Network inference and Inference on networks

Jean-Philippe.Vert@mines.org

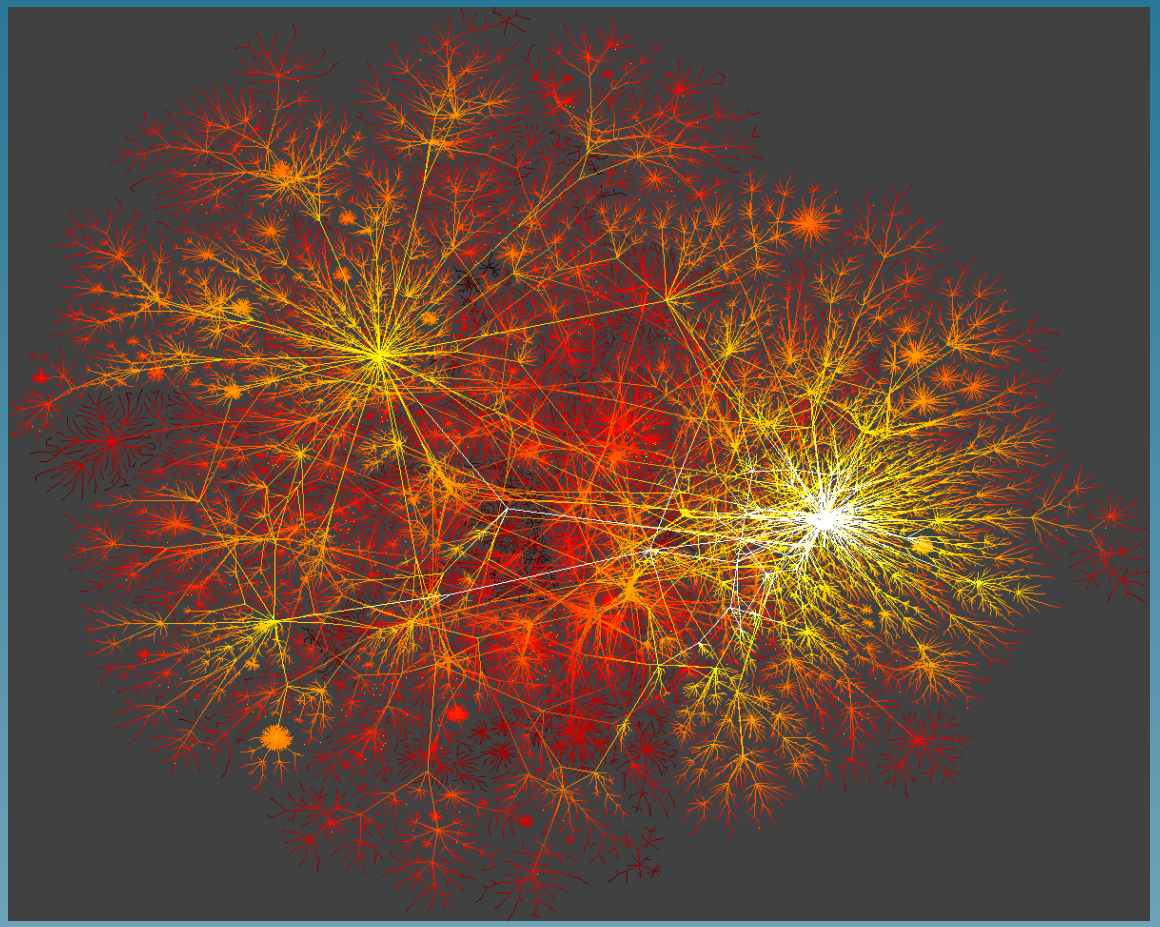
Ecole des Mines de Paris
Computational Biology group

University of Washington, Genome Science, April 14, 2004.

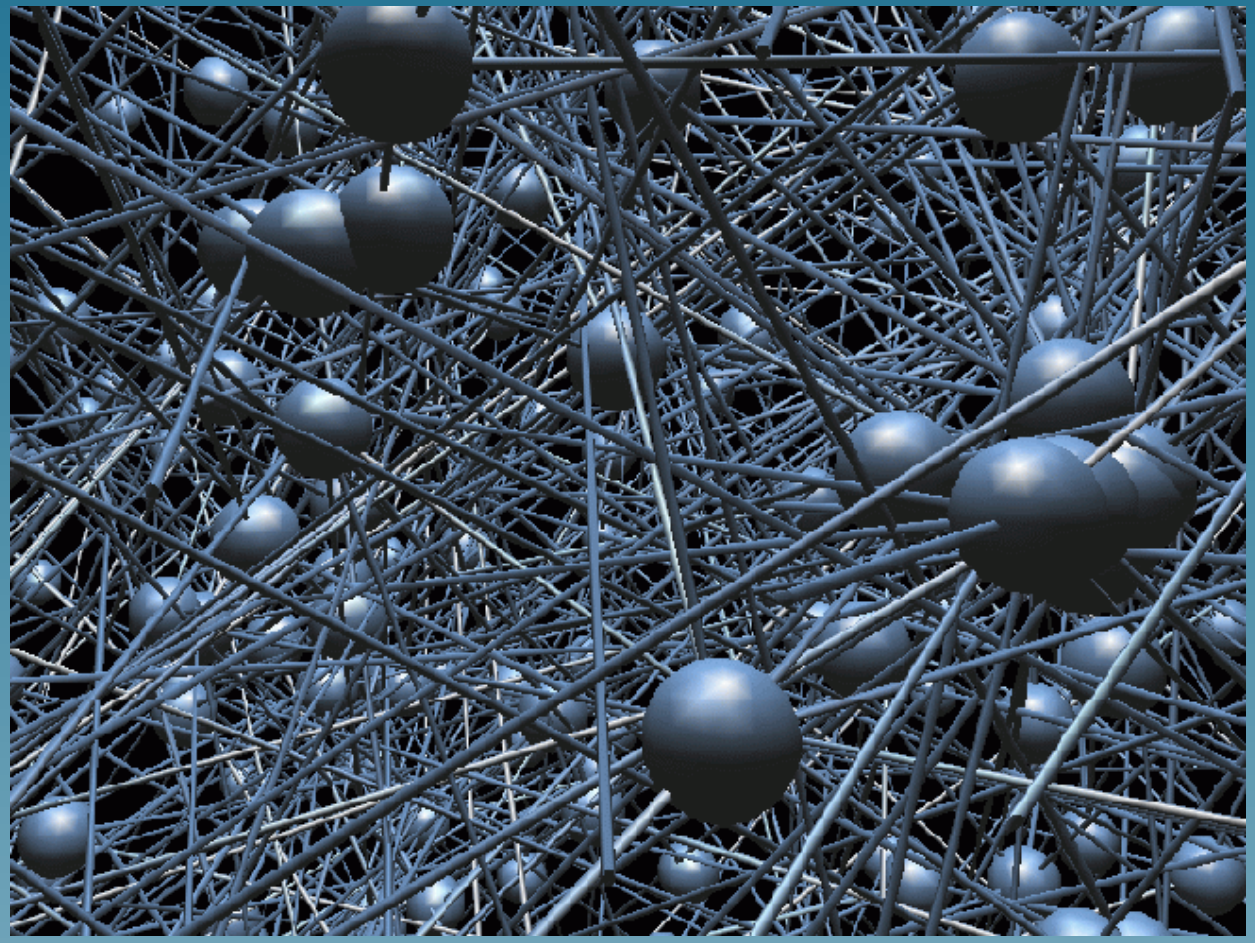
Motivations

- Large-scale graphs are nowadays ubiquitous in many research fields in particular genomics/biology...
- Large-scale high-throughput technologies, systems biology, ...
- They are getting popular in machine learning / statistics too and new methods are being developed to deal with real-world networks

