

# Some contributions of machine learning to bioinformatics

Jean-Philippe Vert

Jean-Philippe.Vert@ensmp.fr

ParisTech, Ecole des Mines de Paris  
Institut Curie  
INSERM U900

“Informatics for Biology” course, Pasteur Institute, Paris, France,  
April 9th, 2008.

- 1 Including prior knowledge in classification and regression
- 2 Virtual screening and chemogenomics
- 3 Inference on biological networks
- 4 Conclusion

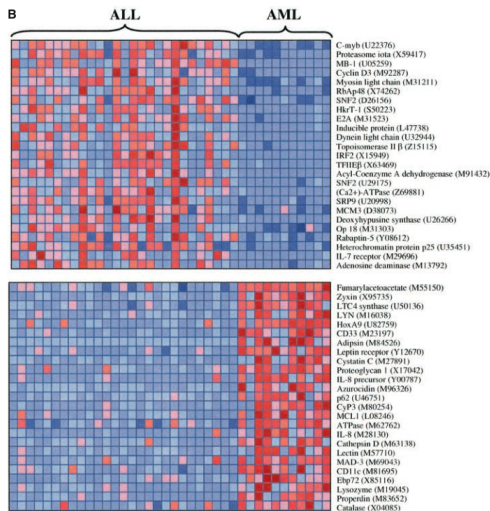
- 1 Including prior knowledge in classification and regression
- 2 Virtual screening and chemogenomics
- 3 Inference on biological networks
- 4 Conclusion

- 1 Including prior knowledge in classification and regression
- 2 Virtual screening and chemogenomics
- 3 Inference on biological networks
- 4 Conclusion

- 1 Including prior knowledge in classification and regression
- 2 Virtual screening and chemogenomics
- 3 Inference on biological networks
- 4 Conclusion

- 1 Including prior knowledge in classification and regression
- 2 Virtual screening and chemogenomics
- 3 Inference on biological networks
- 4 Conclusion

# Motivation



## Goal

- Design a **classifier** to automatically assign a class to future samples from their expression profile
- **Interpret** biologically the differences between the classes

## Difficulty

- Large dimension
- Few samples

## The model

- Each sample is represented by a vector  $x = (x_1, \dots, x_p)$
- **Goal**: estimate a linear function:

$$f_{\beta}(x) = \sum_{i=1}^p \beta_i x_i + \beta_0 .$$

- **Interpretability**: the weight  $\beta_i$  quantifies the influence of feature  $i$  (but...)



## Training the model

$$f_{\beta}(x) = \sum_{i=1}^p \beta_i x_i + \beta_0 .$$

- Minimize an **empirical risk** on the training samples:

$$\min_{\beta \in \mathbb{R}^{p+1}} R_{emp}(\beta) = \frac{1}{n} \sum_{i=1}^n l(f_{\beta}(x_i), y_i) ,$$

- ... subject to some **constraint** on  $\beta$ , e.g.:

$$\Omega(\beta) \leq C .$$

# Example : Norm Constraints

## The approach

A common method in statistics to learn with few samples in high dimension is to **constrain the Euclidean norm of  $\beta$**

$$\Omega_{ridge}(\beta) = \|\beta\|_2^2 = \sum_{i=1}^p \beta_i^2,$$

(ridge regression, support vector machines...)

### Pros

- Good performance in classification

### Cons

- Limited interpretation (small weights)
- No prior biological knowledge

# Example : Feature Selection

## The approach

Constrain most weights to be 0, i.e., **select a few genes** ( $< 100$ ) whose expression are sufficient for classification.

- Greedy feature selection (T-tests, ...)
- Constrain the norm of  $\beta$ : **LASSO** penalty ( $\|\beta\|_1 = \sum_{i=1}^p |\beta_i|$ ), **elastic net** penalty ( $\|\beta\|_1 + \|\beta\|_2$ ), ... )

## Pros

- Good performance in classification
- **Biomarker** selection
- Interpretability

## Cons

- The gene selection process is usually **not robust**
- No use of prior biological knowledge

## The idea

- If we have a specific **prior knowledge about the “correct” weights**, it can be included in  $\Omega$  in the constraint:

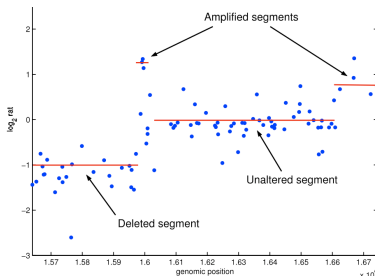
Minimize  $R_{emp}(\beta)$  subject to  $\Omega(\beta) \leq C$ .

- If we design a **convex** function  $\Omega$ , then the algorithm boils down to a convex optimization problem (usually **easy to solve**).
- Similar to priors in Bayesian statistics

# Example: CGH array classification

## The problem

- Comparative genomic hybridization (CGH) data measure the **DNA copy number** along the genome
- Very useful, in particular in cancer research
- Can we **classify CGH arrays** for diagnosis or prognosis purpose?
- **Prior knowledge**: we expect  $\beta$  to be **sparse**, and **piecewise constant** along the genome

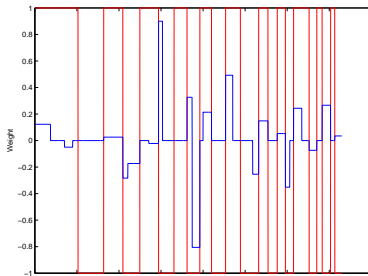
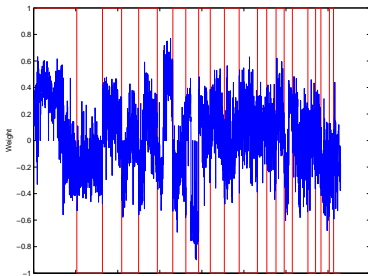


# Example: CGH array classification

A solution (Rapaport et al., 2008)

$$\Omega_{fusedlasso}(\beta) = \sum_i |\beta_i| + \sum_{i \sim j} |\beta_i - \beta_j|.$$

- Good performance on diagnosis for bladder cancer, and prognosis for melanoma.
- More interpretable classifiers



# Example: finding discriminant modules in gene networks

## The problem

- Classification of gene expression: too many genes
- **A gene network is given** (PPI, metabolic, regulatory, signaling, co-expression...)
- We expect that “clusters of genes” (**modules**) in the network contribute similarly to the classification

## Two solutions (Rapaport et al., 2007, 2008)

$$\Omega_{\text{spectral}}(\beta) = \sum_{i \sim j} (\beta_i - \beta_j)^2,$$

$$\Omega_{\text{graphfusion}}(\beta) = \sum_{i \sim j} |\beta_i - \beta_j| + \sum_i |\beta_i|.$$

# Example: finding discriminant modules in gene networks

## The problem

- Classification of gene expression: too many genes
- **A gene network is given** (PPI, metabolic, regulatory, signaling, co-expression...)
- We expect that “clusters of genes” (**modules**) in the network contribute similarly to the classification

