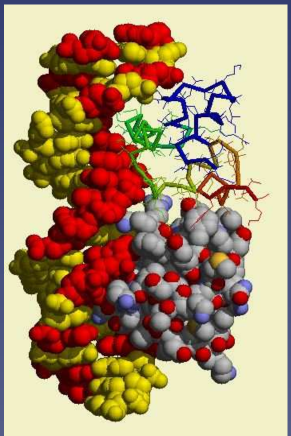
Kernel methods in computational and systems biology

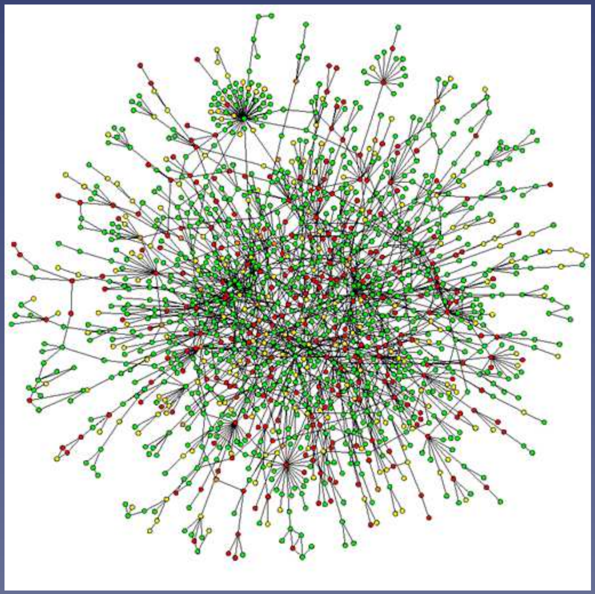
Jean-Philippe.Vert@mines.org

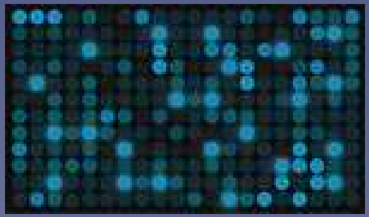
Ecole des Mines de Paris, France
Computational biology

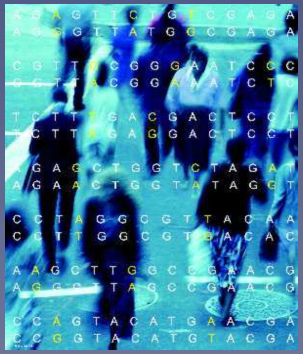
"Perspectives in Computational and Theoretical Biology" symposium, Shanghai, P.R. China, Dec. 19-20, 2004.

The age of data in biology









And many more...

Motivations

Develop a theoretical framework and algorithms in order to

- represent and integrate biological data
- model and conceptualize living systems
- infer properties of living systems

Biological data are often

- structured and heterogeneous : sequences, 3D structures, graphs, networks, expression profiles, phylogenetic trees, SNP, ...

Biological data are often

- structured and heterogeneous : sequences, 3D structures, graphs, networks, expression profiles, phylogenetic trees, SNP, ...

- in large quantities $(10^6$ gene sequences)

Biological data are often

- structured and heterogeneous : sequences, 3D structures, graphs, networks, expression profiles, phylogenetic trees, SNP, ...
- in large quantities $(10^6$ gene sequences)
- in large dimension $(10^5 \sim 10^6$ spots on DNA chips)

A possible solution: kernel methods

Kernel methods (partially) overcome these issues:

- Kernels for structured data

A possible solution: kernel methods

Kernel methods (partially) overcome these issues:

- Kernels for structured data
- Operations on kernels to integrate heterogeneous data

A possible solution: kernel methods

Kernel methods (partially) overcome these issues:

- Kernels for structured data
- Operations on kernels to integrate heterogeneous data
- Regularisation in order to deal with large dimensions

A possible solution: kernel methods

Kernel methods (partially) overcome these issues:

- Kernels for structured data
- Operations on kernels to integrate heterogeneous data
- Regularisation in order to deal with large dimensions
- Statistical framework for the processing of large datasets

What is a kernel?

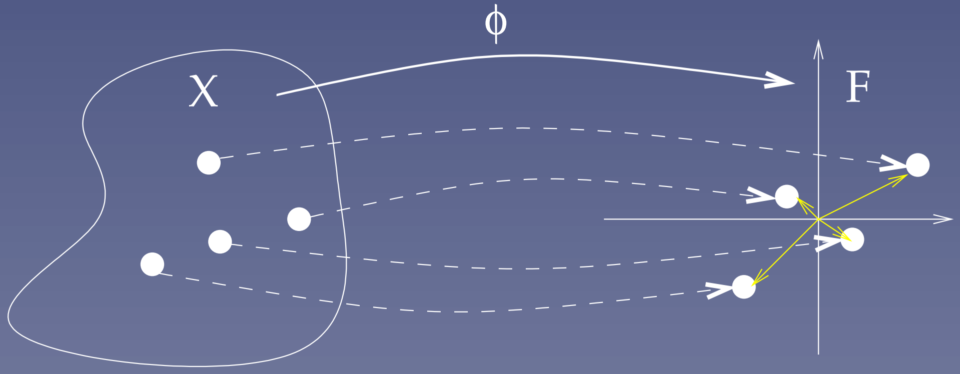
- Let $\mathcal{X}$ be a set to be analyzed (e.g., gene sequences or protein structures)
- A kernel on $\mathcal{X}$ is a measure of similarity $K(x, x')$ between elements of $\mathcal{X}$ (that is symmetric and positive definite).
- Example: a kernel for finite-length sequences

$$K(aatcga, cgaagtagccc) = 0.4$$

Geometric interpretation as inner product

If K is a kernel on $\mathcal{X}$, then $\mathcal{X}$ can be mapped to a Hilbert space $\mathcal{H}$ through $\Phi : \mathcal{X} \rightarrow \mathcal{H}$ in such a way that:

$$\forall \left(\mathbf{x}, \mathbf{x}'\right) \in \mathcal{X}^2, \quad K\left(\mathbf{x}, \mathbf{x}'\right) = \left\langle \Phi\left(\mathbf{x}\right), \Phi\left(\mathbf{x}'\right)\right\rangle_{\mathcal{H}}.$$



Kernel trick

- Any algorithm for vectors that only involves inner products can be performed implicity inthe feature space by remplacing the inner product by a kernel

Kernel trick

- Any algorithm for vectors that only involves inner products can be performed implicity inthe feature space by remplacing the inner product by a kernel

- Example: Support Vector Machines (classification, regression), clustering, PCA, ICA, CCA, logistic regression..= kernel methods

Kernel trick

- Any algorithm for vectors that only involves inner products can be performed implicity inthe feature space by remplacing the inner product by a kernel