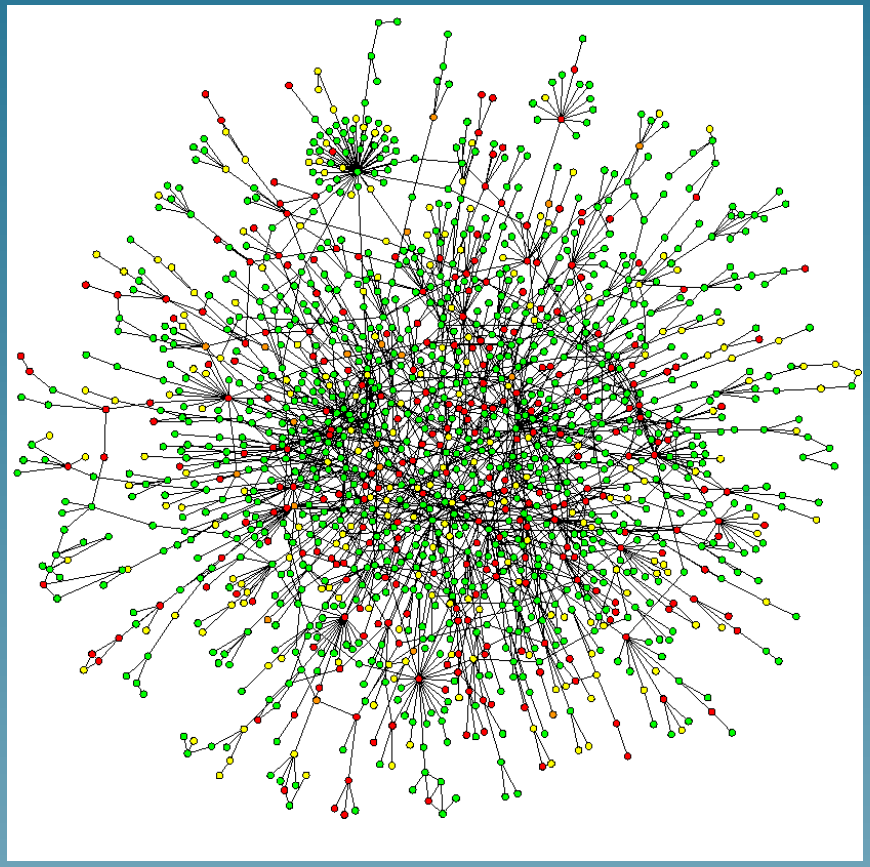
Internet



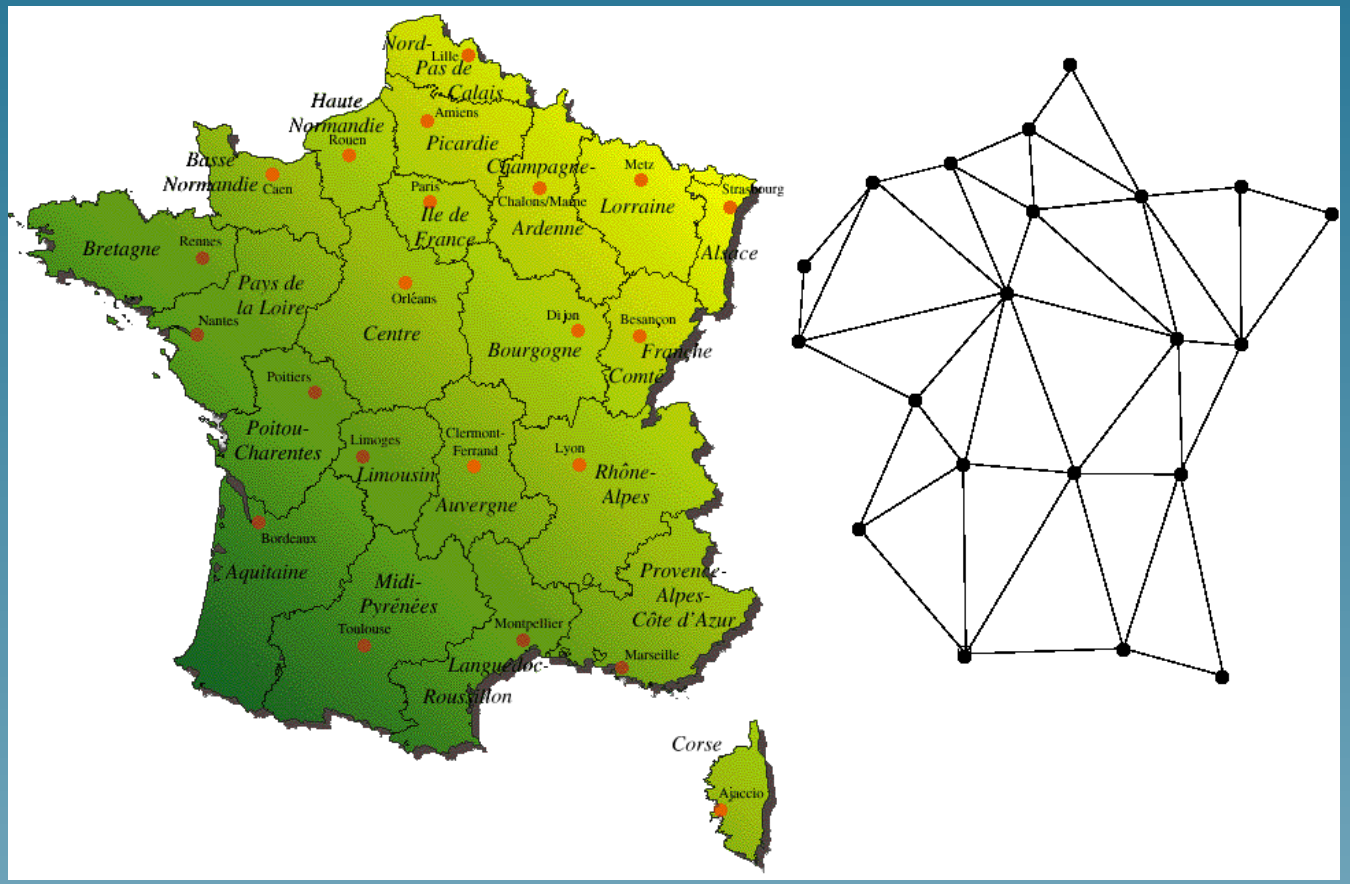
Social Network



Protein interaction network



Spatial data



Two important problems

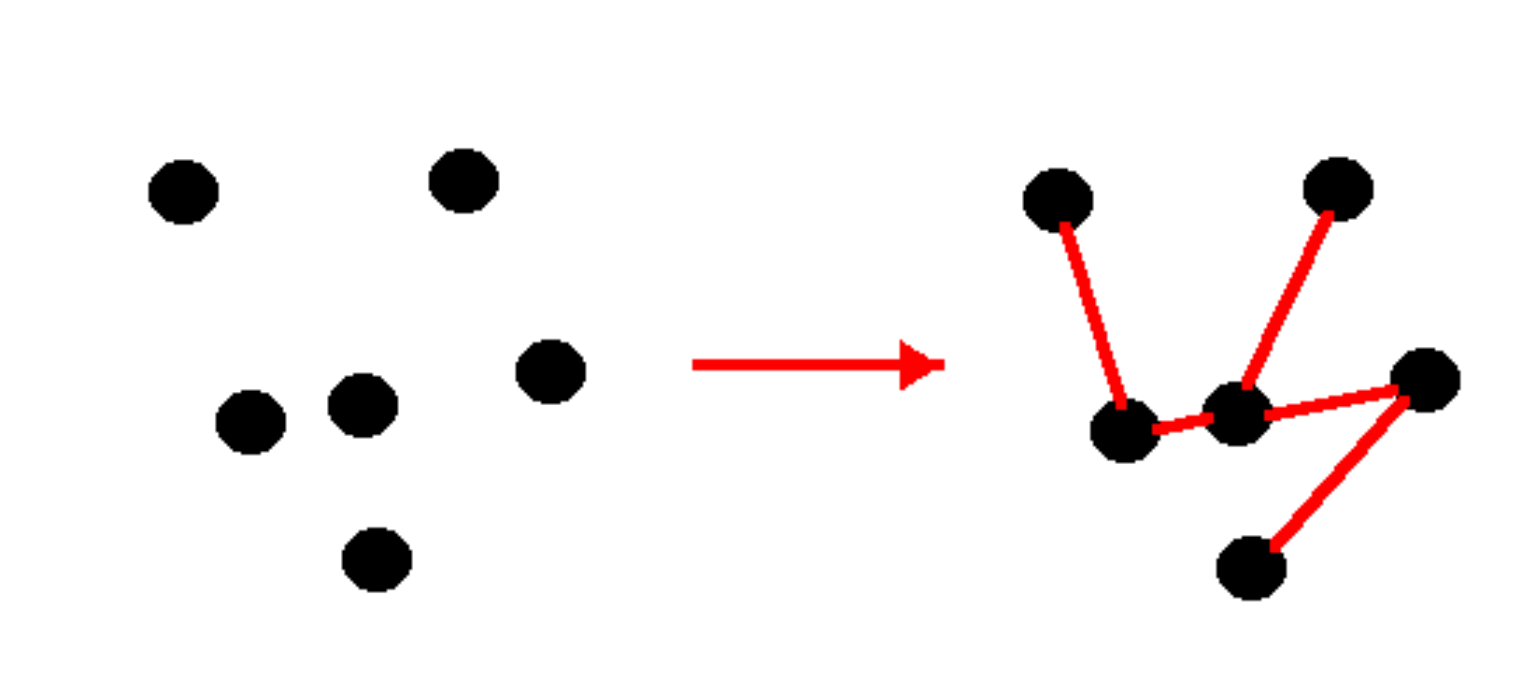
- Inferring network from observation about individual nodes.
  - ⋆ Application: gene network inference, protein interaction inference, gene regulation, metabolic pathways....
  - ⋆ Idea: "similar" nodes should be connected

Two important problems

