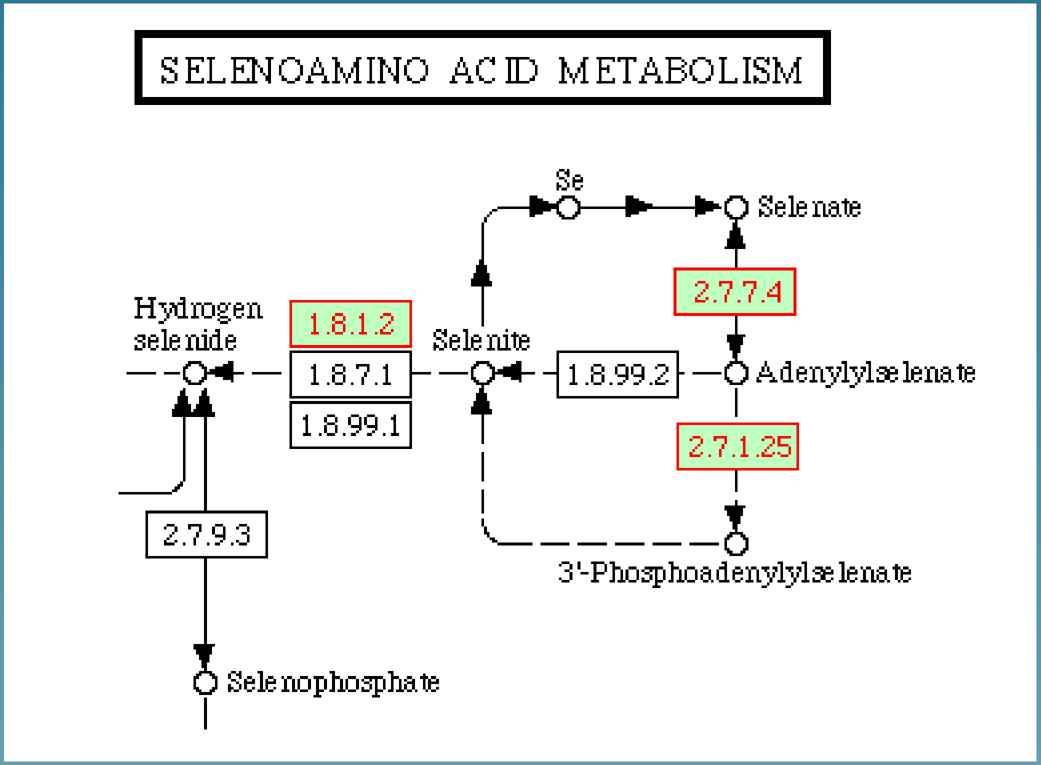


ISP	Cyt b	Cyt c1	COR1	QCR2	QCR6	QCR7	QCR8	QCR9	QCR10

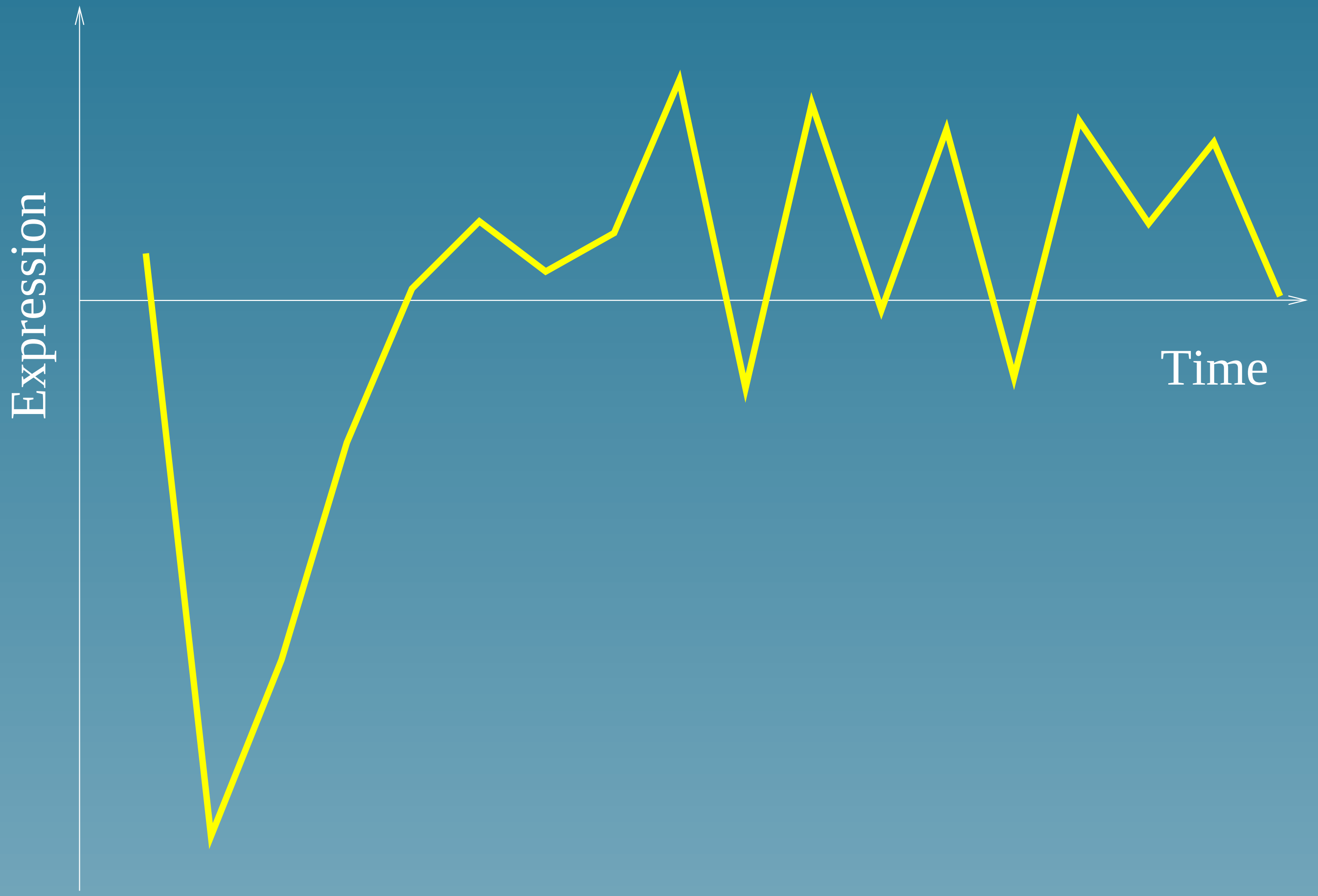
Related genes



Related genes



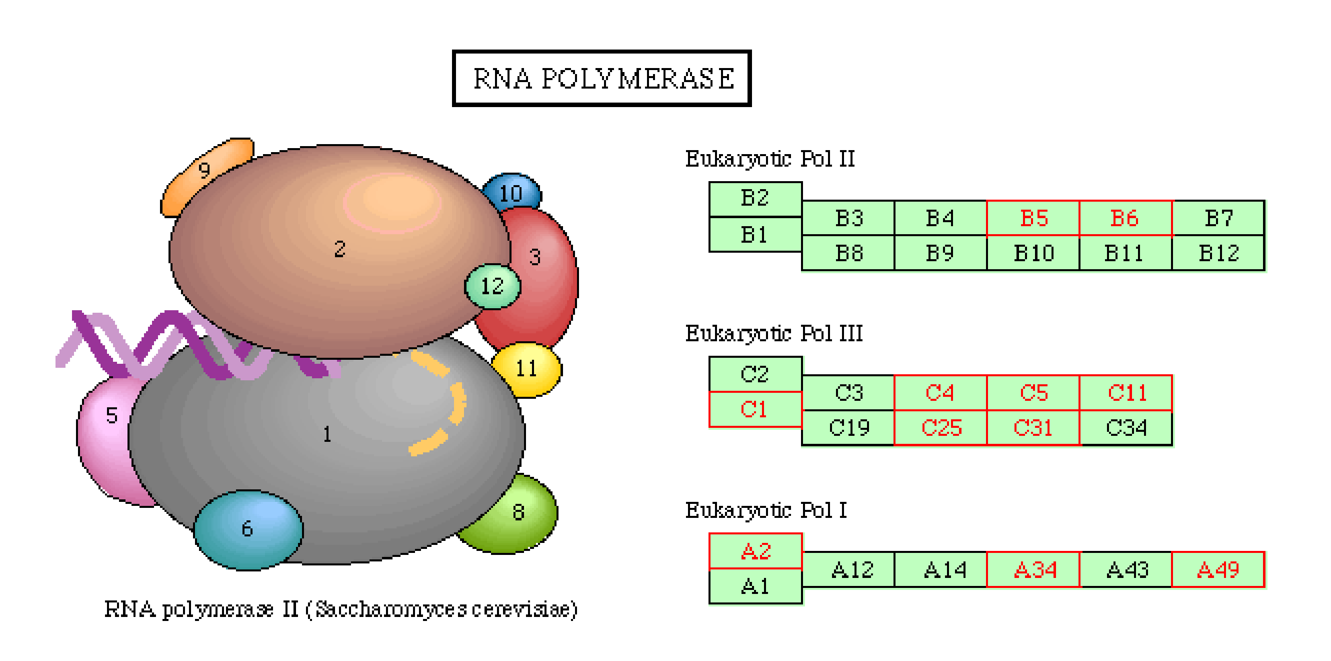