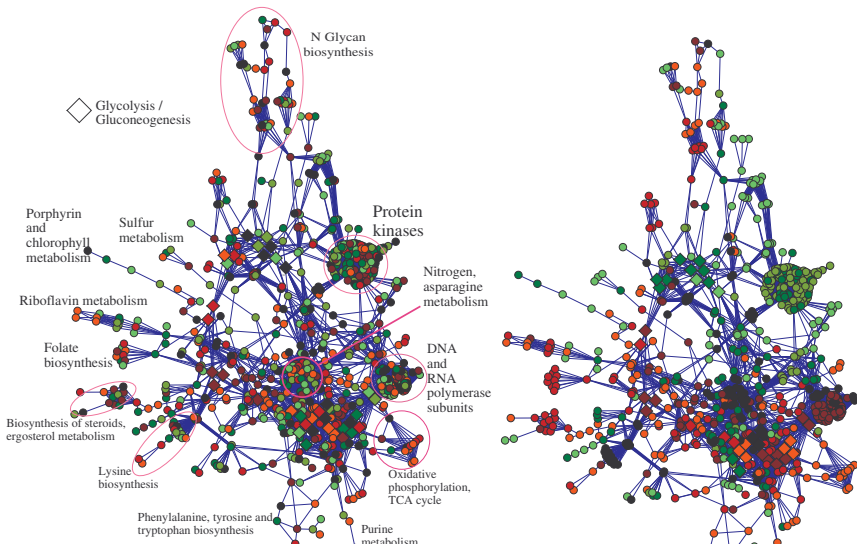
## Two solutions (Rapaport et al., 2007, 2008)

$$\Omega_{\text{spectral}}(\beta) = \sum_{i \sim j} (\beta_i - \beta_j)^2,$$

$$\Omega_{\text{graphfusion}}(\beta) = \sum_{i \sim j} |\beta_i - \beta_j| + \sum_i |\beta_i|.$$

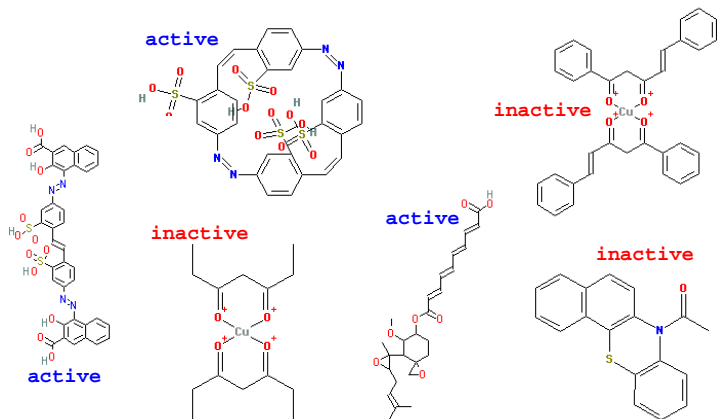


# Example: finding discriminant modules in gene networks



- 1 Including prior knowledge in classification and regression
- 2 Virtual screening and chemogenomics**
- 3 Inference on biological networks
- 4 Conclusion

# Ligand-Based Virtual Screening and QSAR



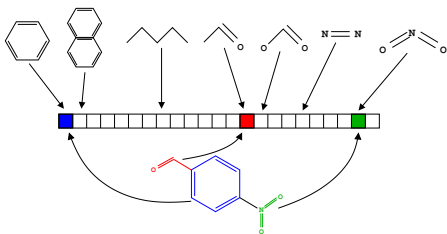
*NCI AIDS screen results (from <http://cactus.nci.nih.gov>).*

# Classical approaches

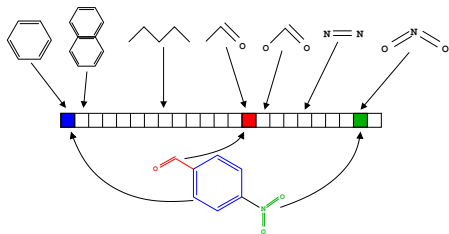
## Two steps

- 1 Map each molecule to a **vector of fixed dimension** using **molecular descriptors**
  - Global properties of the molecules (mass, logP...)
  - 2D and 3D descriptors (substructures, fragments, ....)
- 2 Apply an algorithm for **regression or pattern recognition**.
  - PLS, ANN, ...

## Example: 2D structural keys



# Which descriptors?



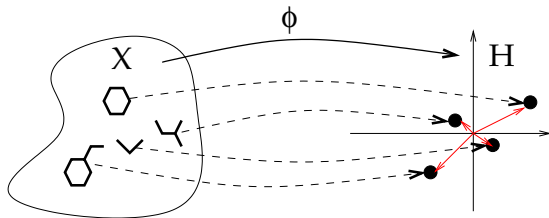
## Difficulties

- **Many** descriptors are **needed** to characterize various features (in particular for 2D and 3D descriptors)
- But **too many** descriptors are **harmful** for memory storage, computation speed, statistical estimation

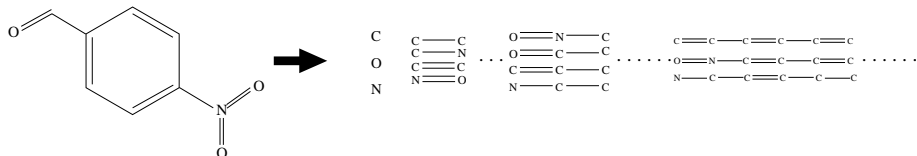
## Definition

- Let  $\Phi(x) = (\Phi_1(x), \dots, \Phi_p(x))$  be a vector representation of the molecule  $x$
- The **kernel** between two molecules is defined by:

$$K(x, x') = \Phi(x)^\top \Phi(x') = \sum_{i=1}^p \Phi_i(x) \Phi_i(x').$$



# Example: 2D fragment kernel



- $\phi_d(x)$  is the vector of counts of **all fragments of length  $d$** :

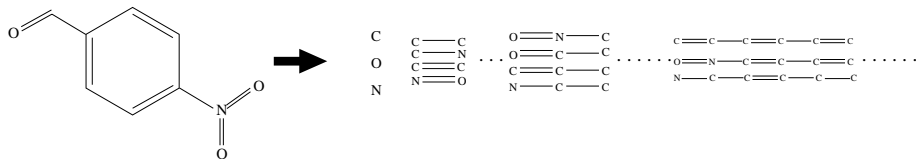
$$\phi_1(x) = (\#(C), \#(O), \#(N), \dots)^T$$

$$\phi_2(x) = (\#(C-C), \#(C=O), \#(C-N), \dots)^T \quad \text{etc...}$$

- The **2D fragment kernel** is defined, for  $\lambda < 1$ , by

$$K_{\text{fragment}}(x, x') = \sum_{d=1}^{\infty} r(\lambda) \phi_d(x)^T \phi_d(x').$$