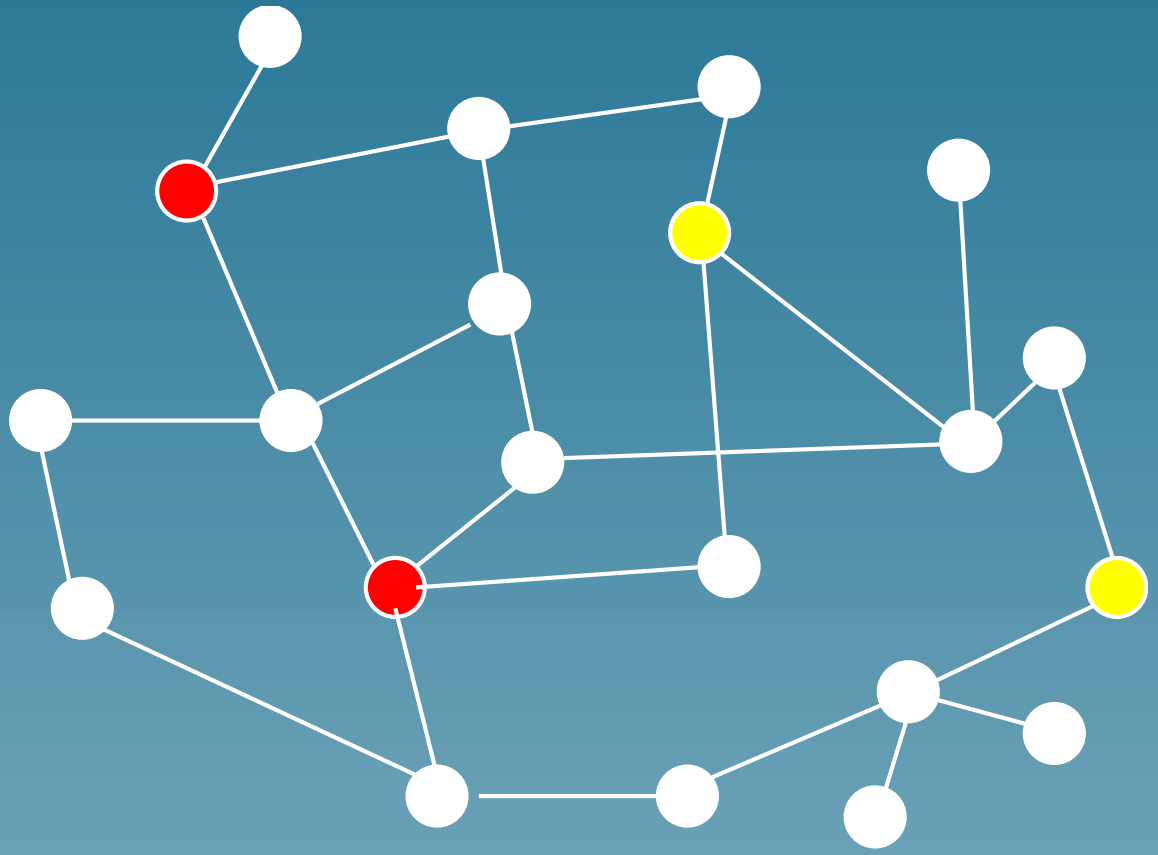
- Inferring network from observation about individual nodes.
  - ⋆ Application: gene network inference, protein interaction inference, gene regulation, metabolic pathways....
  - ⋆ Idea: "similar" nodes should be connected

- Given a network with a few labeled nodes, infer the labels of other nodes.
  - ⋆ Example: infer protein fold on the protein similarity networks.
  - ⋆ Idea = going from local similarities to global inference