Opposite pattern



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

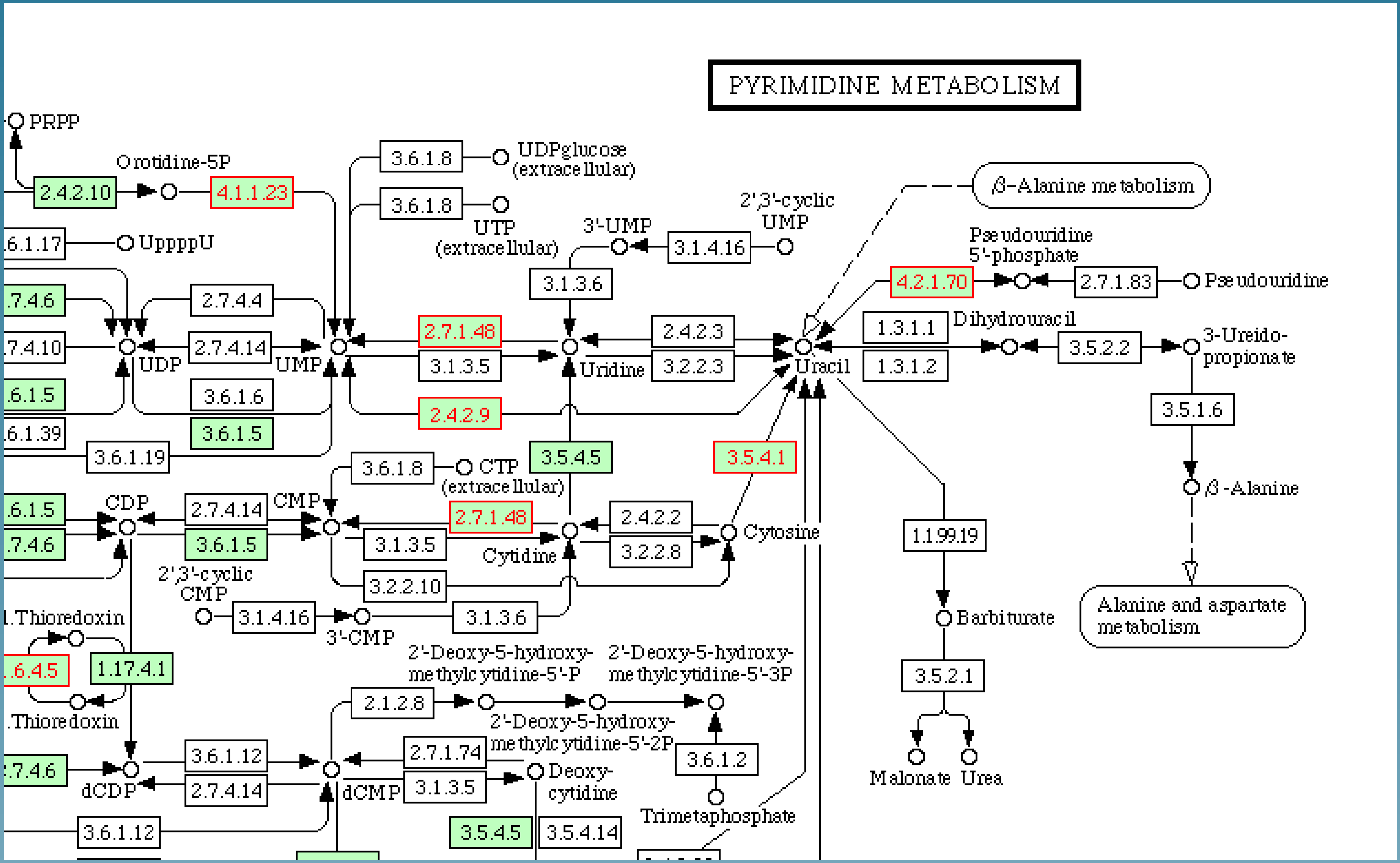
Eukaryotic Pol III

C2	C3	C4	C5	C11
C1	C19	C25	C31	C34