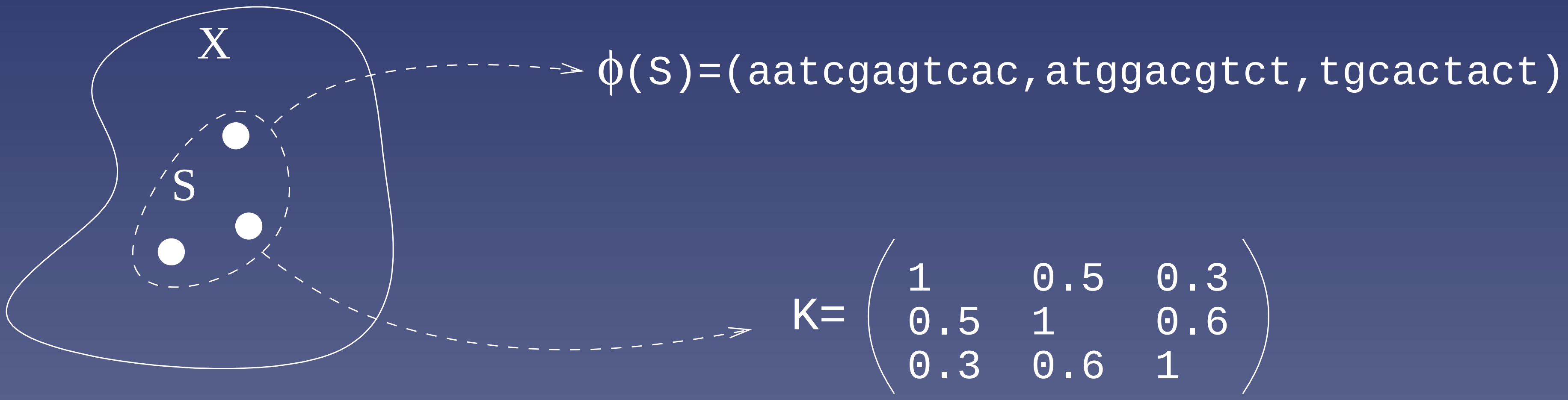
- Example: Support Vector Machines (classification, regression), clustering, PCA, ICA, CCA, logistic regression..= kernel methods

- “Simples kernels” can correspond to “complex” mappings

Kernel trick

- Any algorithm for vectors that only involves inner products can be performed implicity inthe feature space by remplacing the inner product by a kernel
- Example: Support Vector Machines (classification, regression), clustering, PCA, ICA, CCA, logistic regression..= kernel methods
- "Simples kernels" can correspond to "complex" mappings
- Objets are not necessarily vectors!

Data representation with kernels



ϕ(S)=(aatcgagtcac,atggacgtct,tgcactact)

$$K=\begin{pmatrix} 1 & 0.5 & 0.3 \\ 0.5 & 1 & 0.6 \\ 0.3 & 0.6 & 1 \end{pmatrix}$$

- Each data set is a matrix
- Kernel methods process these matrices

A few kernels for biological data

- Interpolated kernel for fixed-length sequences (*PSB'02*)

A few kernels for biological data

- Interpolated kernel for fixed-length sequences (*PSB'02*)
- Kernel for phylogenetic profiles (*ISMB'02*)

A few kernels for biological data

- Interpolated kernel for fixed-length sequences (*PSB'02*)
- Kernel for phylogenetic profiles (*ISMB'02*)
- Kernel for molecular 2D structures de molécules (*ICML'04*)

A few kernels for biological data

- Interpolated kernel for fixed-length sequences (*PSB'02*)
- Kernel for phylogenetic profiles (*ISMB'02*)
- Kernel for molecular 2D structures de molécules (*ICML'04*)
- Mutual information kernel for sequences (*IJCNN'04*)

A few kernels for biological data

- Interpolated kernel for fixed-length sequences (*PSB'02*)
- Kernel for phylogenetic profiles (*ISMB'02*)
- Kernel for molecular 2D structures de molécules (*ICML'04*)
- Mutual information kernel for sequences (*IJCNN'04*)
- Local alignment kernel for sequences (*Bioinformatics 04*)

A few kernels for biological data

- Interpolated kernel for fixed-length sequences (*PSB'02*)
- Kernel for phylogenetic profiles (*ISMB'02*)
- Kernel for molecular 2D structures de molécules (*ICML'04*)
- Mutual information kernel for sequences (*IJCNN'04*)
- Local alignment kernel for sequences (*Bioinformatics 04*)
- Kernel for sets of points (*NIPS'04*)

Applications

- Signal peptide detection in protein sequences

Applications

- Signal peptide detection in protein sequences
- Gene function prediction

Applications

- Signal peptide detection in protein sequences
- Gene function prediction
- Virtual screening of small molecules

Applications

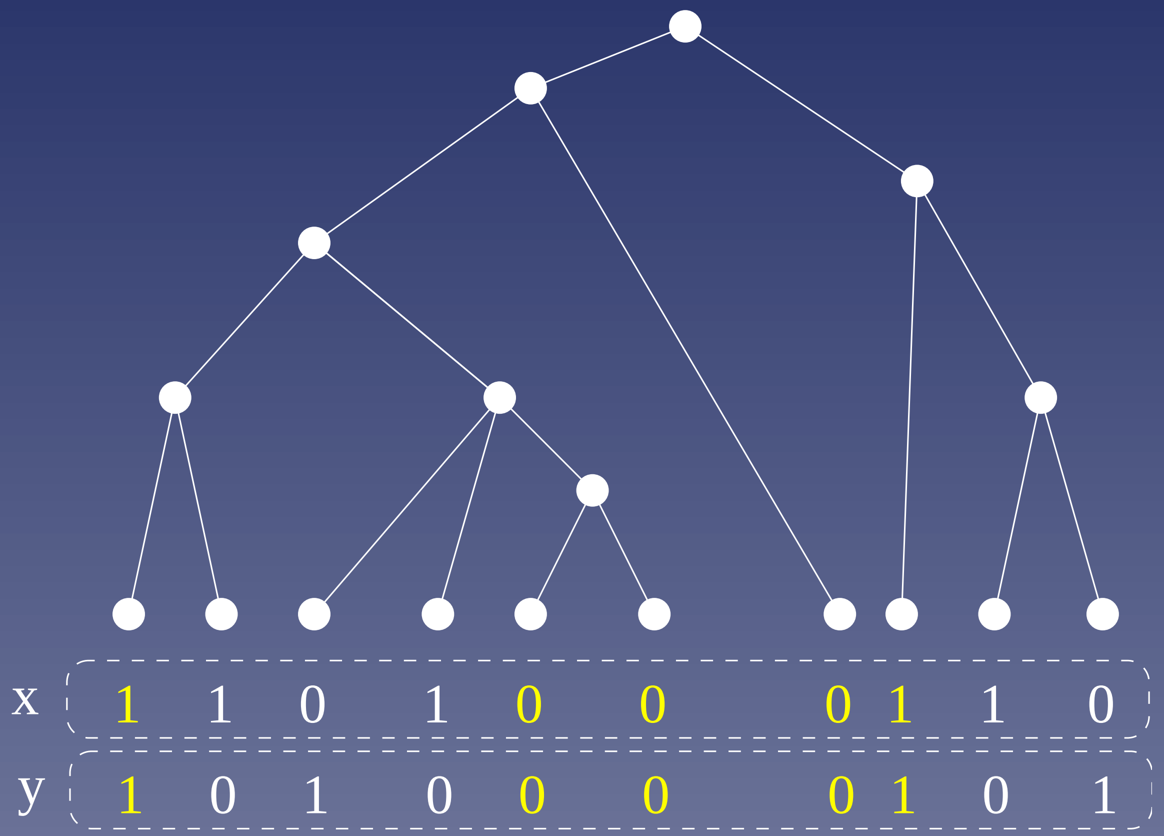
- Signal peptide detection in protein sequences
- Gene function prediction
- Virtual screening of small molecules
- Homology detection between gene sequences