# Example: 2D fragment kernel

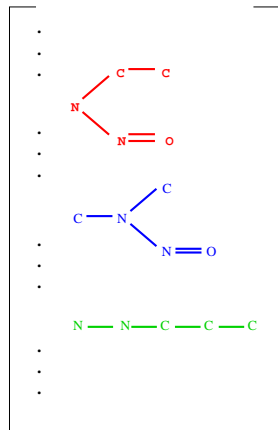
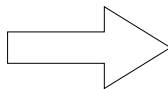
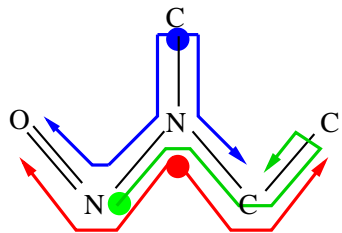


## In practice

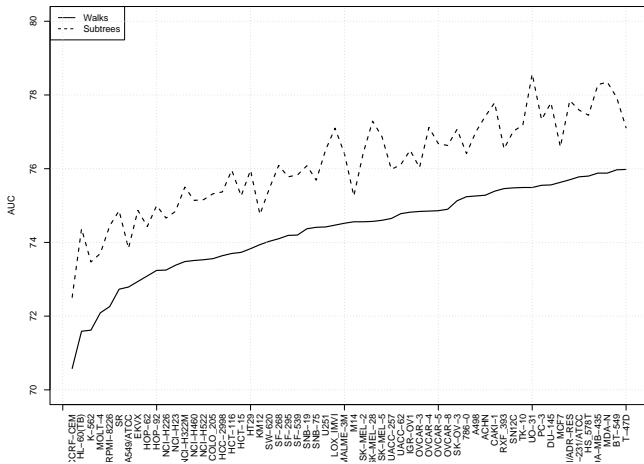
- $K_{fragment}$  can be **computed efficiently** (geometric kernel, random walk kernel...) although the feature space has **infinite dimension**.
- Increasing the **specificity of atom labels** improves performance
- Selecting only **“non-tottering” fragments** can be done efficiently and improves performance.



# Example: 2D subtree kernel

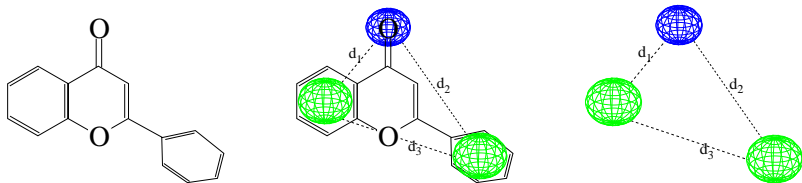


# 2D Subtree vs fragment kernels (Mahé and V, 2007)



Screening of inhibitors for 60 cancer cell lines (from Mahé and V., 2008)

# Example: 3D pharmacophore kernel (Mahé et al., 2005)



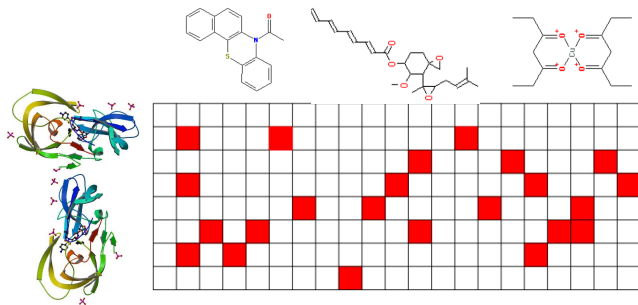
$$K(x, y) = \sum_{p_x \in \mathcal{P}(x)} \sum_{p_y \in \mathcal{P}(y)} \exp(-\gamma d(p_x, p_y)) .$$

## Results (accuracy)

Kernel	BZR	COX	DHFR	ER
2D (Tanimoto)	71.2	63.0	76.9	77.1
3D fingerprint	75.4	67.0	76.9	78.6
3D not discretized	<b>76.4</b>	<b>69.8</b>	<b>81.9</b>	<b>79.8</b>

## The problem

- Similar targets bind similar ligands
- Instead of focusing on each target individually, can we screen the biological space (target families) vs the chemical space (ligands)?
- Mathematically, learn  $f(\text{target}, \text{ligand}) \in \{\text{bind}, \text{notbind}\}$



## Tensor product SVM

- Take the kernel:

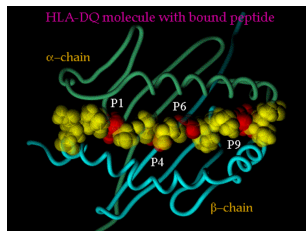
$$K((t, l), (t', l')) = K_t(t, t')K_l(l, l').$$

- Equivalently, represent a pair  $(t, l)$  by the vector  $\phi_t(t) \otimes \phi_l(l)$
- Allows to use **any** kernel for proteins  $K_t$  with **any** kernel for small molecules  $K_l$
- When  $K_t$  is the **Dirac** kernel, we recover the **classical paradigm**: each target is treated independently from the others.
- Otherwise, information is **shared across targets**. The more similar the targets, the more they share information.

# Example: MHC-I epitope prediction across different alleles

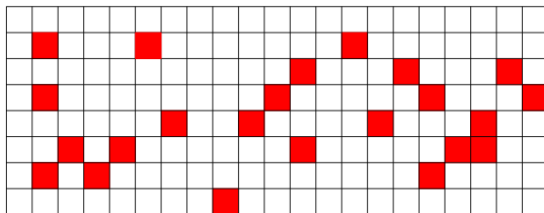
## The approach (Jacob and V., 2007)

- take a kernel to compare different MHC-I alleles (e.g., based on the amino-acids in the peptide recognition pocket)
- take a kernel to compare different epitopes (9-mer peptides)
- Combine them to learn the  $f(\text{allele}, \text{epitope})$  function
- State-of-the-art performance
- Available at <http://cbio.ensmp.fr/kiss>



# Generalization: collaborative filtering with attributes

- General problem: learn  $f(x, y)$  with a kernel  $K_x$  for  $x$  and a kernel  $K_y$  for  $y$ .
- SVM with a tensor product kernel  $K_x \otimes K_y$  is a particular case of something more general: estimating an operator with a spectral regularization.
- Other spectral regularization are possible (e.g., trace norm) and lead to efficient algorithms
- More details in Abernethy et al. (2008).