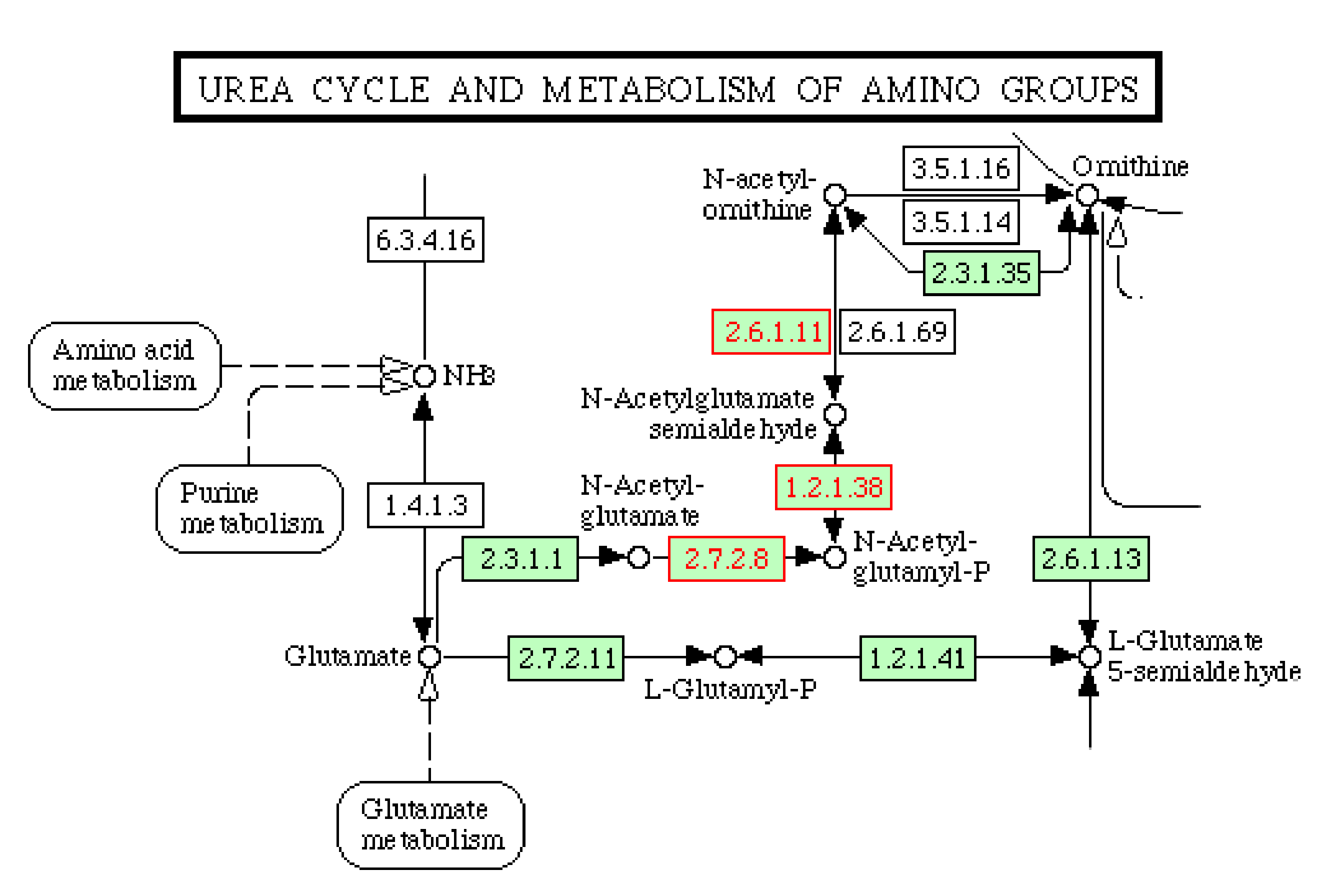
Eukaryotic Pol I

A2	A12	A14	A34	A43	A49
A1					

Related genes



Related genes



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

- SVM and kernel methods work well on real-life problems, in particular in high dimension and with noise
- Kernels can be engineered for non-vectorial data
- Kernels povides a general framework to integrate heterogeneous data