Kernel methods in computational biology

Jean-Philippe.Vert@mines.org

Ecole des Mines de Paris
Computational Biology group

Learning@Snowbird, April 8, 2004.

Motivations

Biology is facing many machine learning challenges. Massive amounts of data are generated, characterized by:

- structured and heterogeneous data (sequences, 3D structures, graphs, networks, expression profiles, phylogenetic trees, SNP, ...)
- in large quantities (10^6 gene sequences)
- in high dimension (one DNA chip monitors $10^5 \sim 10^6$ genes)

Motivations

Kernel methods provide (partial) solutions to this challenges:

- Kernels for structured data
- Operations on kernels to integrate heterogeneous data
- Regularization (in rkhs) to cope with high dimension
- Statistical approaches to extract informations from large amounts of data

Motivations

SVM and kernel methods are becoming popular in bioinformatics

- "Kernel methods in computational biology", MIT Press, 2004
- "Applications of SVM in computational biology", Bill Noble, 2004, available on the web

Overview

1. Local alignment kernels for biological sequences
2. Supervised gene network inference

Part 1

Local alignment kernel for biological sequences

(with S. Hiroto and T. Akutsu)

Biological sequences

- High-throughput genome sequencing produces many sequences
- 181 published genomes (including human!), 1084 ongoing projects

Gene sequences

- Genes are short parts in the genome, automatically detected by computational methods.
- Genes encode proteins $=$ molecules of interest
- Currently $\sim 10^6$ gene sequences available
- Challenges: annotate, classify, predict structures, functions, interactions, regulation...

Kernel methods

- In order to apply kernel methods, we need a kernel for gene sequences
- Sequences of length $50 \sim 1000$ over a 20-letter alphabet $\mathcal{A}$ (the amino acids)

Related work

- Spectrum/mismatch kernel (Leslie et al.,2002/03):

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

Related work

- Spectrum/mismatch kernel (Leslie et al.,2002/03):

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

- Fisher kernel (Jaakkola et al., 2000): 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.

Our approach

- Remember a kernel $K(x, y)$ can be thought of as a measure of similarity between x and y
- Methods to score the similarity of gene sequences have been developed and "optimized" over the last 20 years.
- Can they be used as kernels?
- How to develop kernels that mimic them?

Local alignment

- Let two strings:

$$x = \texttt{AMACGGSLIAMMWFGVRFF}$$
$$y = \texttt{LGCLIVMMNRLMWFGVSGVV}$$

- A local alignment with gaps π is for example:

```
AMACGGSLIAMM----WFGVRFF.
...|...|||||....||||....
.LGC---LIVMMNRLMWFGVSGVV
```

Local alignment score

- $S : \mathcal{A}^2 \to \mathbb{R}$ (substitution matrix)
- $g : \mathbb{N} \to \mathbb{R}$ (gap penalty function)

```
AMACGGSLIAMM----WFGVRFF.
...|...|||||....||||....
.LGC---LIVMMNRLMWFGVSGVV
```

$$s_{S,g}(\pi) = S(C,C) + S(L,L) + S(I,I) + S(A,V) + 2S(M,M) + S(W,W) + S(F,F) + S(G,G) + S(V,V) - g(3) - g(4)$$

Smith-Waterman (SW) score

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

- Computed by dynamic programming $O(|x||y|)$
- Not a kernel in general (VSA, 2004)

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

- Interpretation in terms of local alignment scores:

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

Properties

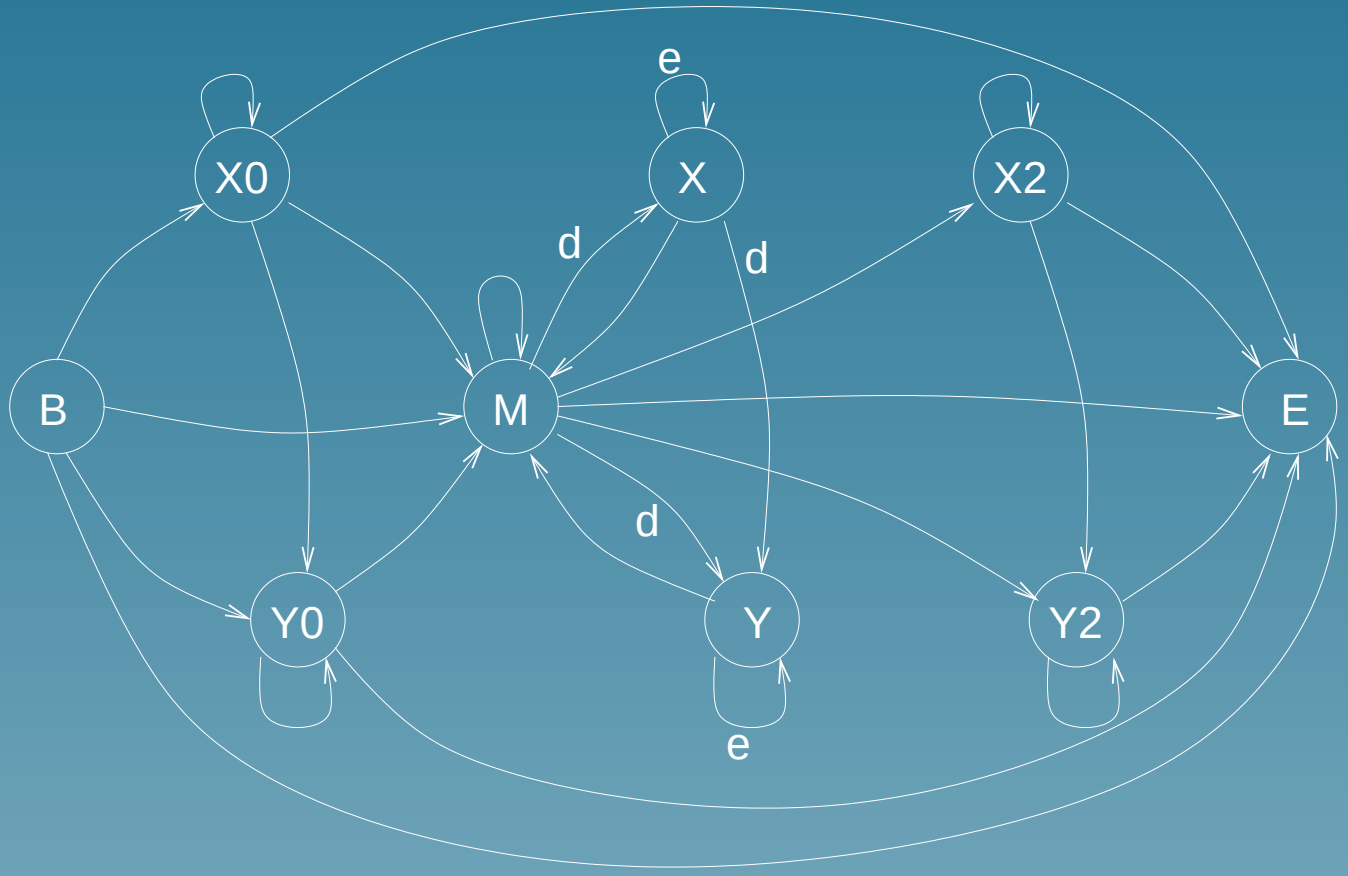
- Interpretation in terms of local alignment scores:

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

- Link with the SW score:

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

Kernel computation



LA Kernel in practice

- $K(x, y)$ decreases exponentially with $|x|$ and $|y|$
- Problem of diagonal dominance, and normalization
- Caveat: take

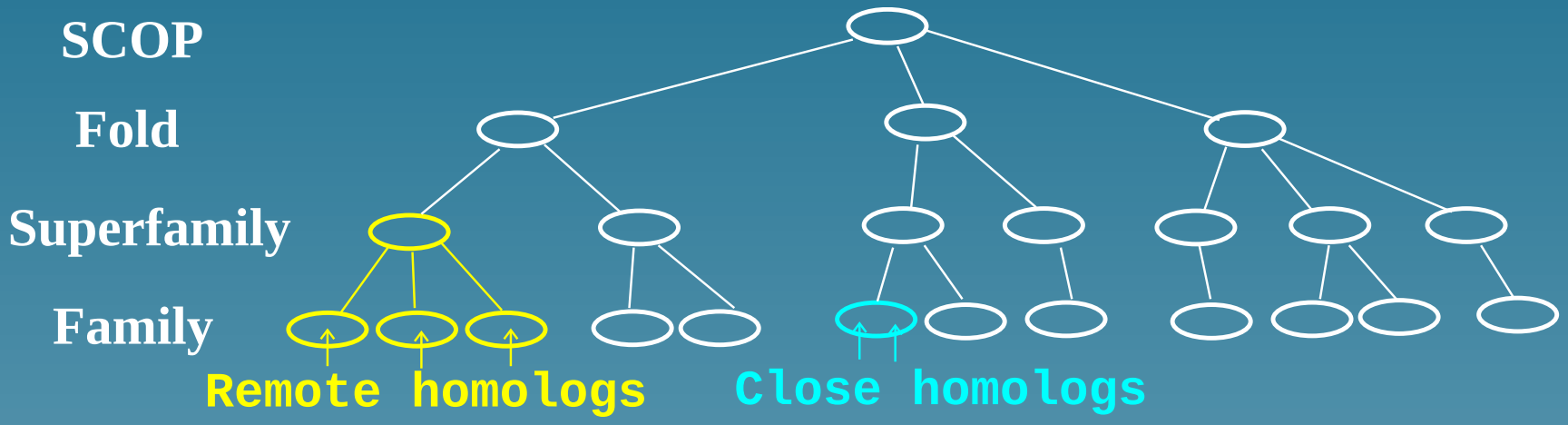
$$\tilde{K}^{(\beta)}_{LA}(x, y) = \frac{1}{\beta} \ln K^{(\beta)}_{LA}(x, y)$$

and "massage the matrix" to make it positive definite

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?