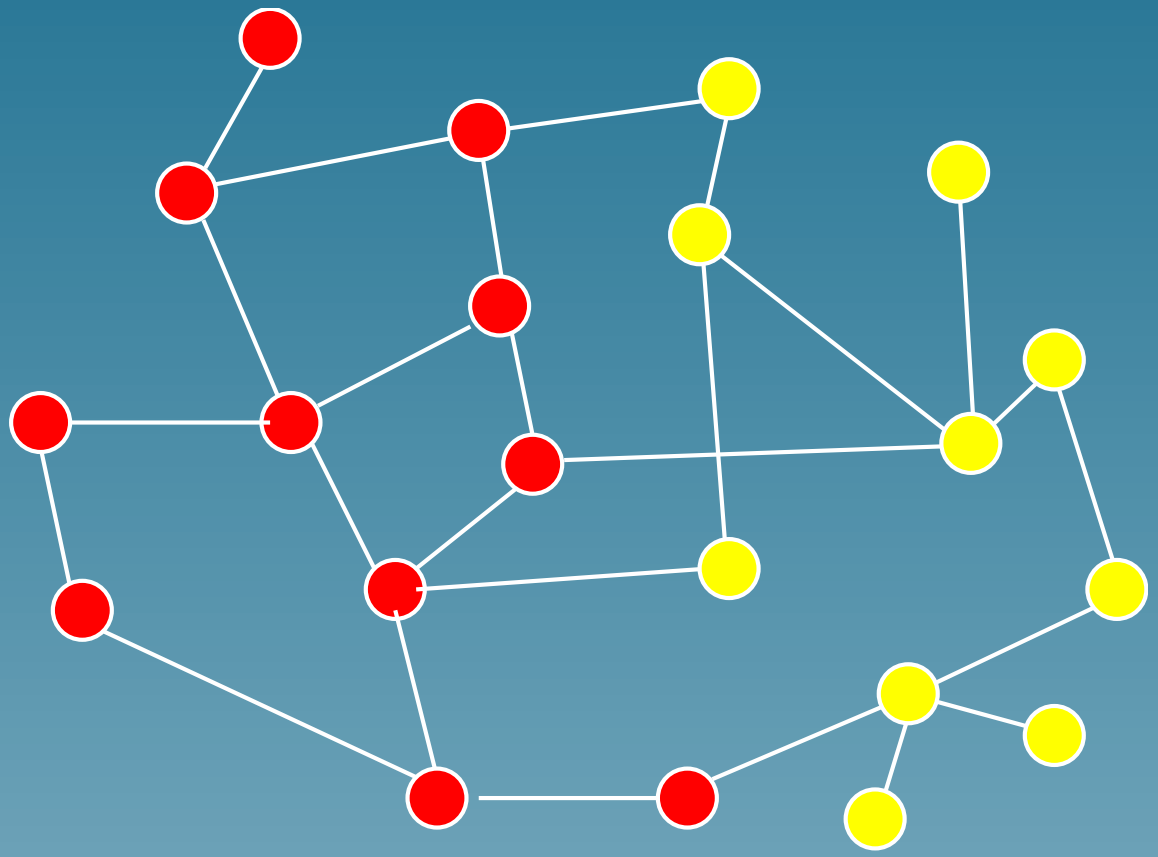
Problem 1



Problem 2



Problem 2



Part 1

Supervised gene network inference

(with Y.Yamanishi)