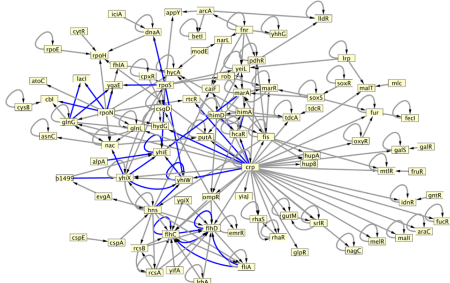
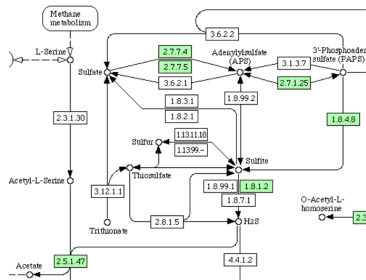
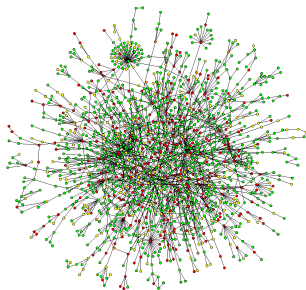


# Outline

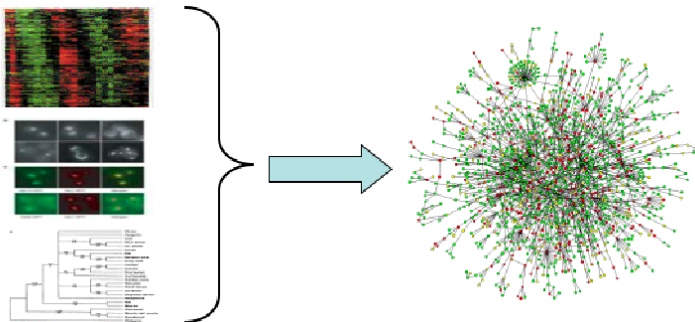
- 1 Including prior knowledge in classification and regression
- 2 Virtual screening and chemogenomics
- 3 Inference on biological networks**
- 4 Conclusion



# Biological networks



# Our goal



## Data

- Gene expression,
- Gene sequence,
- Protein localization, ...

## Graph

- Protein-protein interactions,
- Metabolic pathways,
- Signaling pathways, ...

## “De novo” inference

- Given data about individual genes and proteins
- Infer the edges between genes and proteins

## “Supervised” inference

- Given data about individual genes and proteins
- **and** given some known interactions
- infer unknown interactions

## “De novo” inference

- Given data about individual genes and proteins
- Infer the edges between genes and proteins

## “Supervised” inference

- Given data about individual genes and proteins
- **and** given some known interactions
- infer unknown interactions

- 1 Most methods developed so far are “**de novo**” (e.g., co-expression, Bayesian networks, mutual information nets, dynamical systems...)
- 2 However most **real-world** application fit the “**supervised**” framework
- 3 Solving the “supervised” problem is **much easier** (and more efficient) than the “de novo” problem. It requires less hypothesis.

## Typical strategies

- Fit a **dynamical system** to time series (e.g., PDE, boolean networks, state-space models)
- Detect **statistical conditional independence or dependency** (Bayesian network, mutual information networks, co-expression)

## Pros

- **Excellent approach** if the model is correct and enough data are available
- **Interpretability** of the model
- Inclusion of **prior knowledge**

## Cons

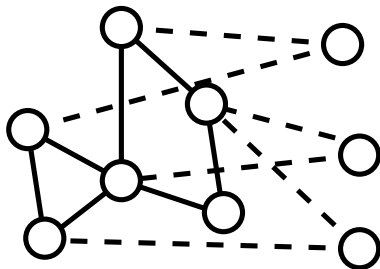
- **Specific** to particular data and networks
- **Needs a correct model!**
- Difficult **integration** of heterogeneous data
- Often needs a **lot of data** and long computation time

# Supervised methods

## Motivation

In actual applications,

- we know in advance parts of the network to be inferred
- the problem is to add/remove nodes and edges using genomic data as side information



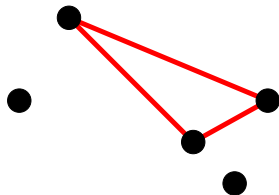
## Supervised method

- Given genomic data **and** the currently known network...
- Infer **missing edges** between current nodes and additional nodes.

# Supervised approach by Metric learning

## Idea

- The direct similarity-based method fails because the **distance metric used might not be adapted** to the inference of the targeted protein network.
- Solution: use the **known subnetwork** to **refine the distance measure**, before applying the similarity-based method
- Examples: **kernels CCA** (Yamanishi et al. 2004), **kernel metric learning** (V and Yamanishi, 2005)

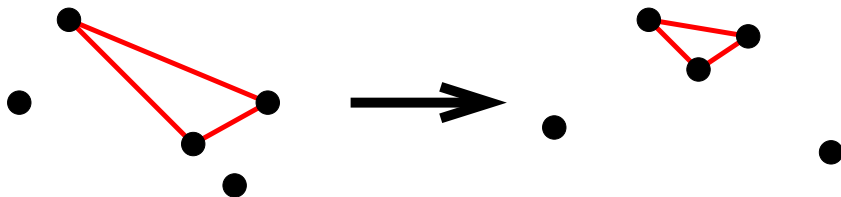




# Supervised approach by Metric learning

## Idea

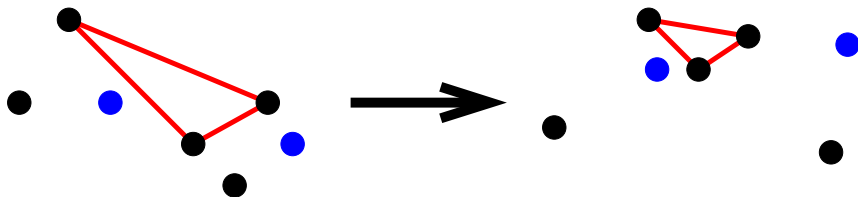
- The direct similarity-based method fails because the **distance metric used might not be adapted** to the inference of the targeted protein network.
- Solution: use the **known subnetwork** to **refine the distance measure**, before applying the similarity-based method
- Examples: **kernels CCA** (Yamanishi et al. 2004), **kernel metric learning** (V and Yamanishi, 2005)



# Supervised approach by Metric learning

## Idea

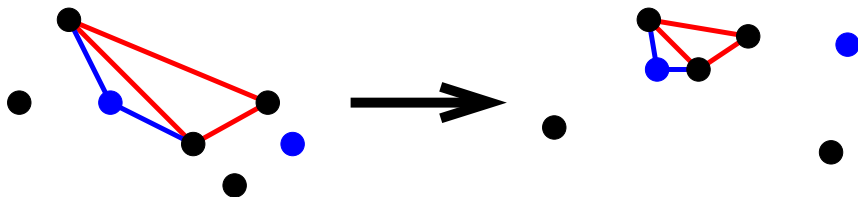
- The direct similarity-based method fails because the **distance metric used might not be adapted** to the inference of the targeted protein network.
- Solution: use the **known subnetwork** to **refine the distance measure**, before applying the similarity-based method
- Examples: **kernels CCA** (Yamanishi et al. 2004), **kernel metric learning** (V and Yamanishi, 2005)



# Supervised approach by Metric learning

## Idea

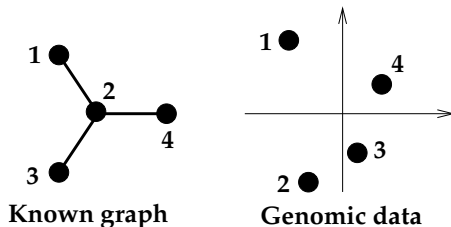
- The direct similarity-based method fails because the **distance metric used might not be adapted** to the inference of the targeted protein network.
- Solution: use the **known subnetwork** to **refine the distance measure**, before applying the similarity-based method
- Examples: **kernels CCA** (Yamanishi et al. 2004), **kernel metric learning** (V and Yamanishi, 2005)