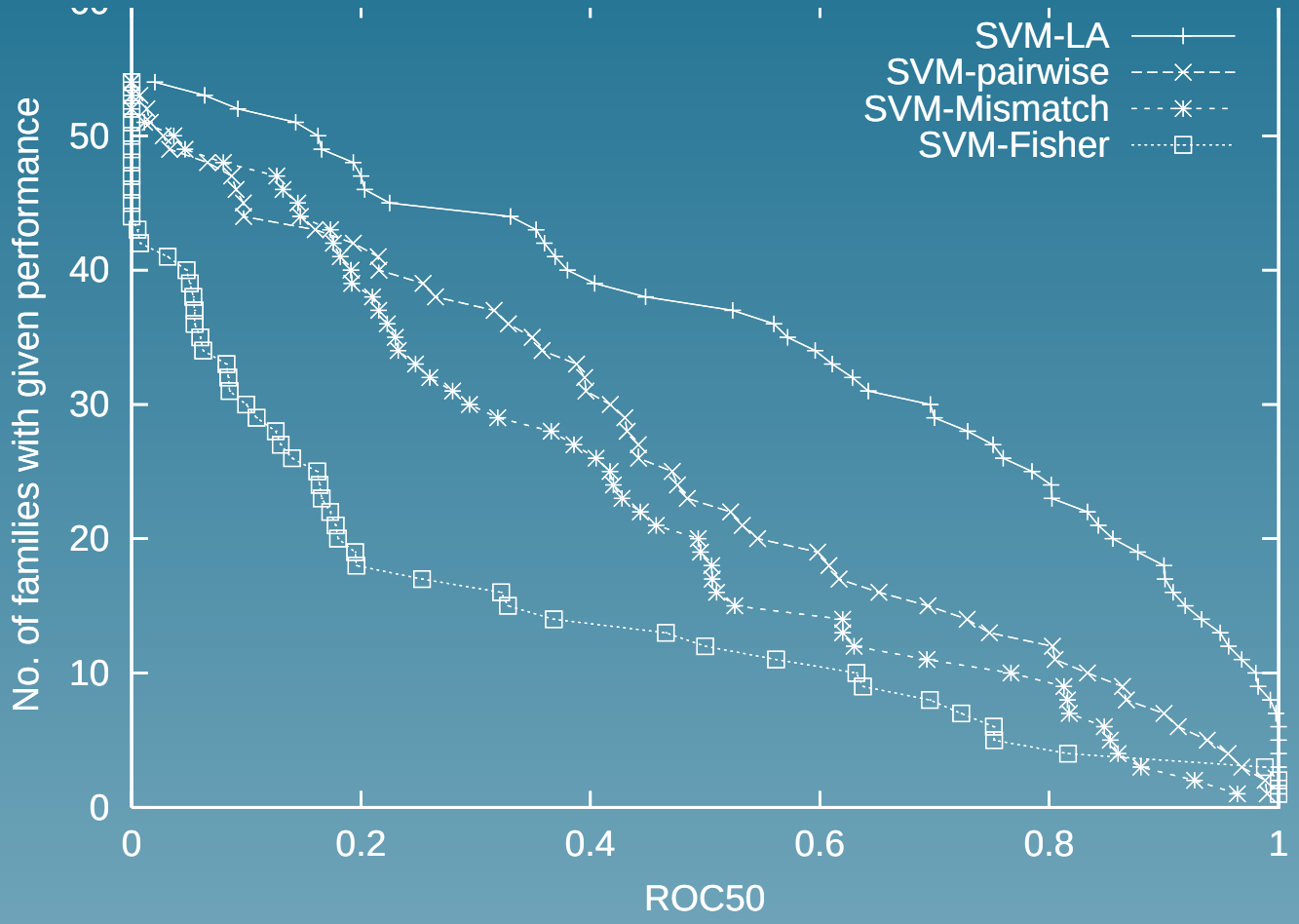
A benchmark experiment

- 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



Open questions / Ongoing work

- Length normalization?

Open questions / Ongoing work

- Length normalization?
- For which parameters g and S is SW a valid kernel?

Open questions / Ongoing work

- Length normalization?
- For which parameters g and S is SW a valid kernel?
- What is the trade-off between diagonal dominance issues and other properties of string kernels?

Part 2

Supervised gene network inference

(with Y.Yamanishi)

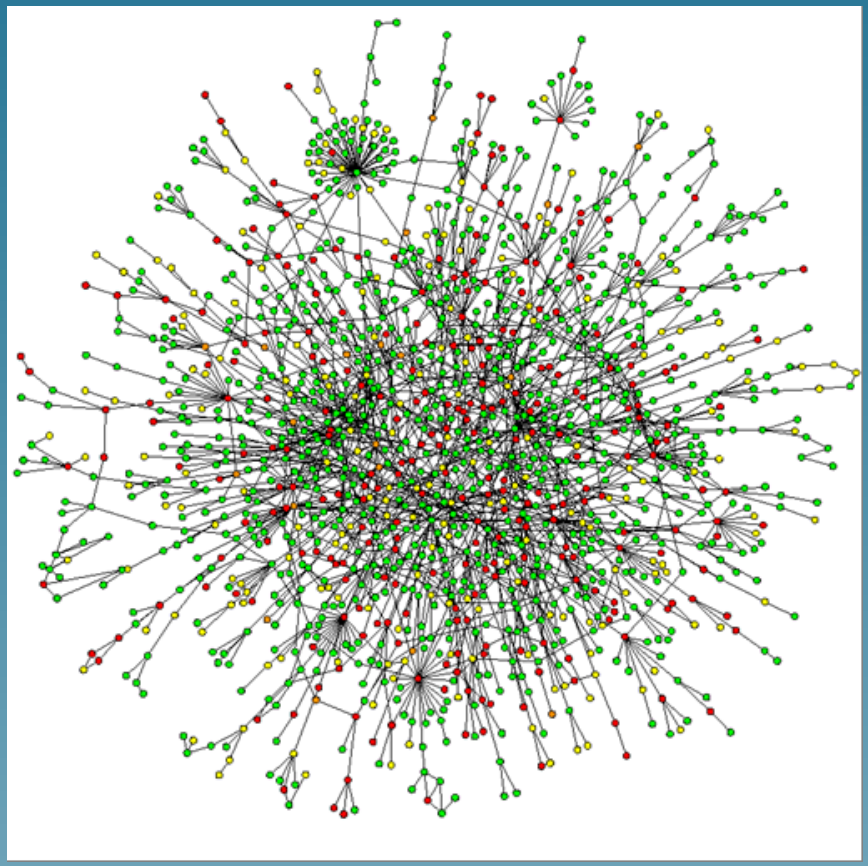
Motivations

- Most biochemical/biological processes involve interactions between genes
- Deciphering these interactions is the next big challenge in computational biology ("systems biology")
- Mathematically, a graph is a convenient representation when only pairwise interactions are considered