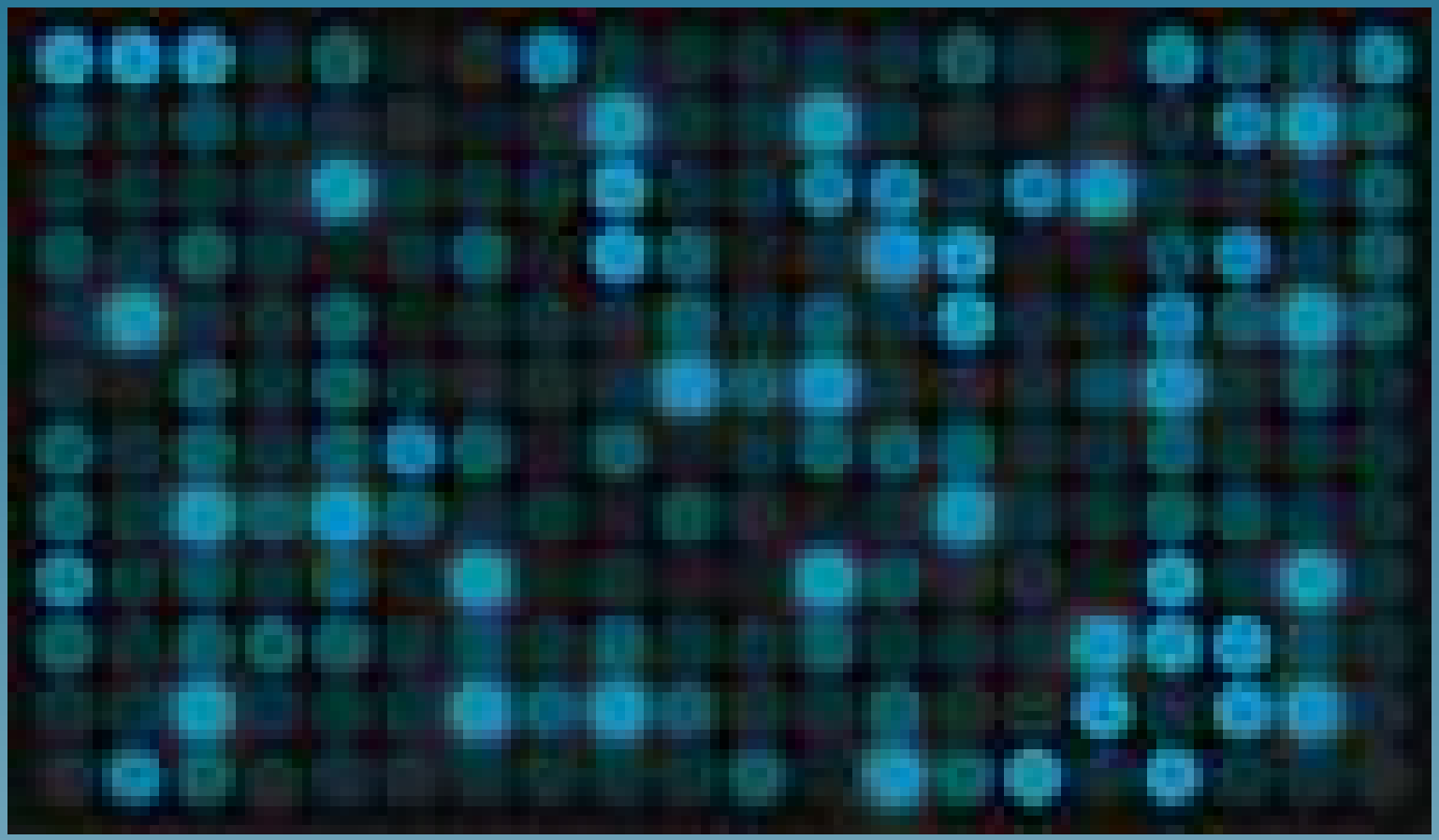
Motivations

- 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

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.,

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

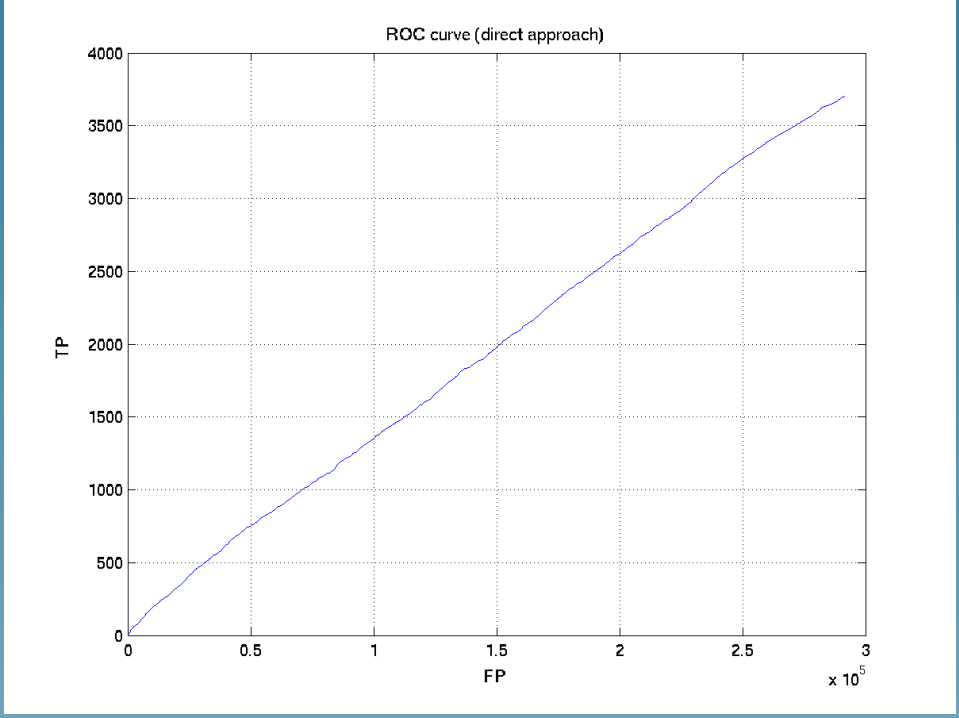
- For a set of n genes $\{x_1, \ldots, x_n\}$, let K be the $n \times n$ matrix of pairwise similarity (Gram matrix)

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

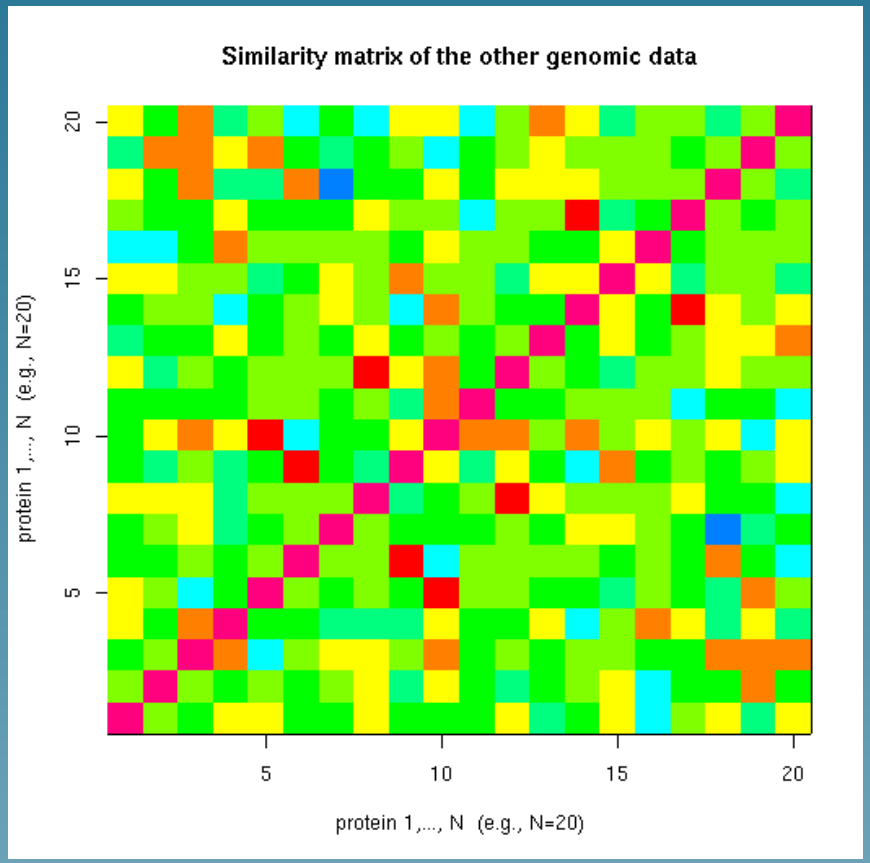
Example of similarity matrix

Evaluation of the direct approach

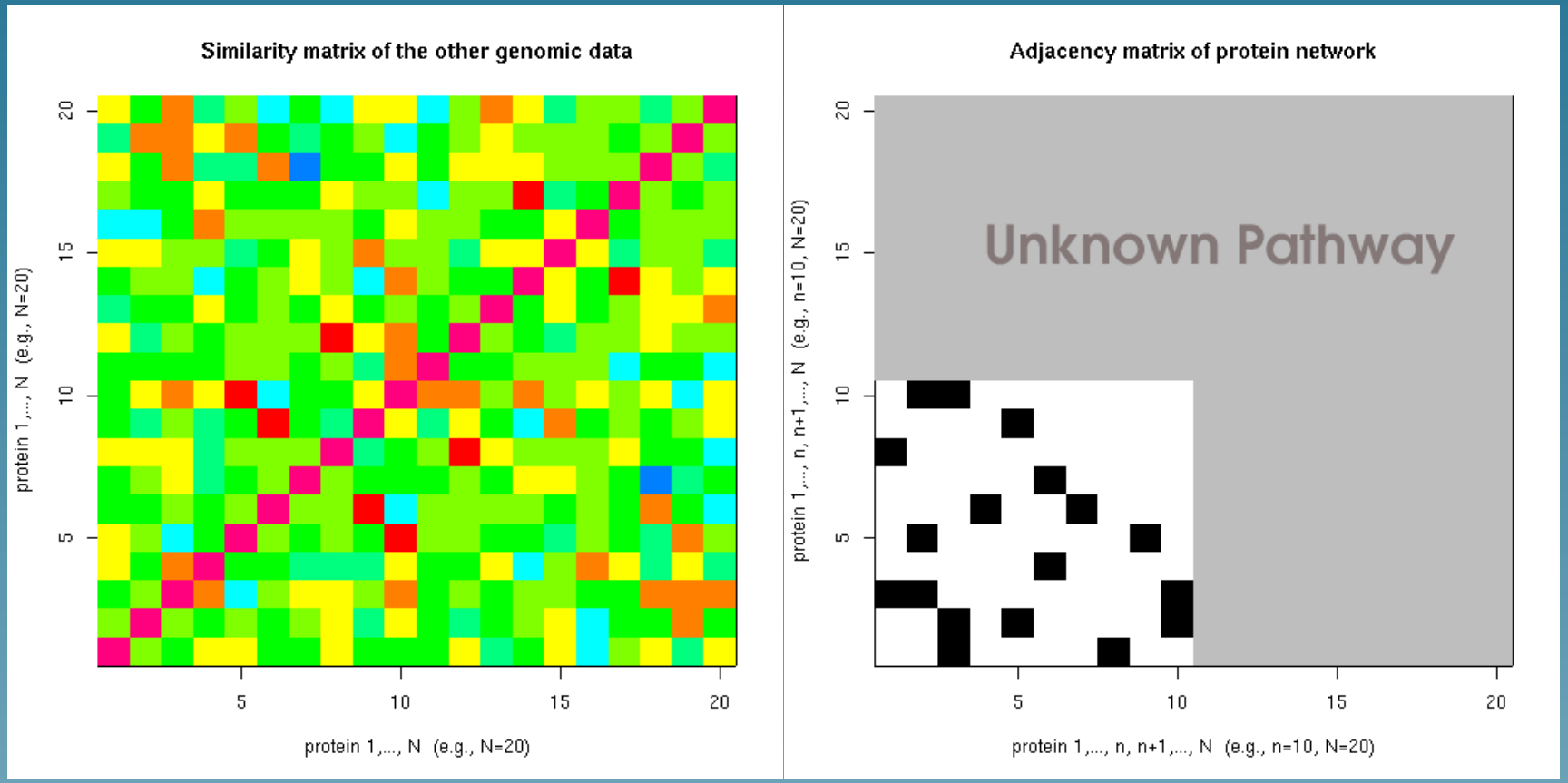
The metabolic network of the yeast involves 769 genes. Each gene is represented by 157 expression measurements. (ROC=0.52)