# Supervised inference by pattern recognition

## Formulation and basic issue

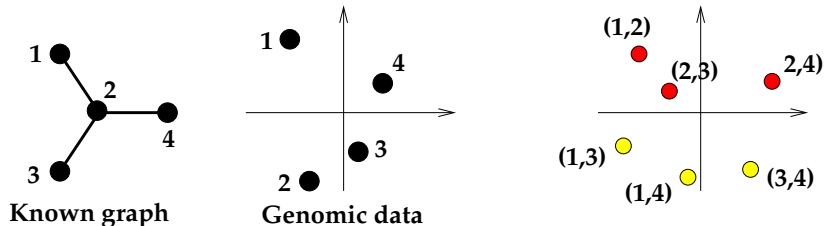
- A pair can be **connected (1)** or **not connected (-1)**
- From the known subgraph we can **extract examples** of connected and non-connected pairs
- However the genomic data characterize **individual** proteins; we need to work with **pairs** of proteins instead!



# Supervised inference by pattern recognition

## Formulation and basic issue

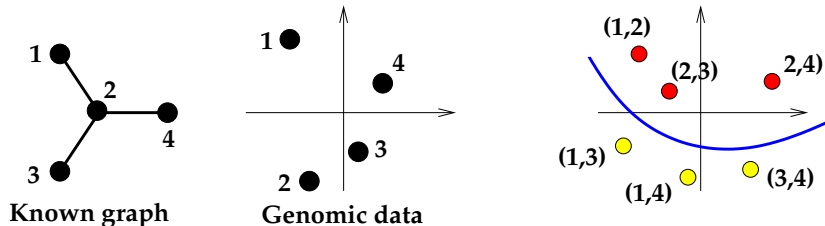
- A pair can be **connected (1)** or **not connected (-1)**
- From the known subgraph we can **extract examples** of connected and non-connected pairs
- However the genomic data characterize **individual** proteins; we need to work with **pairs** of proteins instead!



# Supervised inference by pattern recognition

## Formulation and basic issue

- A pair can be **connected (1)** or **not connected (-1)**
- From the known subgraph we can **extract examples** of connected and non-connected pairs
- However the genomic data characterize **individual** proteins; we need to work with **pairs** of proteins instead!



# Tensor product SVM (Ben-Hur and Noble, 2006)

- **Intuition:** a pair  $(A, B)$  is similar to a pair  $(C, D)$  if:
  - $A$  is similar to  $C$  **and**  $B$  is similar to  $D$ , **or**...
  - $A$  is similar to  $D$  **and**  $B$  is similar to  $C$
- **Formally**, define a similarity between pairs from a similarity between individuals by

$$K_{TPPK}((a, b), (c, d)) = K(a, c)K(b, d) + K(a, d)K(b, c) .$$

- If  $K$  is a positive definite kernel for individuals then  $K_{TPPK}$  is a p.d. kernel for pairs which can be used by SVM
- This amounts to representing a pair  $(a, b)$  by the **symmetrized tensor product**:

$$(a, b) \rightarrow (a \otimes b) \oplus (b \otimes a) .$$

# Tensor product SVM (Ben-Hur and Noble, 2006)

- **Intuition:** a pair  $(A, B)$  is similar to a pair  $(C, D)$  if:
  - $A$  is similar to  $C$  **and**  $B$  is similar to  $D$ , **or**...
  - $A$  is similar to  $D$  **and**  $B$  is similar to  $C$
- **Formally**, define a similarity between pairs from a similarity between individuals by

$$K_{TPPK}((a, b), (c, d)) = K(a, c)K(b, d) + K(a, d)K(b, c) .$$

- If  $K$  is a positive definite kernel for individuals then  $K_{TPPK}$  is a p.d. kernel for pairs which can be used by SVM
- This amounts to representing a pair  $(a, b)$  by the **symmetrized tensor product**:

$$(a, b) \rightarrow (a \otimes b) \oplus (b \otimes a) .$$



# Tensor product SVM (Ben-Hur and Noble, 2006)

- **Intuition:** a pair  $(A, B)$  is similar to a pair  $(C, D)$  if:
  - $A$  is similar to  $C$  **and**  $B$  is similar to  $D$ , **or**...
  - $A$  is similar to  $D$  **and**  $B$  is similar to  $C$
- **Formally**, define a similarity between pairs from a similarity between individuals by

$$K_{TPPK}((a, b), (c, d)) = K(a, c)K(b, d) + K(a, d)K(b, c) .$$

- If  $K$  is a positive definite kernel for individuals then  $K_{TPPK}$  is a p.d. kernel for pairs which can be used by SVM
- This amounts to representing a pair  $(a, b)$  by the **symmetrized tensor product**:

$$(a, b) \rightarrow (a \otimes b) \oplus (b \otimes a) .$$

# Metric learning pairwise SVM (V. et al, 2007)

- **Intuition:** a pair  $(A, B)$  is similar to a pair  $(C, D)$  if:
  - $A - B$  is similar to  $C - D$ , **or...**
  - $A - B$  is similar to  $D - C$ .
- **Formally**, define a similarity between pairs from a similarity between individuals by

$$K_{MLPK}((a, b), (c, d)) = (K(a, c) + K(b, d) - K(a, d) - K(b, c))^2 .$$

- If  $K$  is a positive definite kernel for individuals then  $K_{MLPK}$  is a p.d. kernel for pairs which can be used by SVM
- This amounts to representing a pair  $(a, b)$  by the **symmetrized difference**:

$$(a, b) \rightarrow (a - b)^{\otimes 2} .$$

# Metric learning pairwise SVM (V. et al, 2007)

- **Intuition:** a pair  $(A, B)$  is similar to a pair  $(C, D)$  if:
  - $A - B$  is similar to  $C - D$ , or...
  - $A - B$  is similar to  $D - C$ .
- **Formally**, define a similarity between pairs from a similarity between individuals by

$$K_{MLPK}((a, b), (c, d)) = (K(a, c) + K(b, d) - K(a, d) - K(b, c))^2 .$$

- If  $K$  is a positive definite kernel for individuals then  $K_{MLPK}$  is a p.d. kernel for pairs which can be used by SVM
- This amounts to representing a pair  $(a, b)$  by the symmetrized difference:

$$(a, b) \rightarrow (a - b)^{\otimes 2} .$$

- **Intuition:** a pair  $(A, B)$  is similar to a pair  $(C, D)$  if:
  - $A - B$  is similar to  $C - D$ , **or...**
  - $A - B$  is similar to  $D - C$ .
- **Formally**, define a similarity between pairs from a similarity between individuals by