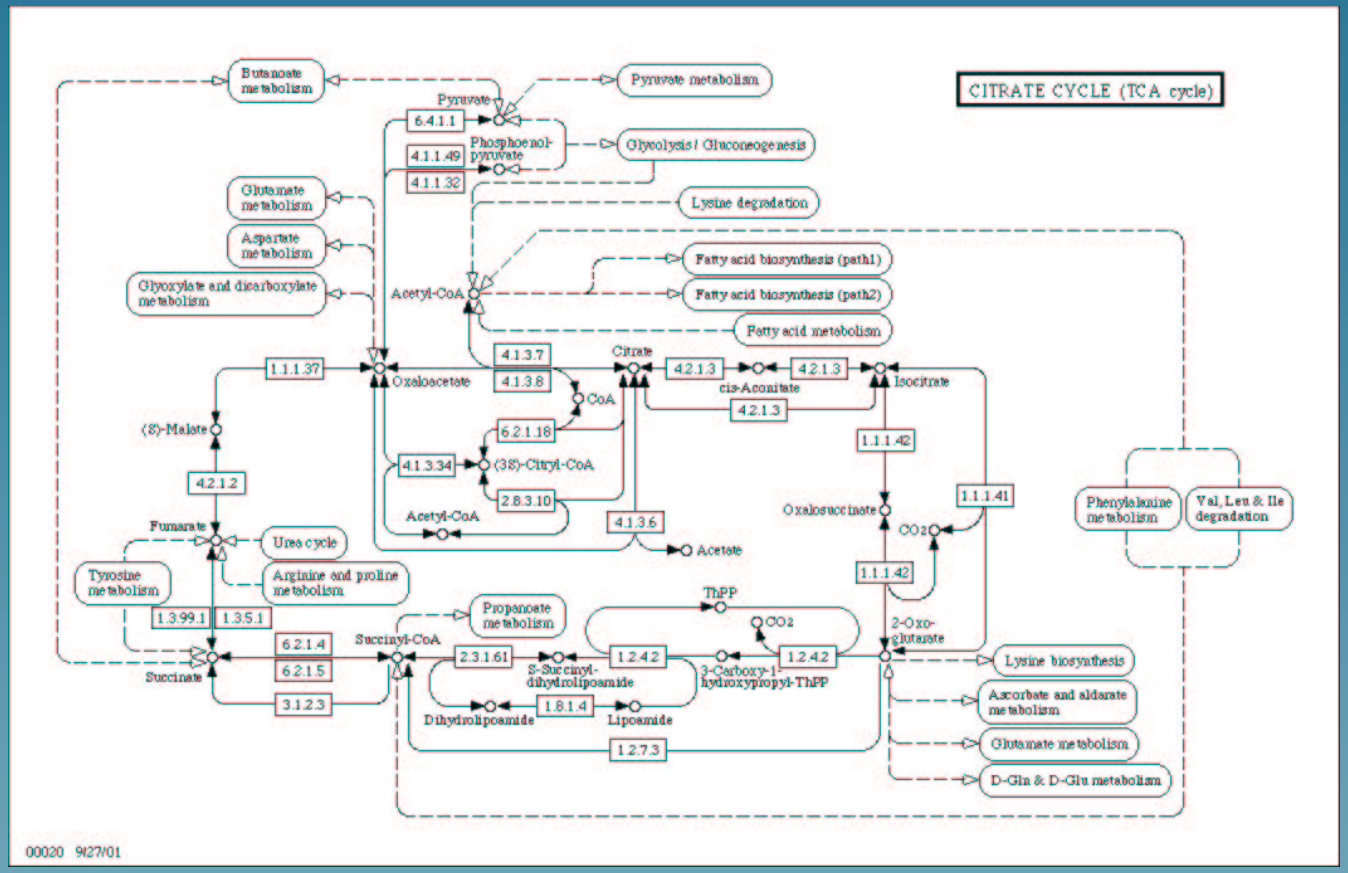
Gene/protein network examples

- physical interaction network (interactome)
- gene regulatory network
- biochemical/metabolic network