Example 1: Phylogenetic profiles (*ISMB 02*)

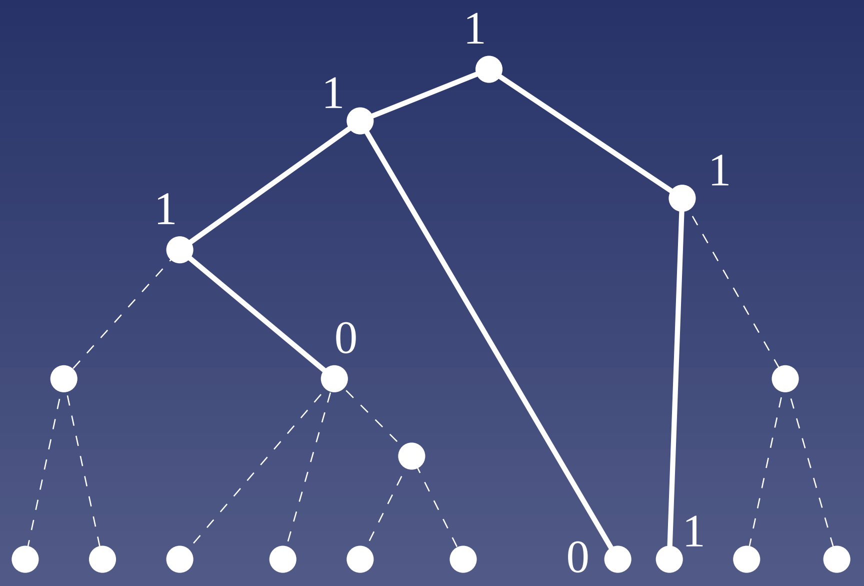
Gene	human	yeast	. . .	HIV	E. coli
YAL001C	1	1	. . .	0	0
YAB002W	0	0	. . .	0	1
⋮	⋮	⋮	⋮	⋮	⋮

- Computed *in silico*
- Useful to infer gene function

How to compare two profiles?



“Phylogenetic” kernel



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

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

Gene function prediction (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%*

Example 2: Local alignment kernel (*Bioinfo. 04*)

- The Smith-Waterman local alignment score:

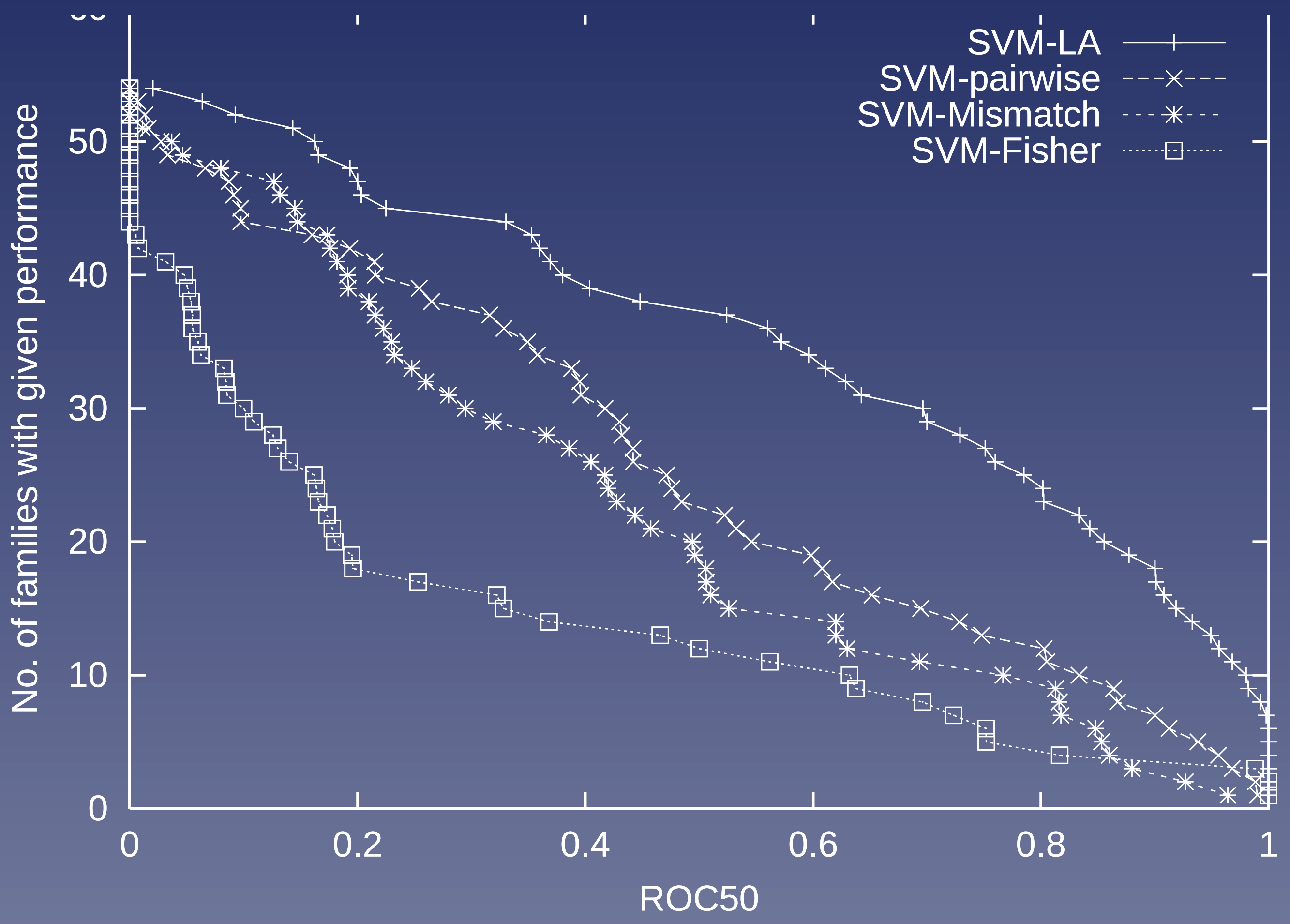
$$SW(x,y) = \max_{\pi \in \Pi(x,y)} s(x,y,\pi)$$

is a widely-used measure of similarity between biological sequences, but... it is not a kernel