$$K_{MLPK}((a, b), (c, d)) = (K(a, c) + K(b, d) - K(a, d) - K(b, c))^2 .$$

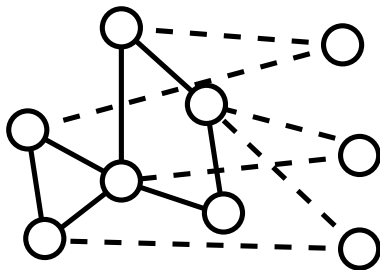
- If  $K$  is a positive definite kernel for individuals then  $K_{MLPK}$  is a p.d. kernel for pairs which can be used by SVM
- This amounts to representing a pair  $(a, b)$  by the **symmetrized difference**:

$$(a, b) \rightarrow (a - b)^{\otimes 2} .$$

# Supervised inference with local models

## The idea (Bleakley et al., 2007)

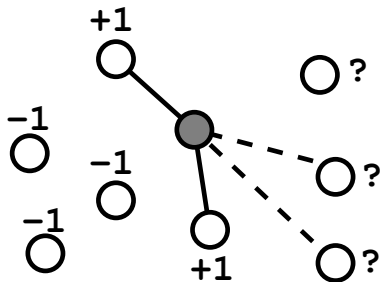
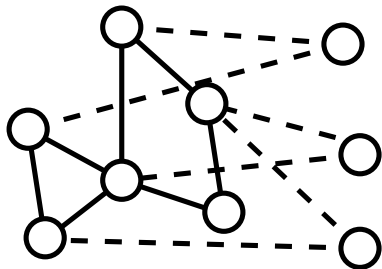
- Motivation: define **specific models** for **each target node** to discriminate between its neighbors and the others
- Treat each node independently from the other. Then **combine** predictions for ranking candidate edges.



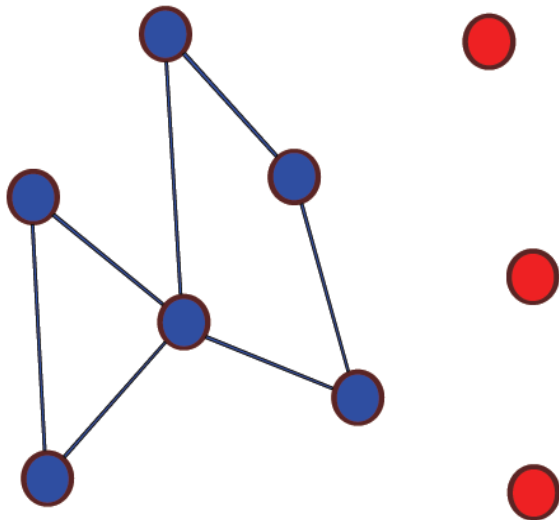
# Supervised inference with local models

## The idea (Bleakley et al., 2007)

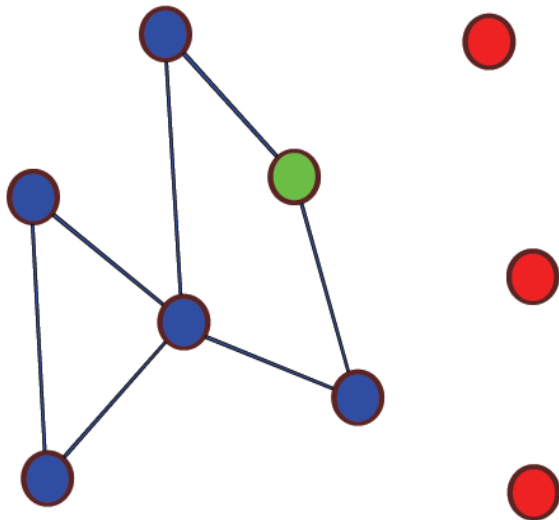
- Motivation: define **specific models** for **each target node** to discriminate between its neighbors and the others
- Treat each node independently from the other. Then **combine** predictions for ranking candidate edges.