Example: the yeast interactome



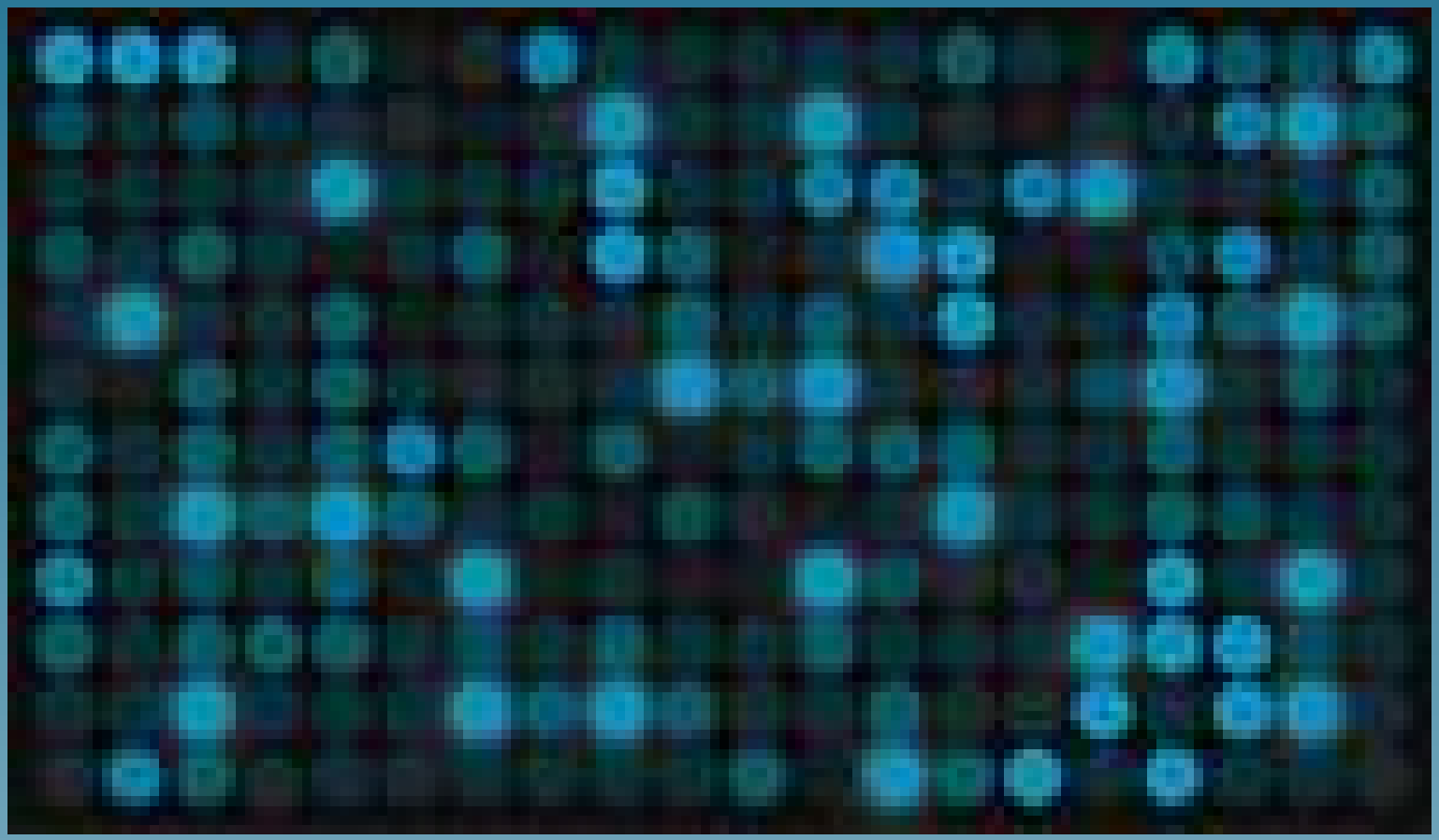
Example: metabolic network



The network inference problem

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

Example: gene expression



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.,

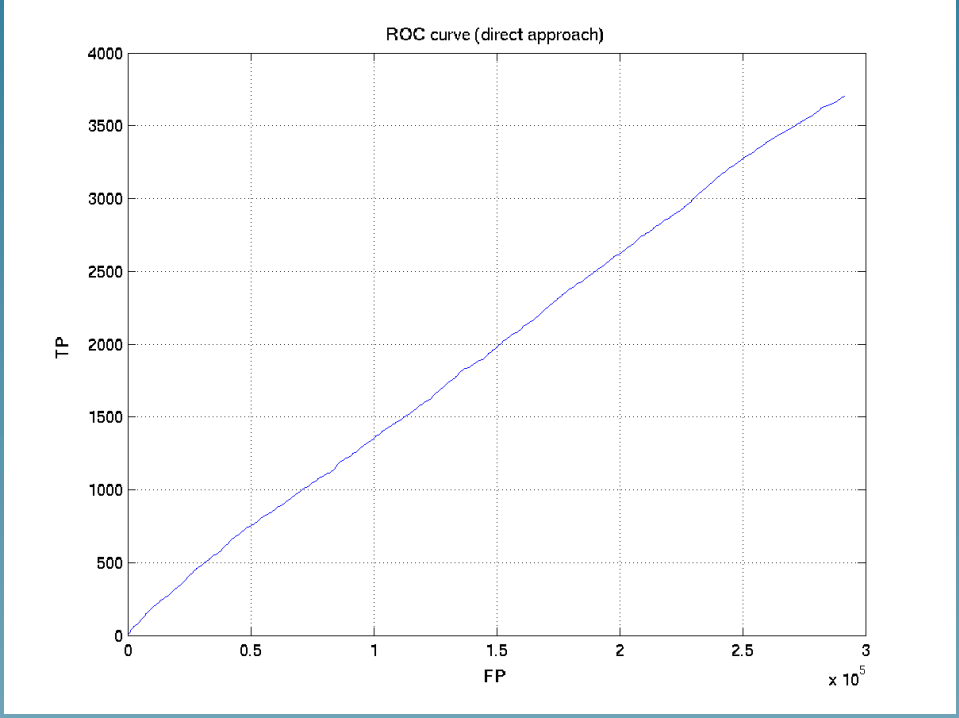
$$K(x, y) = \exp\left(-\frac{||e(x) - e(y)||^2}{2\sigma^2}\right)$$

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

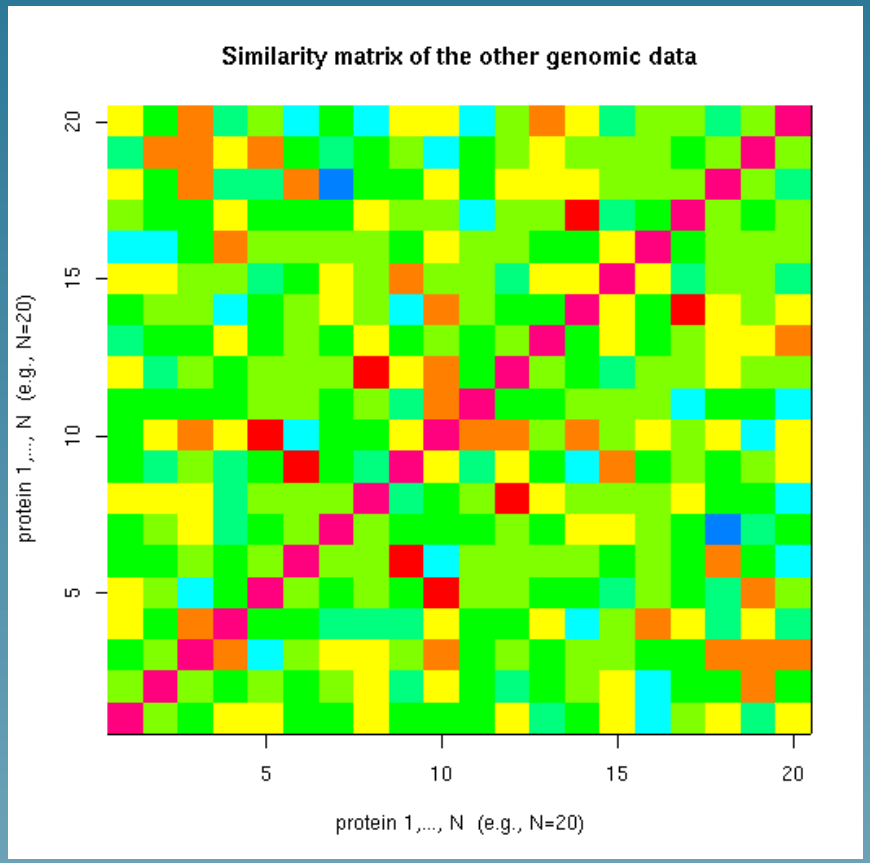
- Direct strategy: add edges between genes by decreasing similarity.