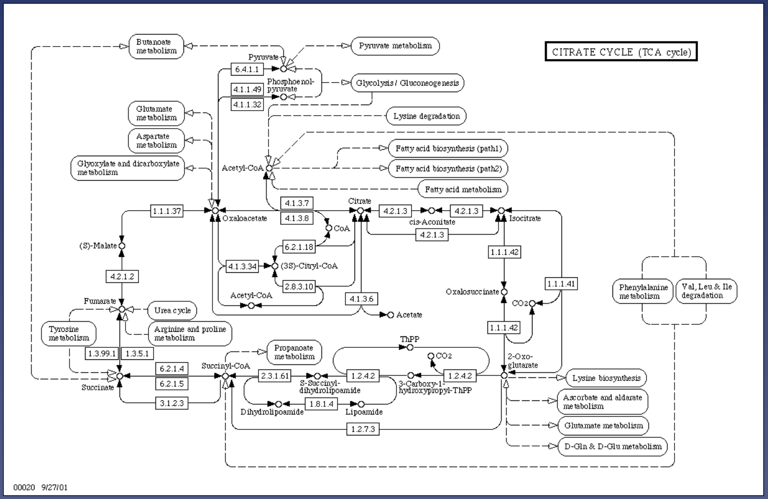
- The following local alignment kernel is valid:

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

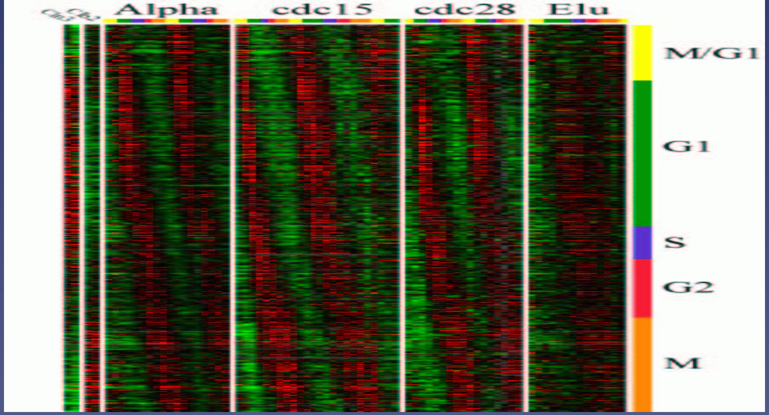
Empirical evaluation



Comparison of heterogeneous data (NIPS’02)



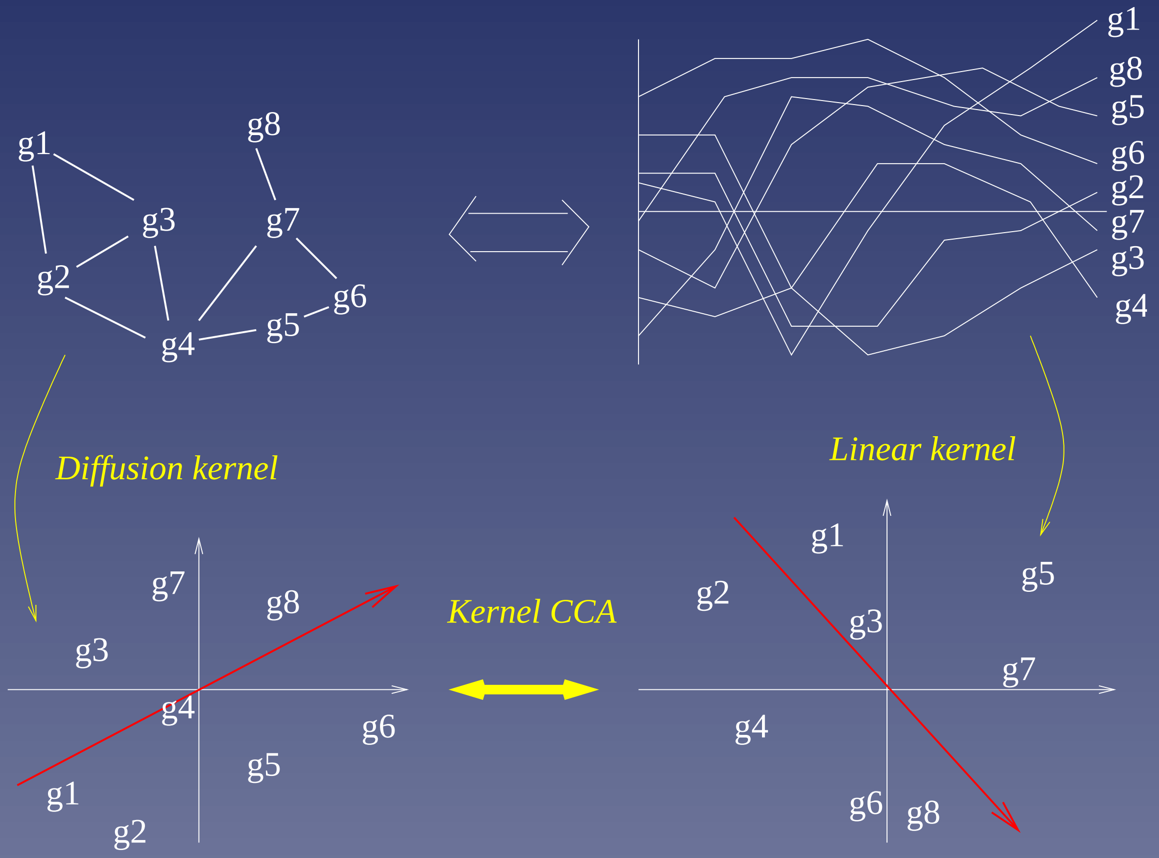
VS



Detecting pathway activity? Data and graph denoising?

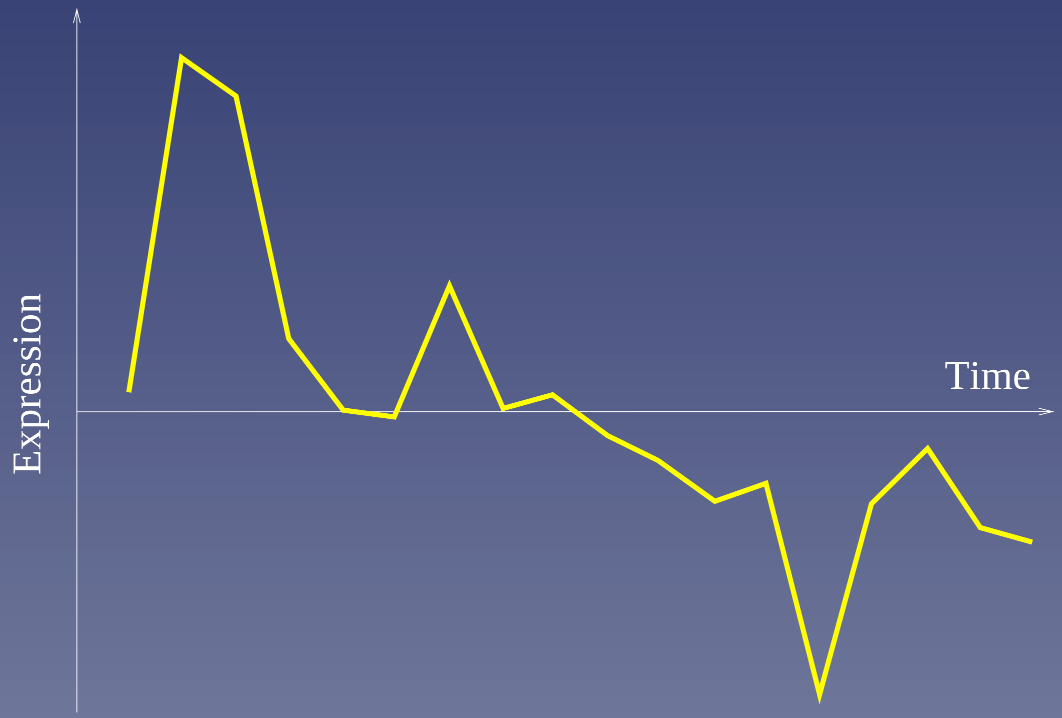
Network reconstruction?