The supervised gene inference problem



The supervised gene inference problem



The idea in a nutshell

- Use the known network to define a more relevant measure of similarity
- For any positive definite similarity $n \times n$ matrix, there exists a representation as n-dimensional vectors such that the matrix similarity is exactly the similarity between vectors.
- In this space, look for projections onto small-dimensional spaces that better fit the known network.

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:

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

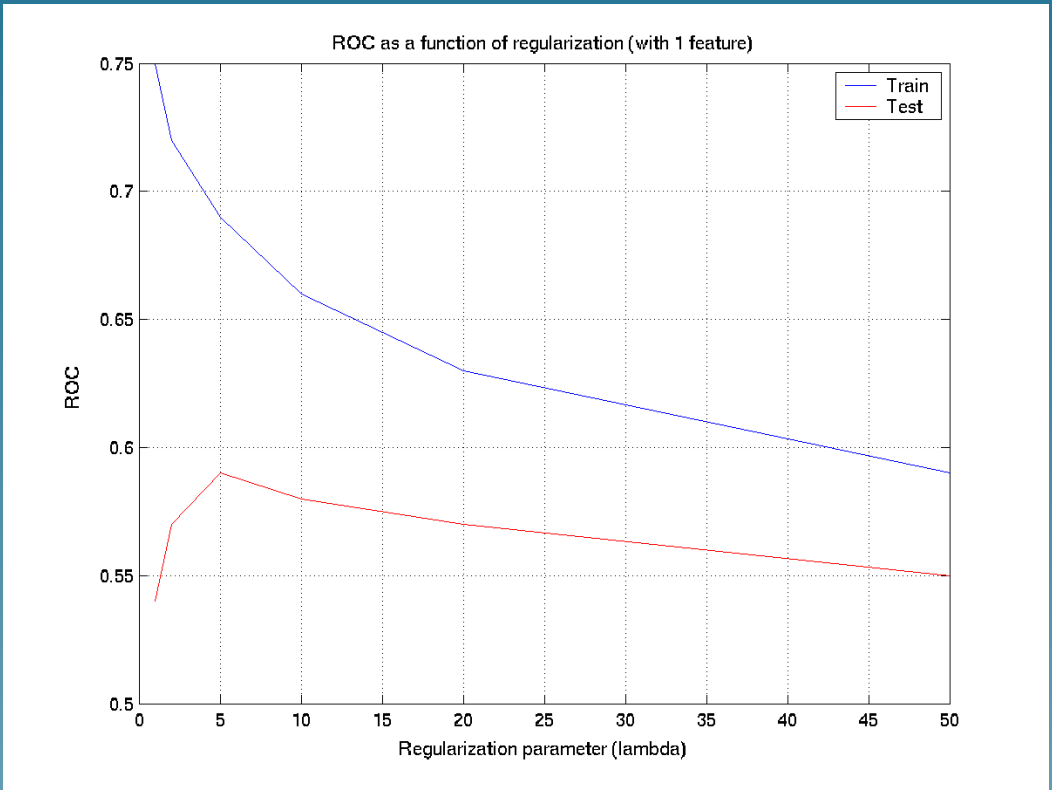
  under the constraint $\alpha^\top K_0^2 \alpha = 1$, where:

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

- It is equivalent to solving the generalized eigenvalue problem:

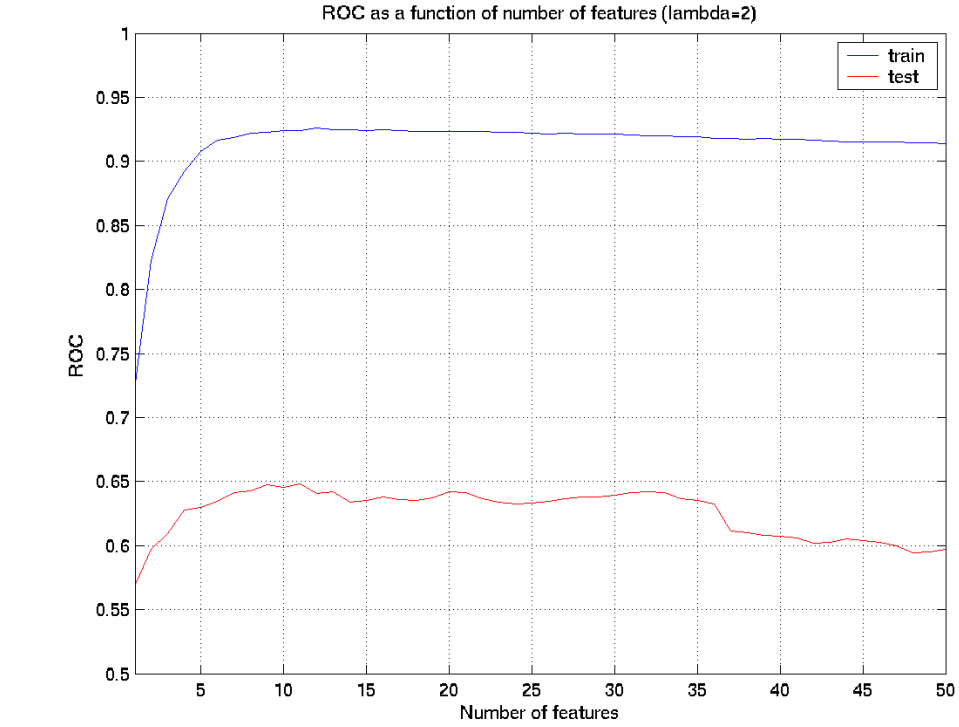
$$(LK_0 + \lambda I)\alpha = \mu K_0 \alpha.$$