Evaluation of the direct approach

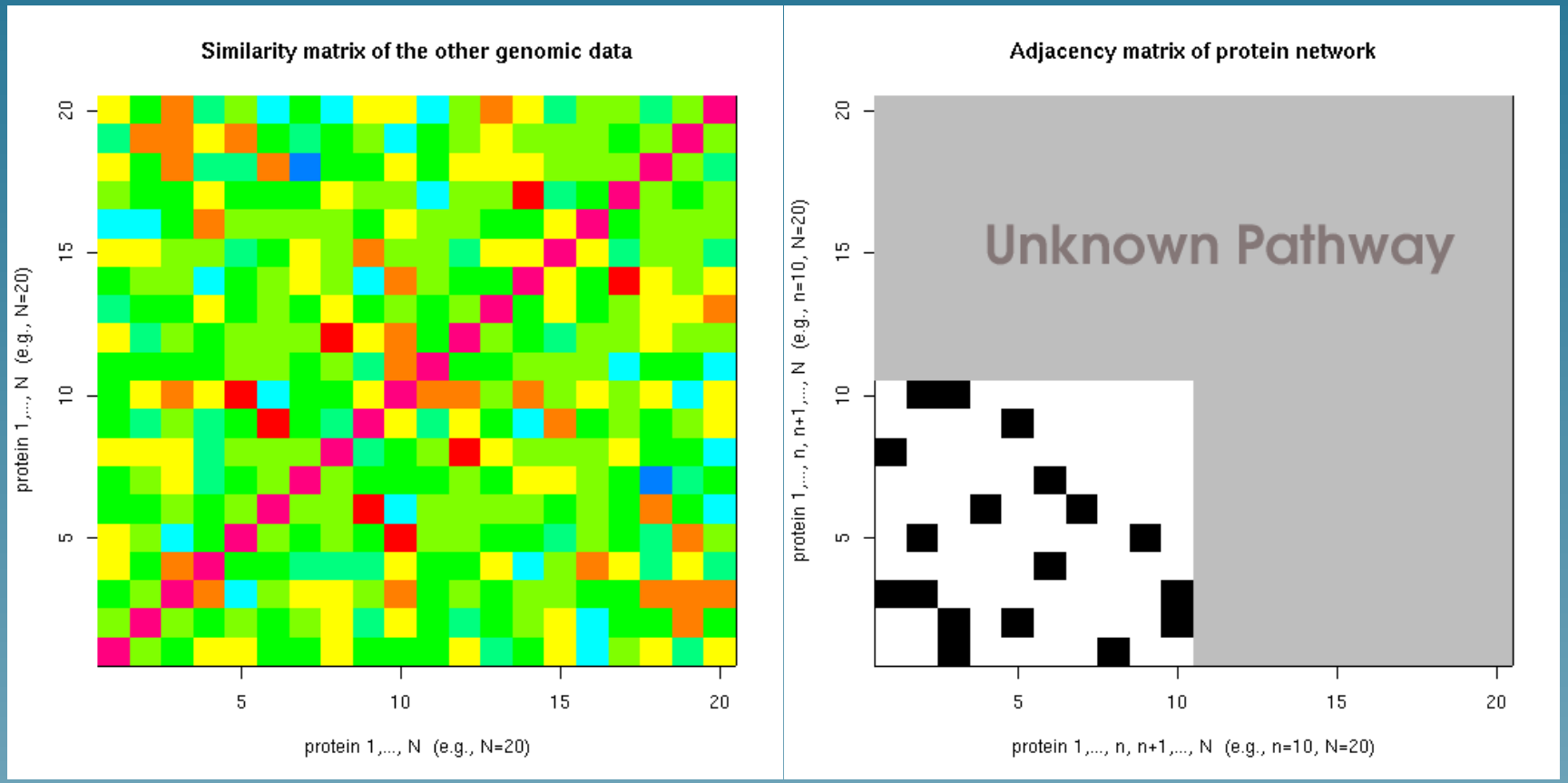
The metabolic network of the yeast involves 769 genes. Each gene is represented by 157 expression measurements. (ROC=0.52)



The supervised gene inference problem



The supervised gene inference problem



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

Criterion for f

- A feature $f : \mathcal{X} \to \mathbb{R}$ is good on the training set if connected genes have similar value. A possible criterion is:

$$R(f) = \sum_{(x,y)\in E} (f(x) - f(y))^2 - \sum_{(x,y)\notin E} (f(x) - f(y))^2$$

Criterion for f

- A feature $f : \mathcal{X} \to \mathbb{R}$ is good on the training set if connected genes have similar value. A possible criterion is:

$$R(f) = \sum_{(x,y)\in E} (f(x) - f(y))^2 - \sum_{(x,y)\notin E} (f(x) - f(y))^2$$

- When $\sum_{i=1}^{n} f(x_i) = 0$ and $\sum_{i=1}^{n} f(x_i)^2 = 1$, this reduces to:

$$R(f) = \sum_{(x,y)\in E} (f(x) - f(y))^2$$

Working in rkhs

- Searching for features $f : \mathcal{X} \to \mathbb{R}$ in the rkhs $\mathcal{H}$ defined by the kernel K, this suggests the following optimization problem:

$$\min_{f \in \mathcal{H}_0} \sum_{(x,y) \in E} (f(x) - f(y))^2 + \lambda ||f||^2_{\mathcal{H}}$$

where $\mathcal{H}_0$ is the set of functions $f \in \mathcal{H}$ such that $\sum_{i=1}^n f(x_i) = 0$ and $\sum_{i=1}^n f(x_i)^2 = 1$

Solving the problem

- By the representer theorem, f can be expanded as:

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

Solving the problem (cont.)

- The problem can then be rewritten:

$$\min_{\alpha \in \mathbb{R}^n} \left\{ \alpha^\top K_0 L K_0 \alpha + \lambda \alpha^\top K_0 \alpha \right\}$$

under the constraint $\alpha^\top K_0^2 \alpha = 1$, where:

⋆ L is the $n \times n$ graph Laplacian

⋆ K_0 is the centered $n \times n$ Gram matrix

Solving the problem (cont.)