Using kernel-CCA

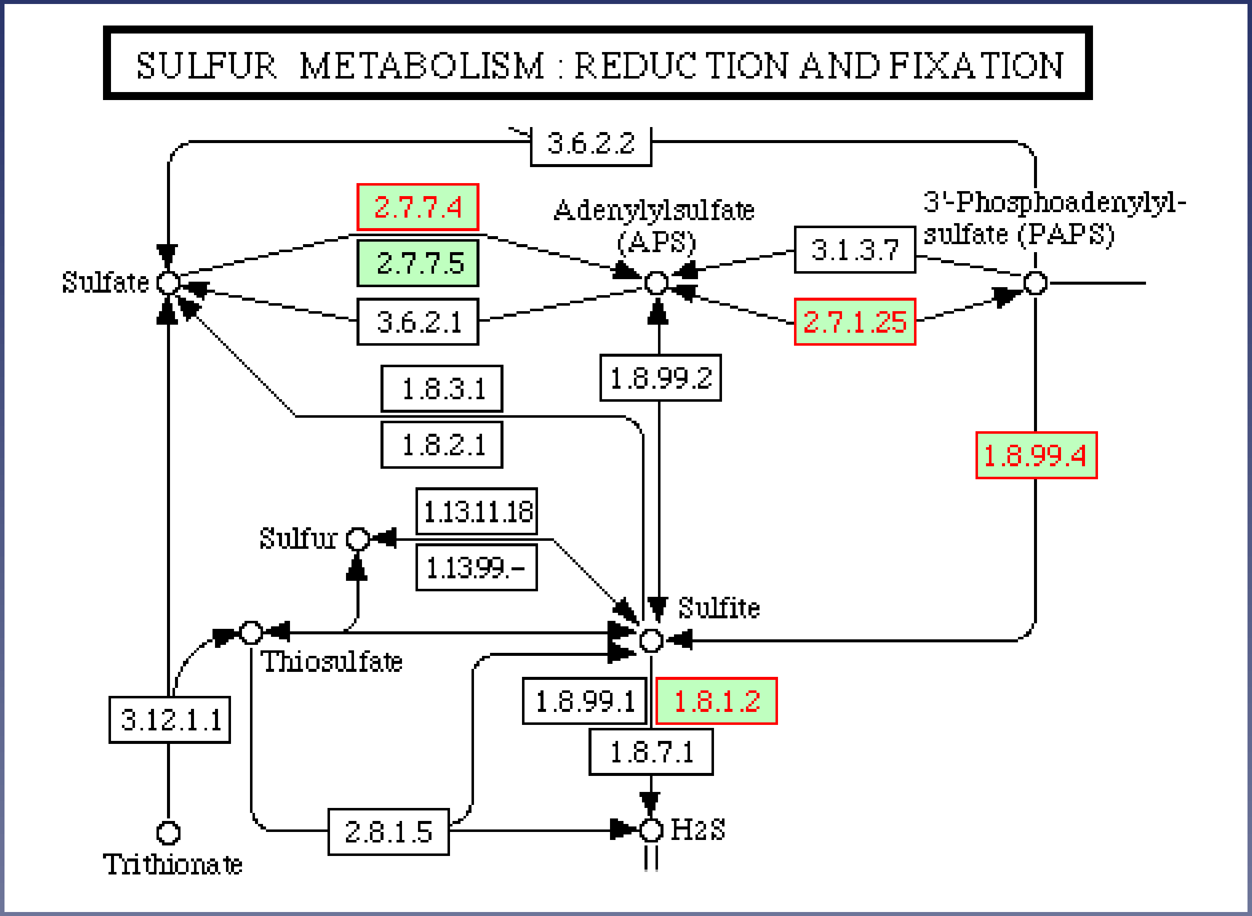


Example (*ECCB'03*)