# The LOCAL model

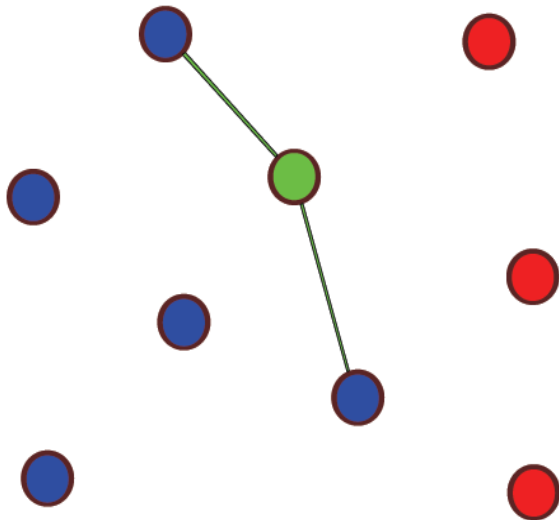


# The LOCAL model

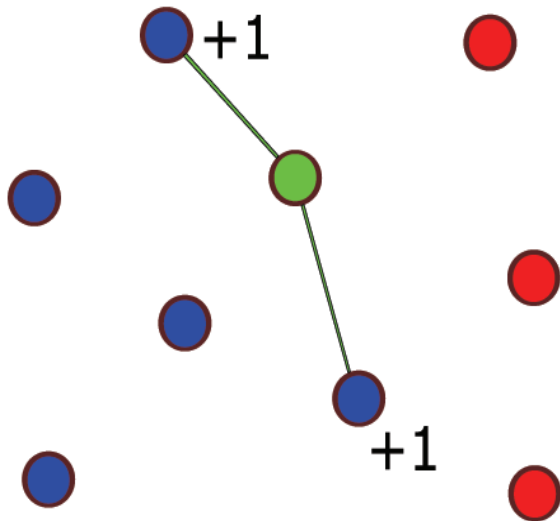




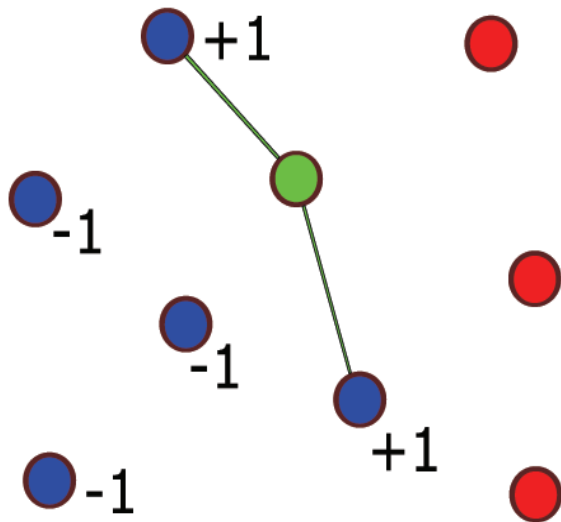
# The LOCAL model



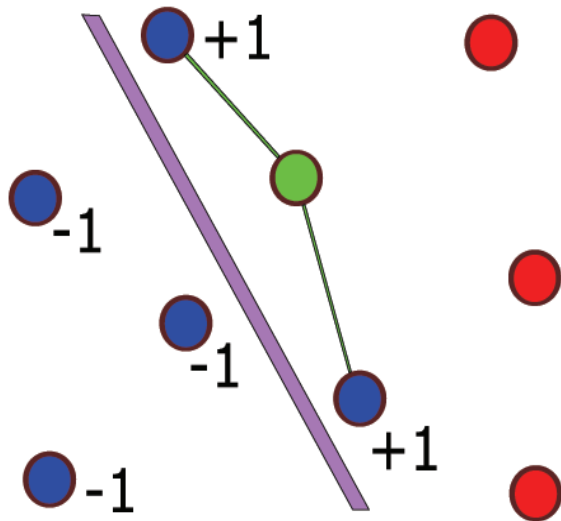
# The LOCAL model



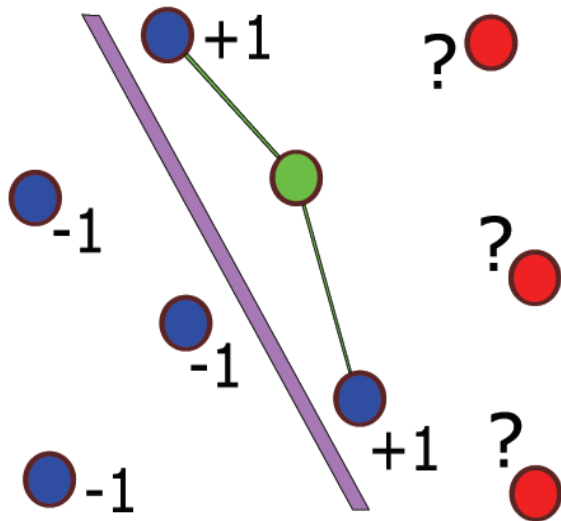
# The LOCAL model



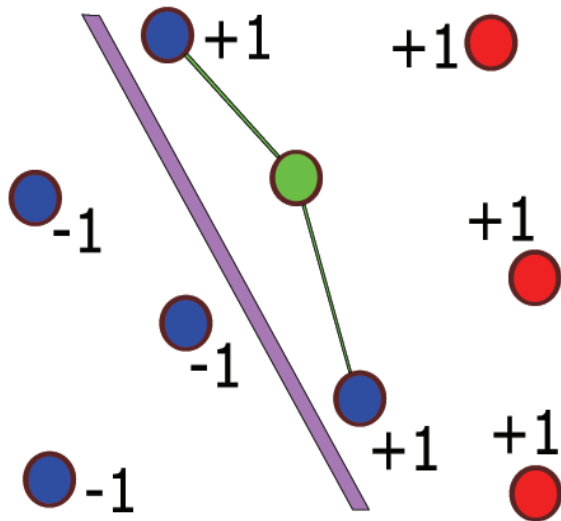
# The LOCAL model



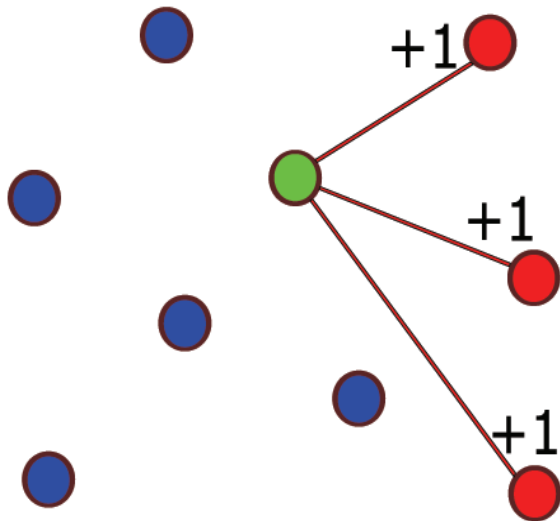
# The LOCAL model



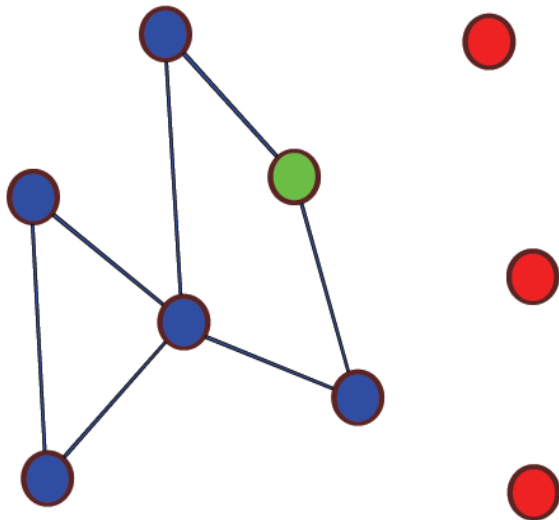
# The LOCAL model



# The LOCAL model

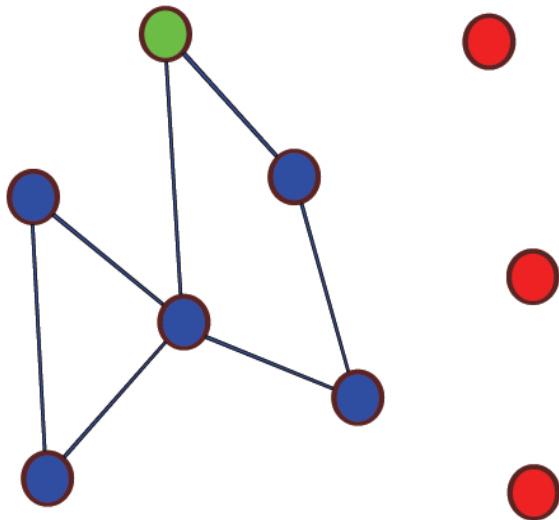


# The LOCAL model

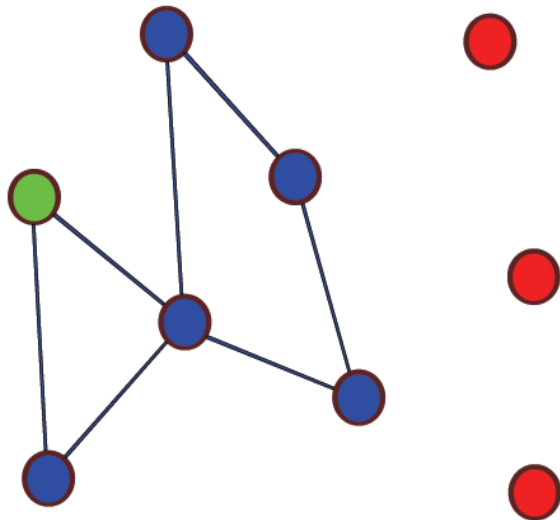




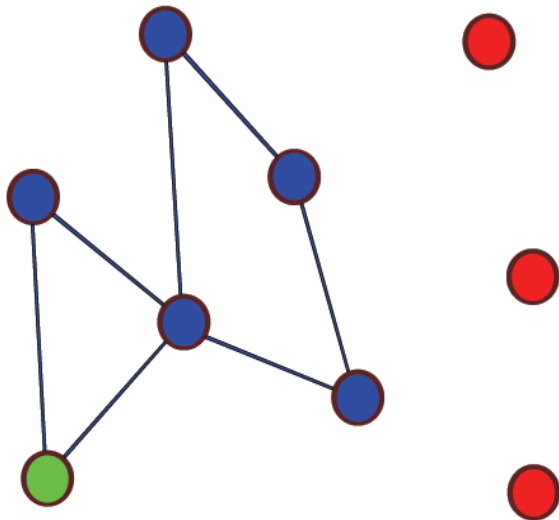
# The LOCAL model



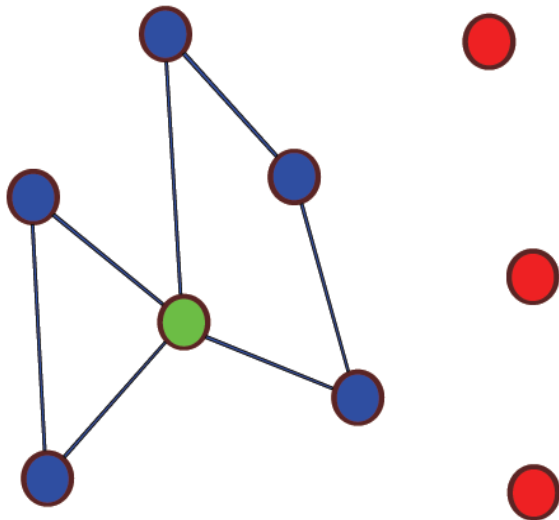
# The LOCAL model



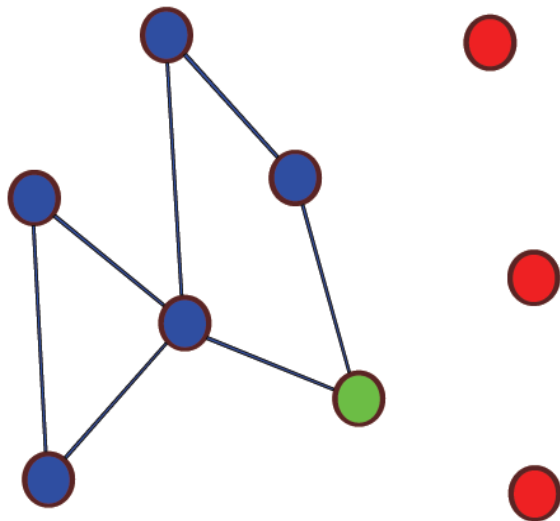
# The LOCAL model



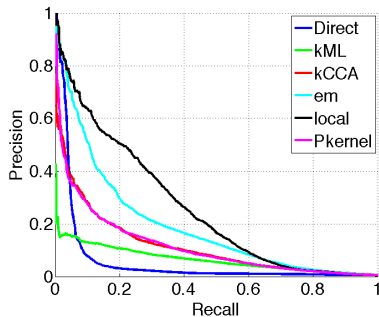
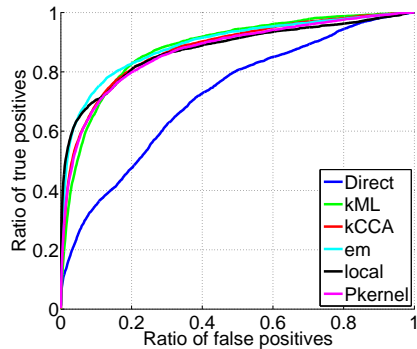
# The LOCAL model



# The LOCAL model

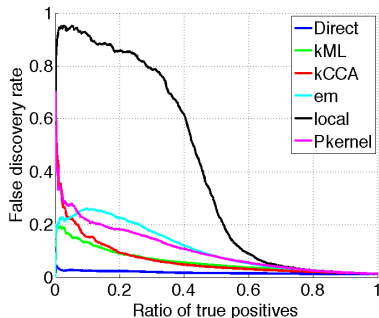
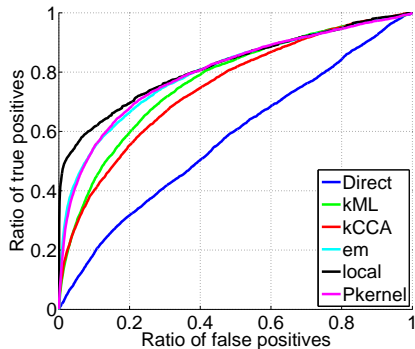


# Results: protein-protein interaction (yeast)



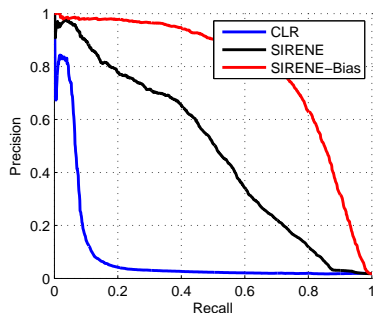
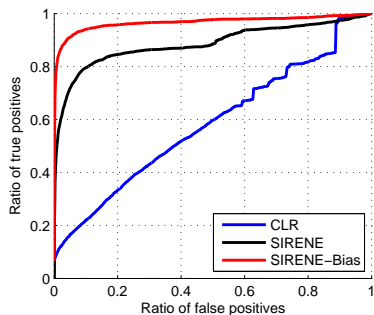
(from Bleakley et al., 2007)

# Results: metabolic gene network (yeast)



(from Bleakley et al., 2007)

# Results: regulatory network (E. coli)



Method	Recall at 60%	Recall at 80%
SIRENE	<b>44.5%</b>	<b>17.6%</b>
CLR	7.5%	5.5%
Relevance networks	4.7%	3.3%
ARACNe	1%	0%
Bayesian network	1%	0%