- The problem can then be rewritten:

$$\min_{\alpha \in \mathbb{R}^n} \left\{ \alpha^\top K_0 L K_0 \alpha + \lambda \alpha^\top K_0 \alpha \right\}$$

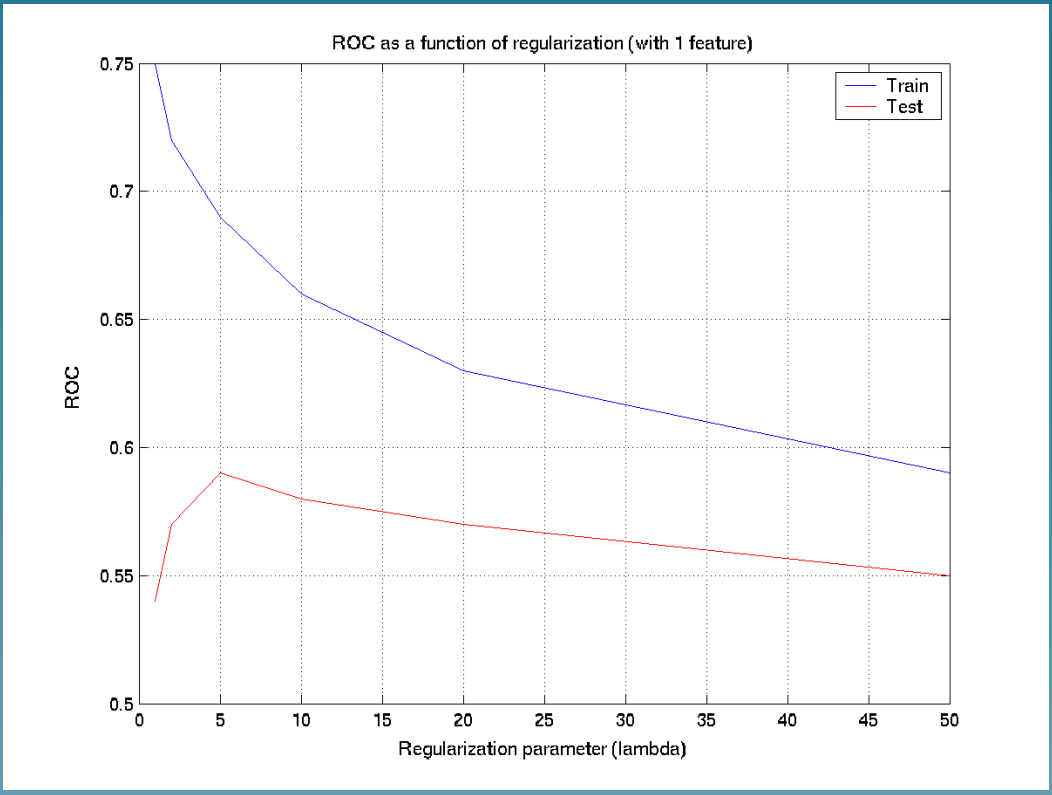
  under the constraint $\alpha^\top K_0^2 \alpha = 1$, where:

  - L is the $n \times n$ graph Laplacian
  - K_0 is the centered $n \times n$ Gram matrix

- It is equivalent to solving the generalized eigenvalue problem:

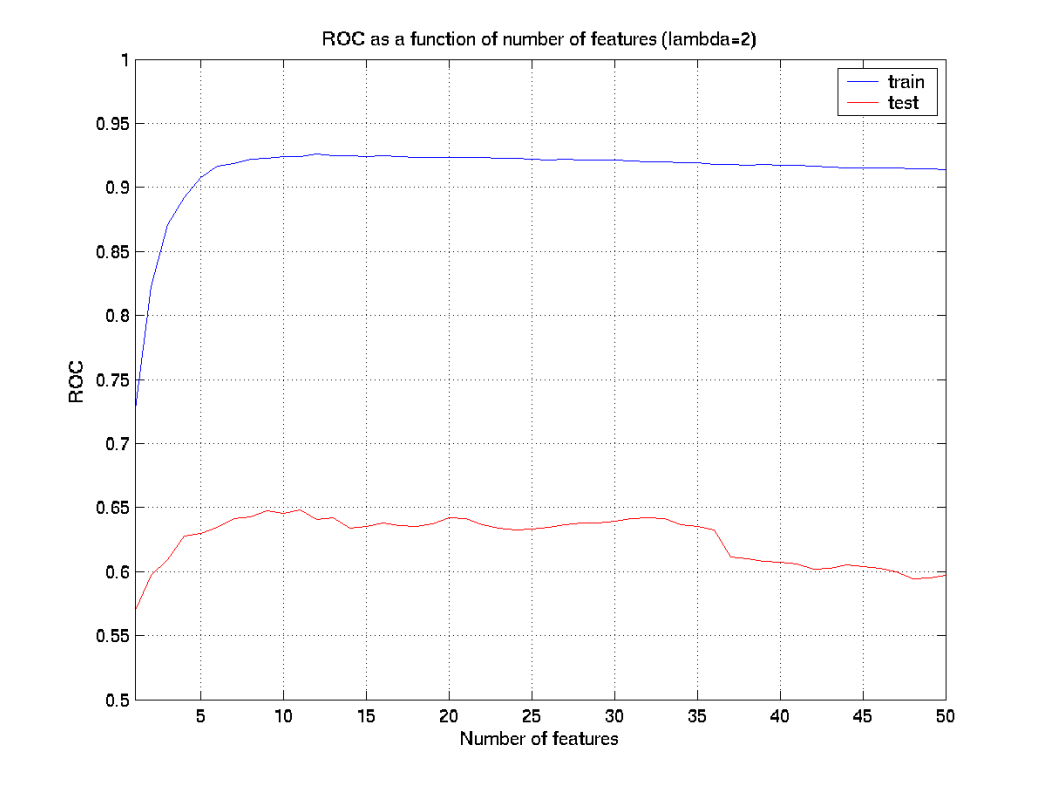
$$(LK_0 + \lambda I)\alpha = \mu K_0 \alpha.$$