Comparison of the metabolic network vs cell cycle gene expression in yeast



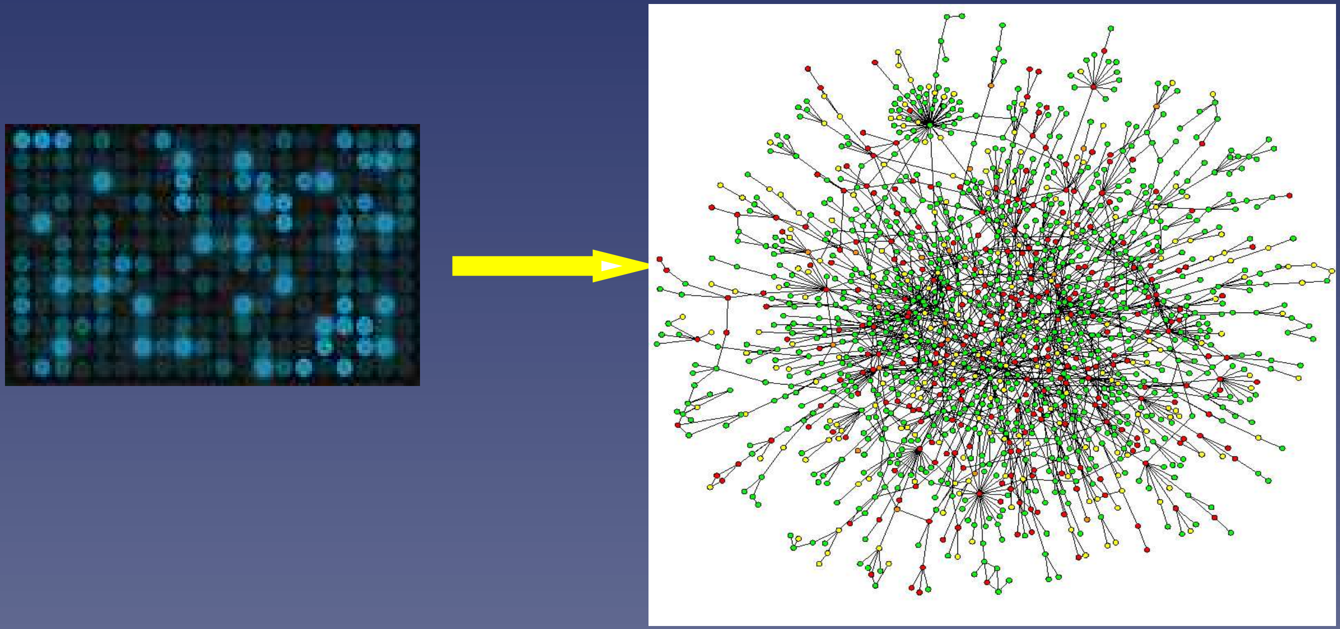
Correlated pathways



Extensions

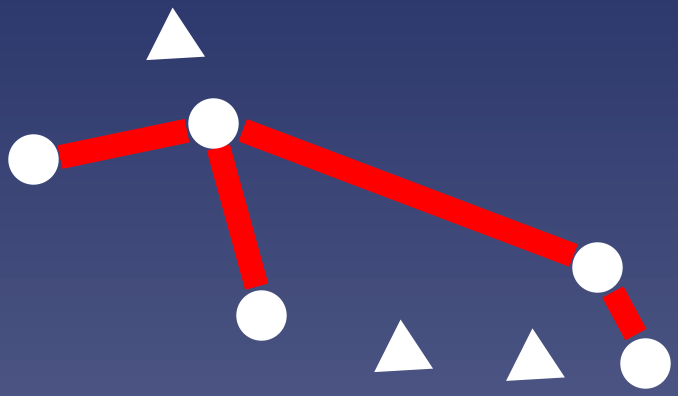
- Feature extraction for gene supervised classification (*NIPS'02*)
- Feature extraction for gene clusteing and operon detection (*ISMB'03*)

Supervised graph inference

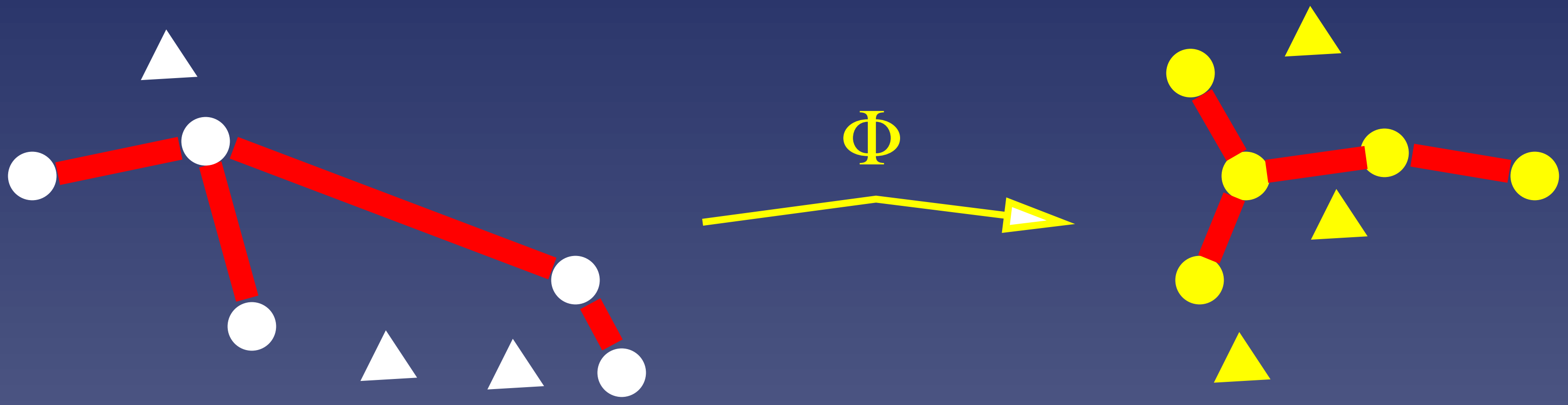


Bayesian networks (Friedman et al., 2001), dynamical systems (Akutsu, 2000), nearest neigbor joinging method (Marcotte et al., 1999)...