Evaluation of the supervised approach: effect of λ



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

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

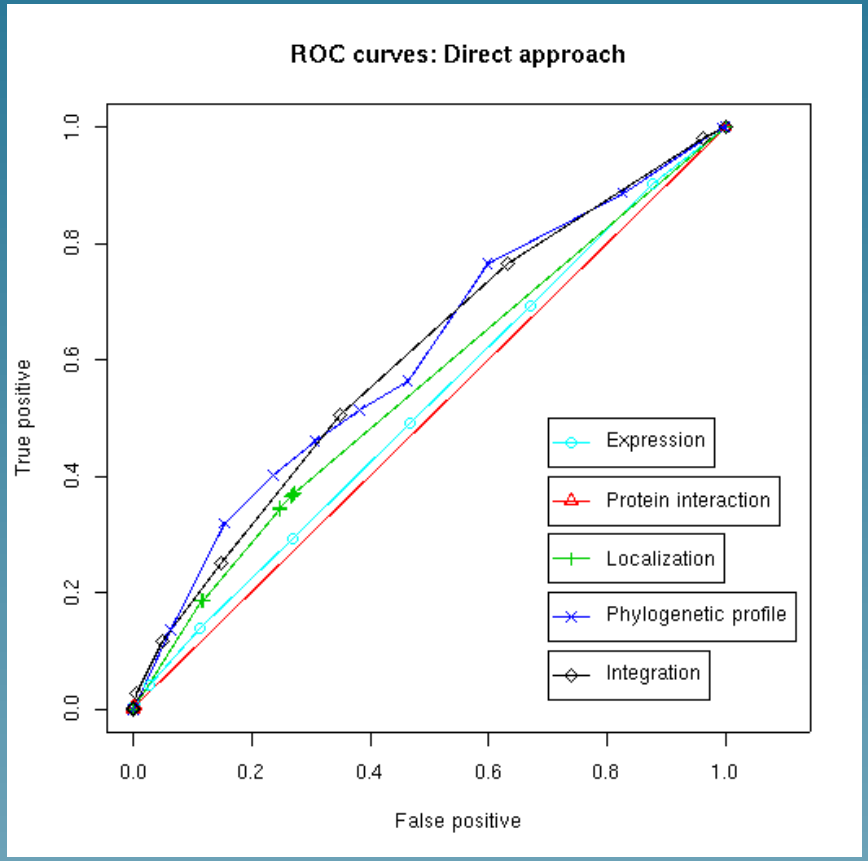


Learning from heterogeneous data

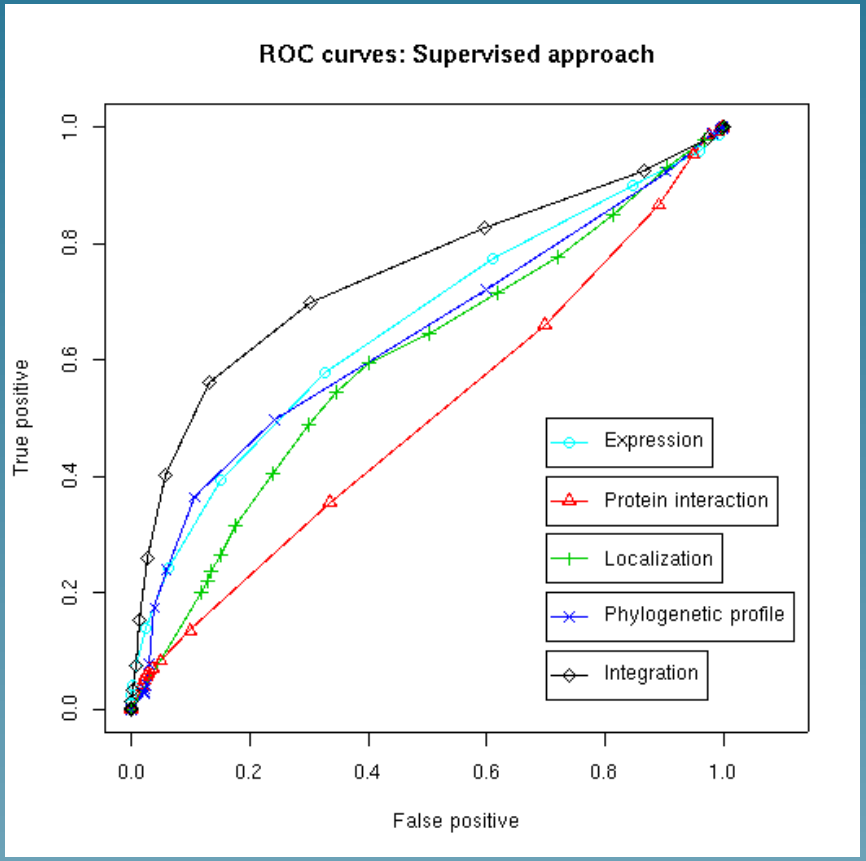
- Suppose several data are available about the genes, e.g., expression, localization, struture, predicted interaction etc...
- Each data can be represented by a kernel matrix $K_1, \ldots, K_p$
- Kernel can be combined by various operations, e.g., addition:

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

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)
- Other cost functions
- How to better integrate several similarities? (semi-definite programming?)

Part 1

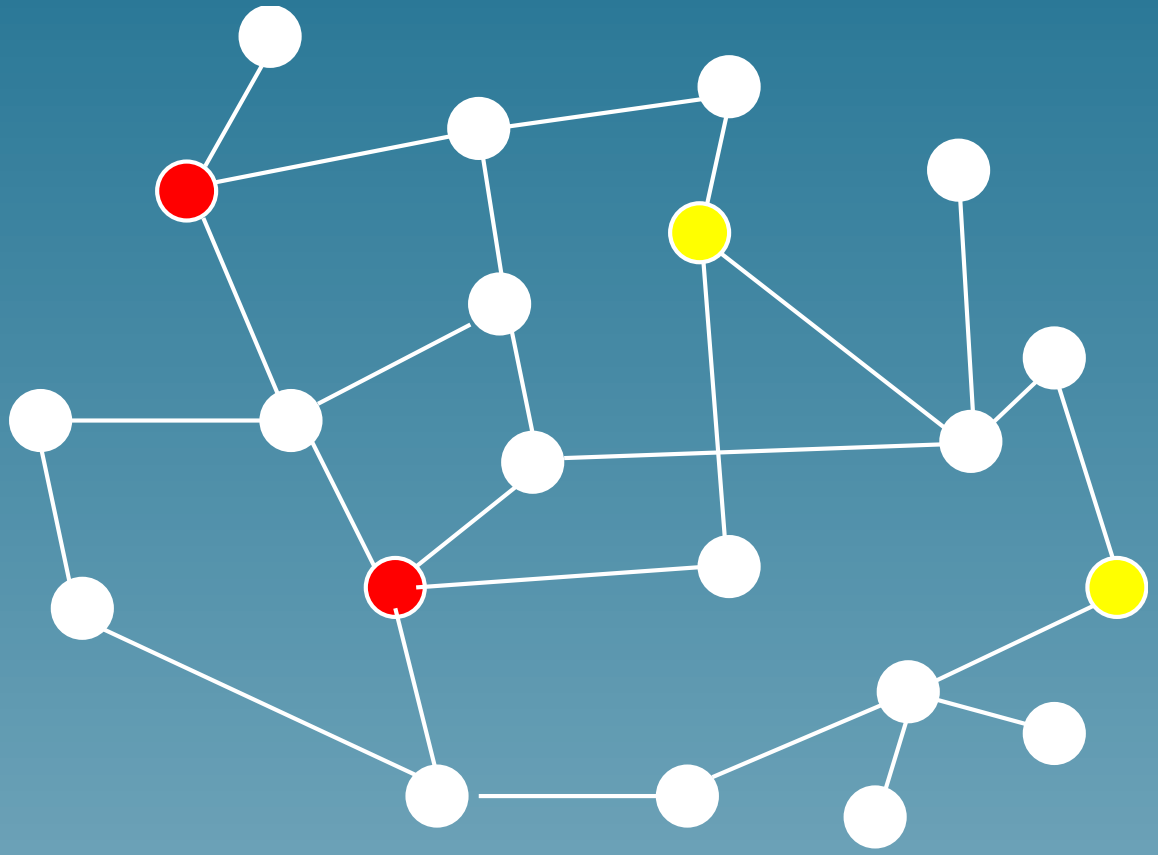
Inference on networks

(ongoing work in progress)