SIRENE = Supervised Inference of REgulatory Networks (Mordelet and V., 2008)





# Outline

- 1 Including prior knowledge in classification and regression
- 2 Virtual screening and chemogenomics
- 3 Inference on biological networks
- 4 Conclusion**

- Modern machine learning methods for regression / classification lend themselves well to the **integration of prior knowledge** in the penalization / regularization function, in particular for feature selection / grouping. Applications in **array CGH classification, siRNA design, microarray classification with gene networks**
- Kernel methods (eg SVM) allow to manipulate complex objects (eg molecules, biological sequences) as soon as **kernels can be defined and computed**. Applications in **virtual screening, QSAR, chemogenomics**.
- Inference of **biological networks** can be formulated as a **supervised problem** if the graph is partly known, and powerful methods can be applied. Application in **PPI, metabolic and regulatory networks inference**.

# People I need to thank

## Including prior knowledge in penalization

Franck Rapaport, Emmanuel Barillot, Andrei Zynoviev, Christian Lajaunie, Yves Vandembrouck, Nicolas Foveau...

## Virtual screening, kernels etc..

Pierre Mahé, Laurent Jacob, Liva Ralaivola, Véronique Stoven, Brice Hoffman, Martial Hue, Francis Bach, Jacob Abernethy, Theos Evgeniou...

## Network inference

Kevin Bleakley, Fantine Mordelet, Yoshihiro Yamanihi, Gérard Biau, Minoru Kanehisa, William Noble, Jian Qiu...