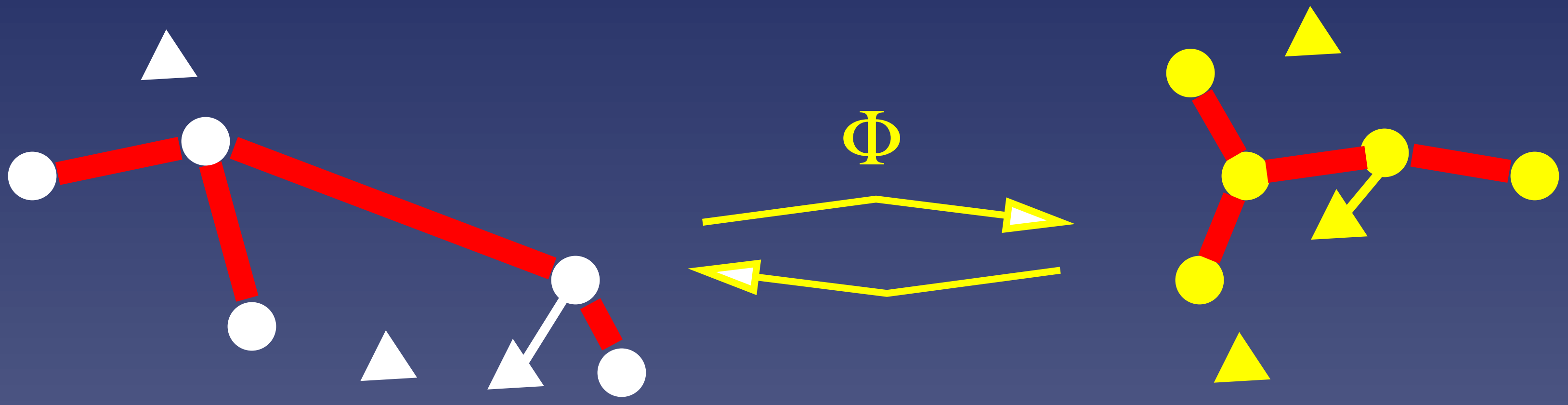
Graph learning through metric learning



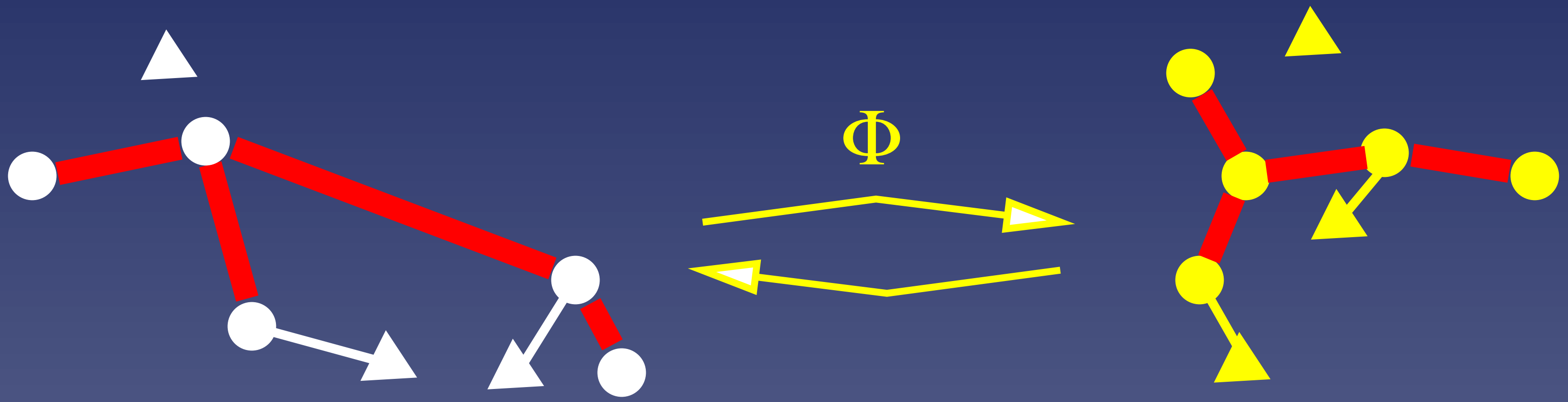
Graph learning through metric learning



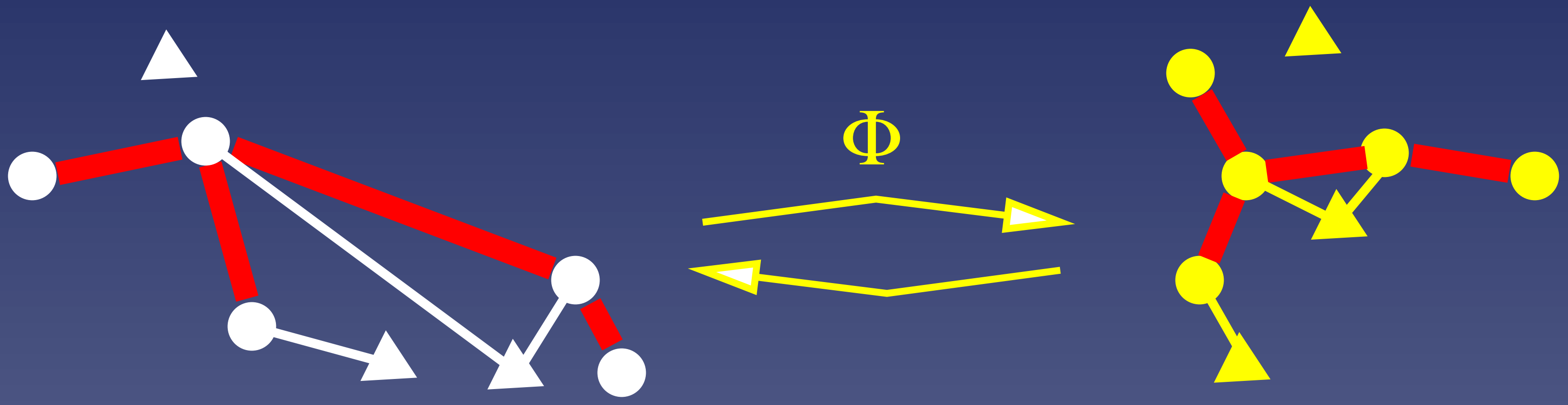
Graph learning through metric learning



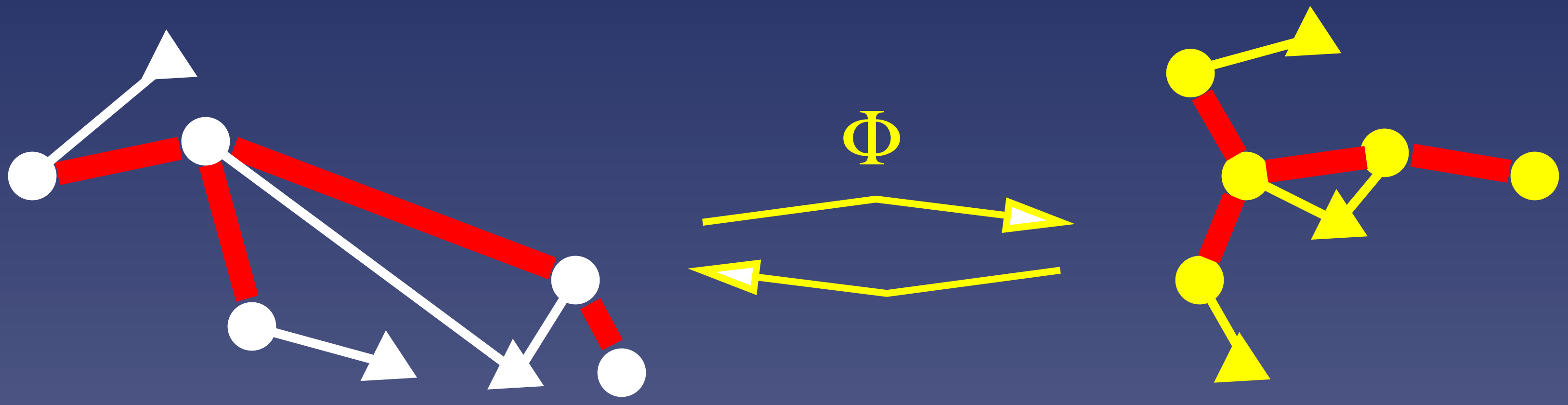
Graph learning through metric learning



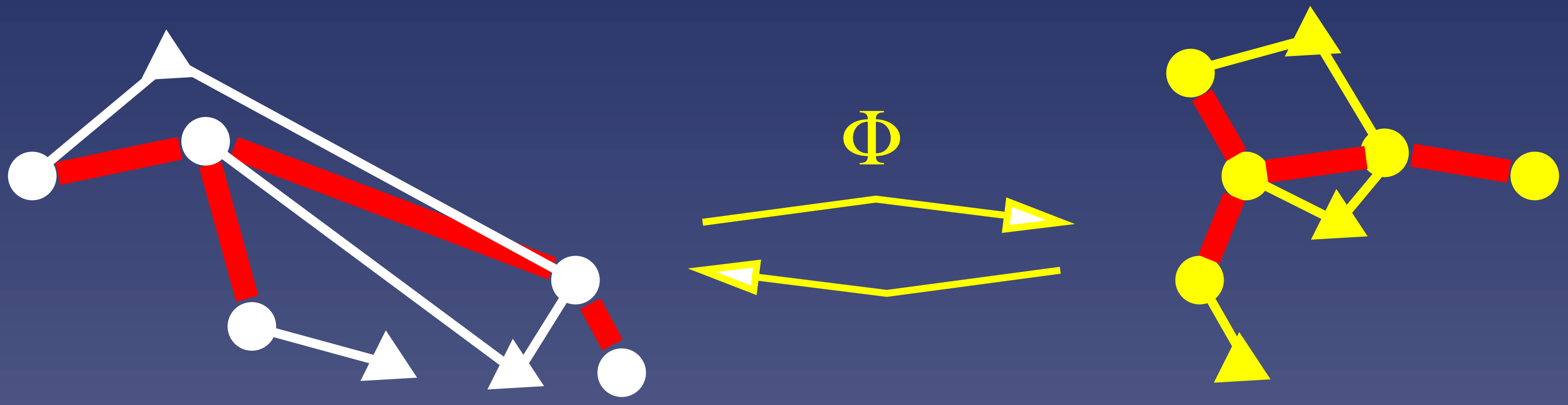
Graph learning through metric learning



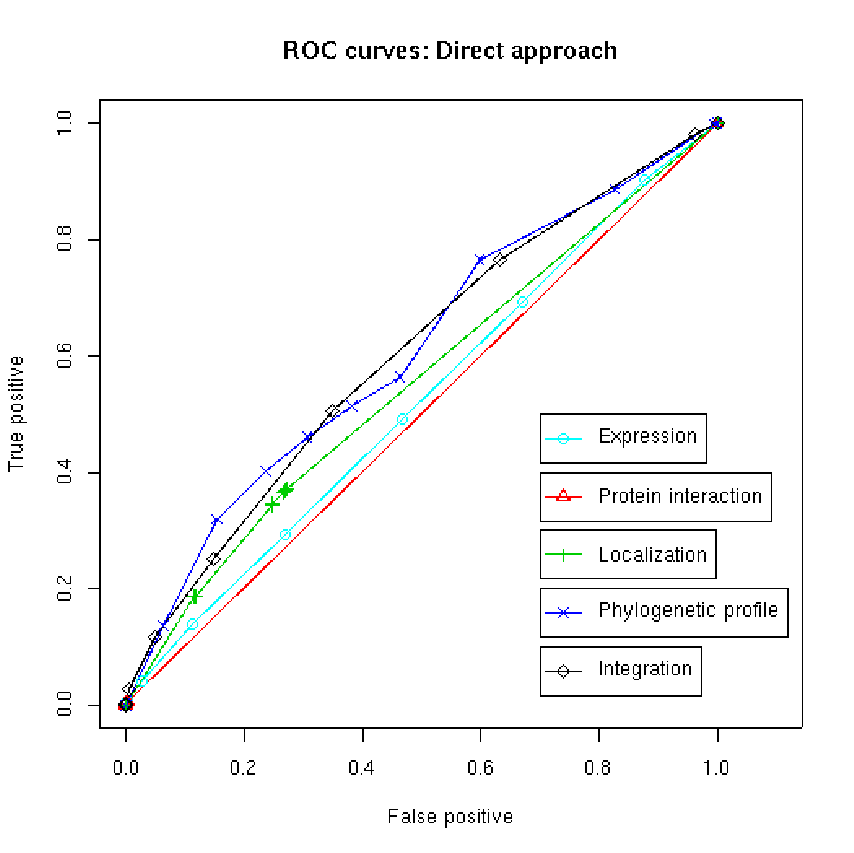
Graph learning through metric learning



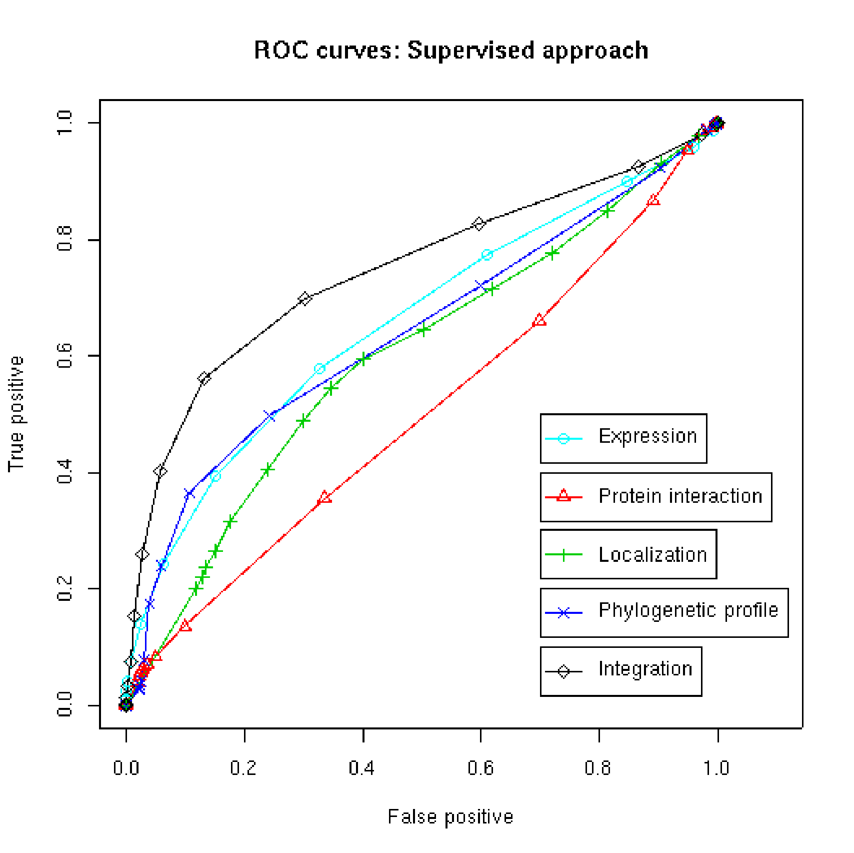
Graph learning through metric learning



Unsupervised graph learning



Supervised graph learning



Future of computational biology

- A strong and increasing demand to solve well-defined problems

Future of computational biology

- A strong and increasing demand to solve well-defined problems
- More and more possibilities to formulate new hypothesis/theories from results of data mining (e.g., scale-free properties...)

Future of computational biology

- A strong and increasing demand to solve well-defined problems
- More and more possibilities to formulate new hypothesis/theories from results of data mining (e.g., scale-free properties...)
- An urgent need for an adapted mathematical framework to represent and integrate biological data (probabilistic? kernel methods? dynamic systems? operator algebra?...)

Future of computational biology

- A strong and increasing demand to solve well-defined problems
- More and more possibilities to formulate new hypothesis/theories from results of data mining (e.g., scale-free properties...)
- An urgent need for an adapted mathematical framework to represent and integrate biological data (probabilistic? kernel methods? dynamic systems? operator algebra?...)
- How to transfer fundamental findings into applications, such as new therapies?

A challenge for the CAS-MPI Institute

- Seek a fast international recognition through an original and high-level research

- Strong collaboration with the CAS biological and medical facilities, and with the MPI excellence centers in computer science and mathematics

- Focus on a small number of well-defined applications, in collaboration with nearby CAS laboratories

- Keep a long-term theoretical goal

Acknowledgements

Collaborators at Kyoto University, University of Washington, UC Berkeley, UC Davis, MPI Tübingen, Institut Pasteur, Institut Curie, Paris 6 University