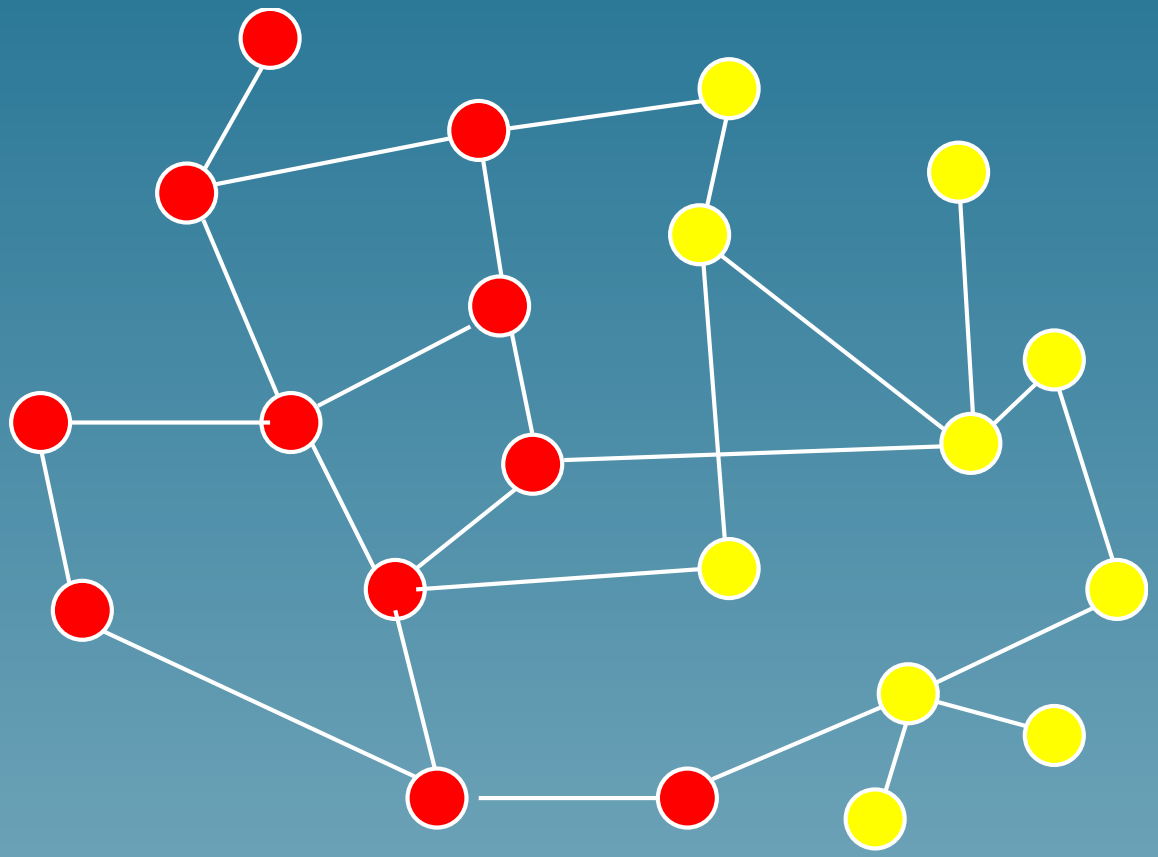
Motivations

- Data can sometimes be represented naturally as nodes of a network
- Networks are convenient to define a global structure from local similarities
- Example: close homology is easy to detect, defines the global protein similarity network (Weston et al., 2004)
- Possible applications: remote homology detection, 3D fold prediction...

Problem 2



Problem 2



General approach

- The vertices $V = V_l \cup V_u$ are either labeled (V_l) or unlabeled (V_u)
- For any function $f : V \to \mathbb{R}$, use the graph to define a "prior" functional $\Omega(f)$ (the smaller $\Omega(f)$, the more likely f.
- Define a loss function on the set of labeled vertices: $L(f(V_u), y_u)$

General approach (cont.)

- Find the best trade-off:

$$\hat{f} = \underset{f:V \to \mathbb{R}}{\arg\min} L(f(V_u), y_u) + \lambda \Omega(f)$$

- The prediction on unlabeled vertices is $\hat{f}(V_u)$

The "prior" on f

- A "likely" label assignment should vary smoothly on the graph
- A general smoothness functional for $f \in \mathbb{R}^V$ is

$$\Omega(f) = f^\top L f,$$

where L is a $n \times n$ "inverse graph kernel"

Prior examples

- Let a graph with weight $W_{i,j}$ between vertices x_i and x_j
- Let D the diagonal matrix with $D_{i,i} = \sum_j W_{i,j}$

Prior examples

- Let a graph with weight $W_{i,j}$ between vertices x_i and x_j
- Let D the diagonal matrix with $D_{i,i} = \sum_j W_{i,j}$
- Average weighted variations:

$$\Omega(f) = \sum_{x_i \sim x_j} W_{i,j} \left(f(x_i) - f(x_j)\right)^2 = f^\top (D - W) f$$

Prior examples (cont.)

- Average weighted variations:

$$\Omega(f) = \sum_{x_i \sim x_j} W_{i,j} \left(\frac{f(x_i)}{\sqrt{D_{i,i}}} - \frac{f(x_j)}{\sqrt{D_{j,j}}} \right)^2 = f^\top (I - D^{-1/2} W D^{-1/2}) f$$

- Fourier spectrum quantization:

$$\Omega(f) = f^\top e^{\beta(D-W)} f$$

Perfect regression (Zhu et al. 2003)

- f_l must fit exactly y_l:

$$\hat{f} = \underset{f:V\to\mathbb{R}, f(V_l)=y_l}{\arg\min} f^\top L f$$

- Solution:

$$f(V_u) = -L_{u,u}^{-1} L_{u,l} y_l$$

- Interpretation: probability of first hitting a certain label by a random walk on the graph starting from an unlabeled node.

Noisy regression (Belkin et al. 2003; Zhu et al. 2003)

- The loss function is mean squares:

$$\hat{f} = \arg\min_{f:V\to\mathbb{R}} \frac{1}{l} \sum i \in L \, (f(x_i - y_i)^2 + \lambda f^\top L f$$

- Solution:

$$\hat{f}(V) = (I_l + l\lambda L)^{-1} I_l y$$

- Interpretation: diffuse labels by iterating

$$f_{t+1} = (\alpha I_l + I - I_l)Lf_t + (1-\alpha)I_l y$$

Other applications

- Dimensionality reduction (Belkin et al., 2001):

$$\hat{f} = \underset{f:V\to\mathbb{R}^d, f^\top Df=1}{\arg\min} \lambda f^\top Lf$$

with solution $Lf = \mu Df$

Other applications

- Dimensionality reduction (Belkin et al., 2001):

$$\hat{f} = \underset{f:V\to\mathbb{R}^d, f^\top Df=1}{\arg\min} \lambda f^\top Lf$$

  with solution $Lf = \mu Df$

- Protein ranking (Weston et al., 2004): one positive example, all other negative, noisy regression ("label diffusion")

Conclusion

Conclusion

- A new approach to supervised network inference, many possible variants and extensions
- Inference on networks is a rapidely expanding field with impressive results. More applications to come
- Both approaches are related and could be combined.