Evaluation of the supervised approach: effect of λ



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

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

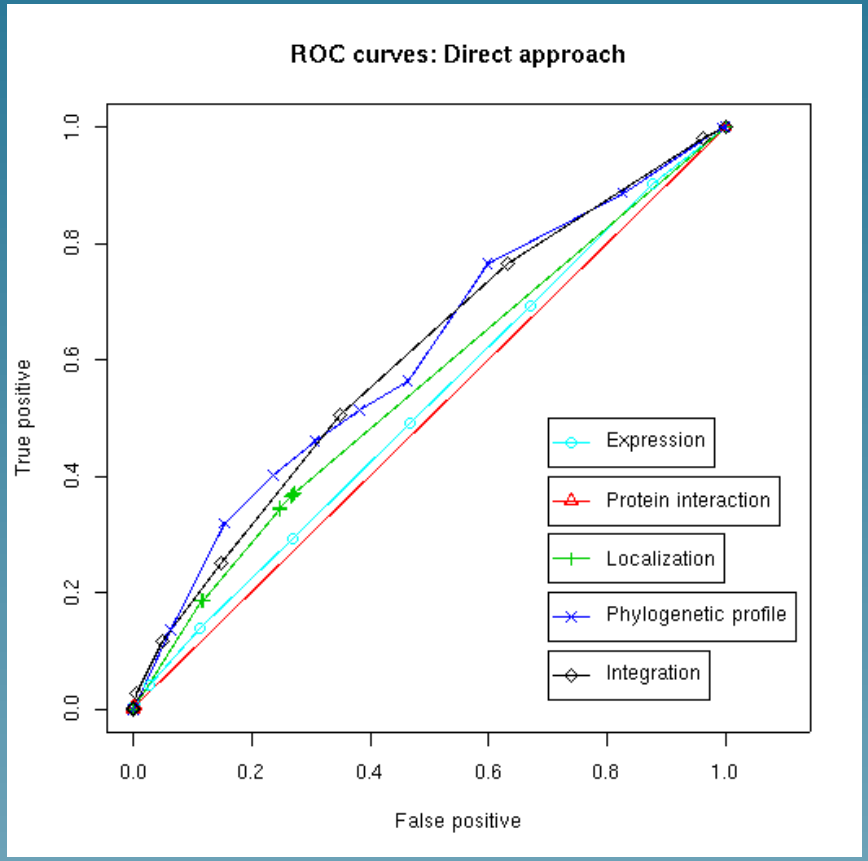


Learning from heterogeneous data

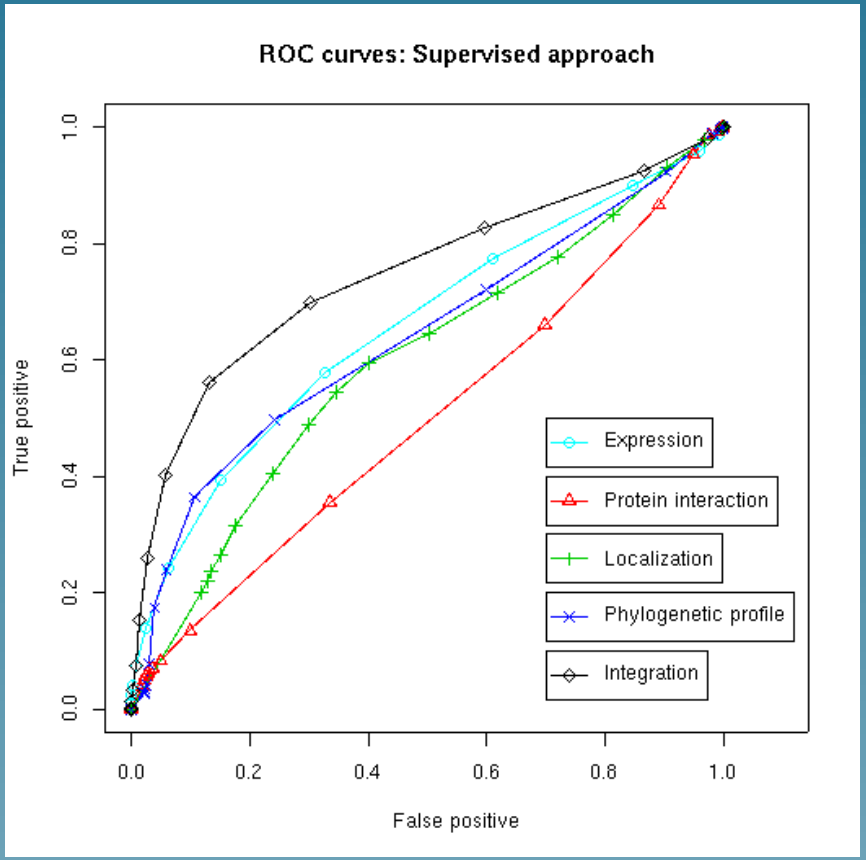
- Suppose several data are available about the genes, e.g., expression, localization, struture, predicted interaction etc...
- Each data can be represented by a kernel matrix $K_1, \ldots, K_p$
- Kernel can be combined by various operations, e.g., addition:

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

Learning from heterogeneous data (unsupervised)



Learning from heterogeneous data (supervised)



Extensions

- The Laplacian can be replaced by another inverse of a graph kernel (e.g., of a diffusion kernel)
- Other formulations can lead to kernel CCA (NIPS 02)
- The feature extracted can be used for datamining (ECCB 2003)

Open questions / Ongoing work

- What should be the number of features (problem of embedding a graph in low dimension)
- Develop a theoretical analysis of the supervised network inference problem
- Other cost functions

Conclusion

Conclusion

- Kernels offer a versatile framework to represent biological data
- A lot of work on kernel design / kernel learning, with good results on real-world data
- A new approach to supervised network inference, many possible variants and more theory required