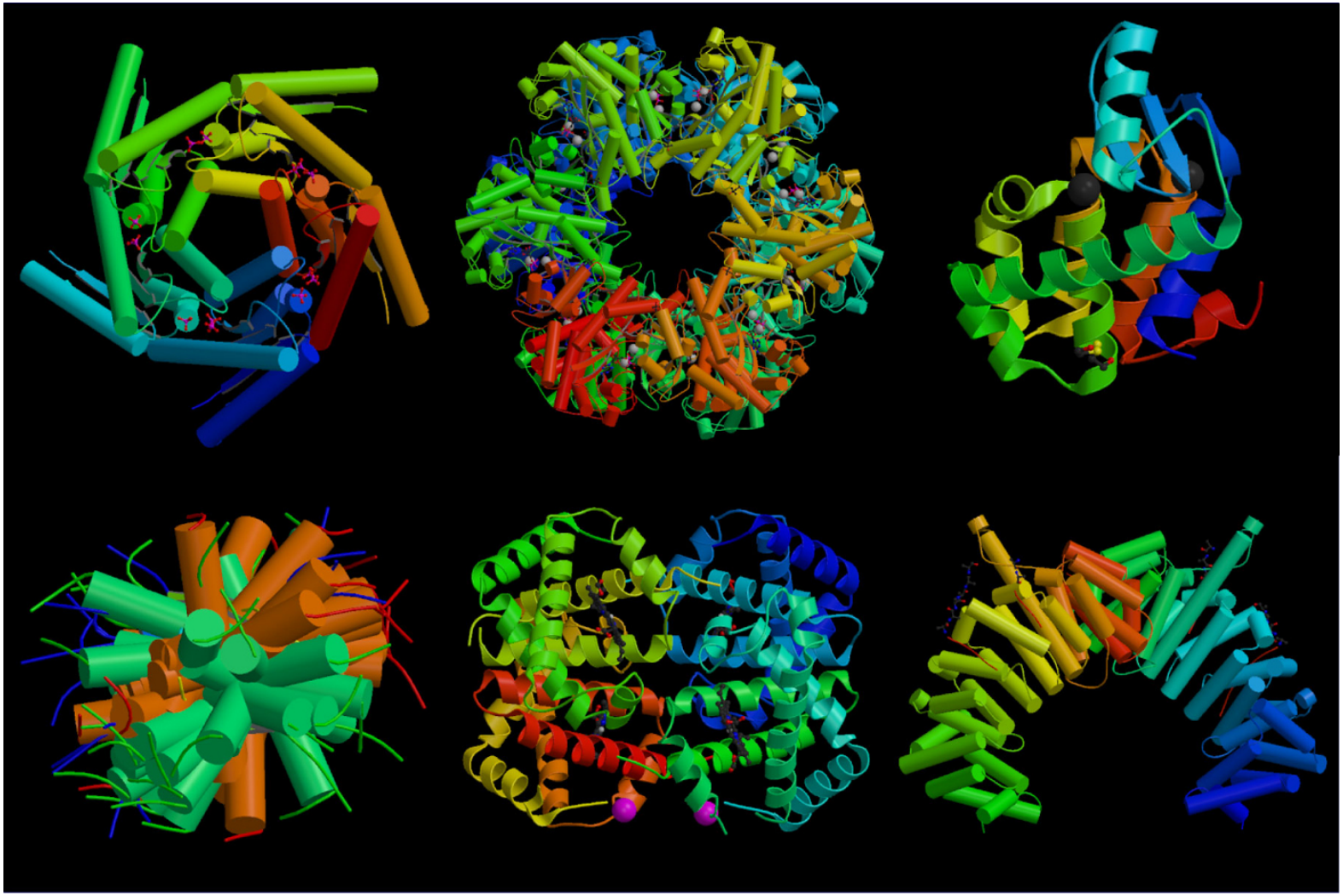
Inferring and using biological networks

Jean-Philippe Vert
Jean-Philippe.Vert@mines-paristech.fr

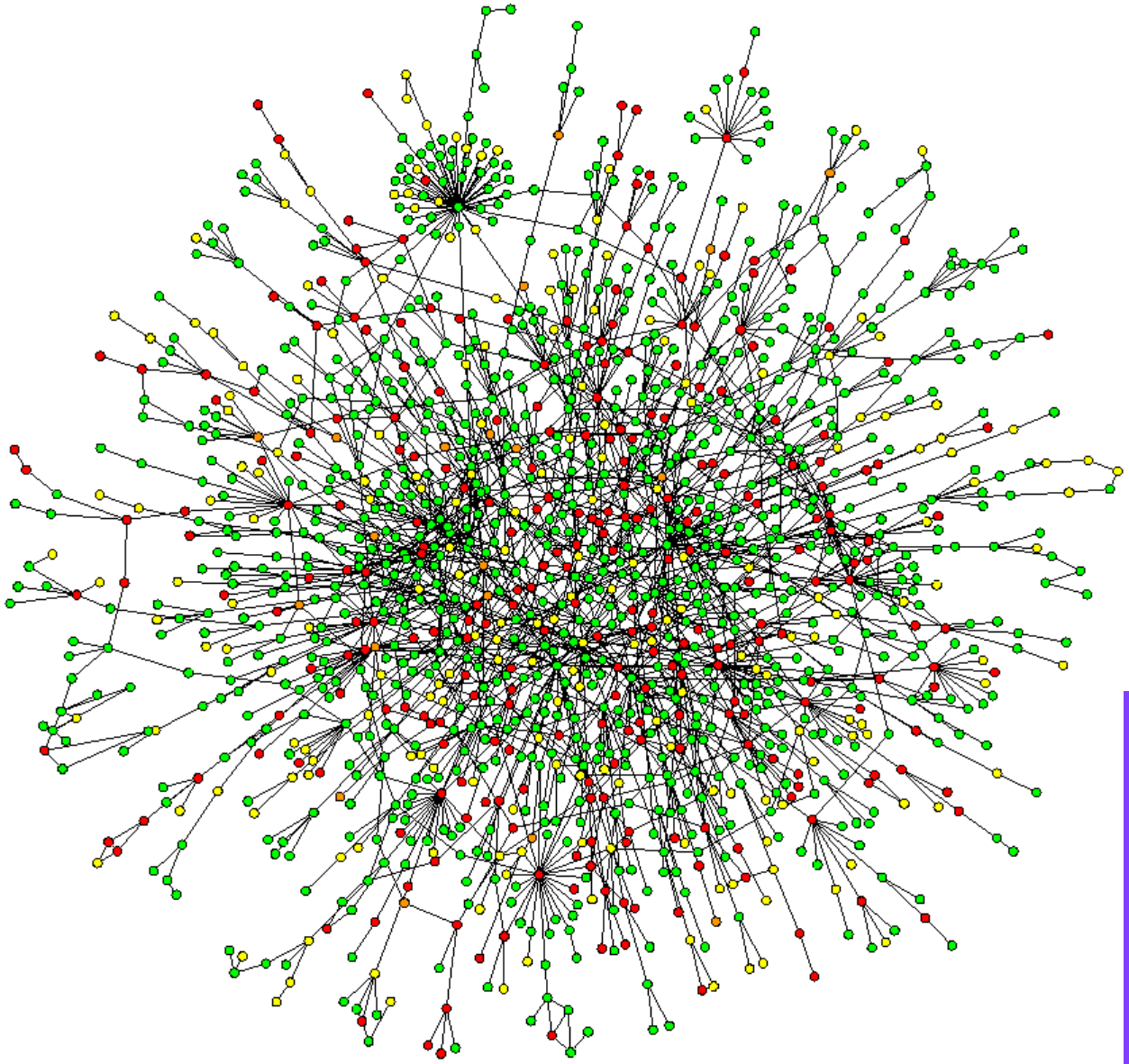
Mines ParisTech / Institut Curie / INSERM U900

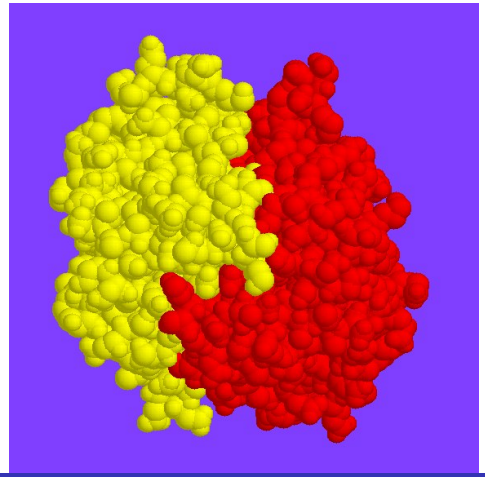
Human Genome Center, Institute of Medical Science, University of Tokyo, August 4, 2009.

We have many genes and proteins..

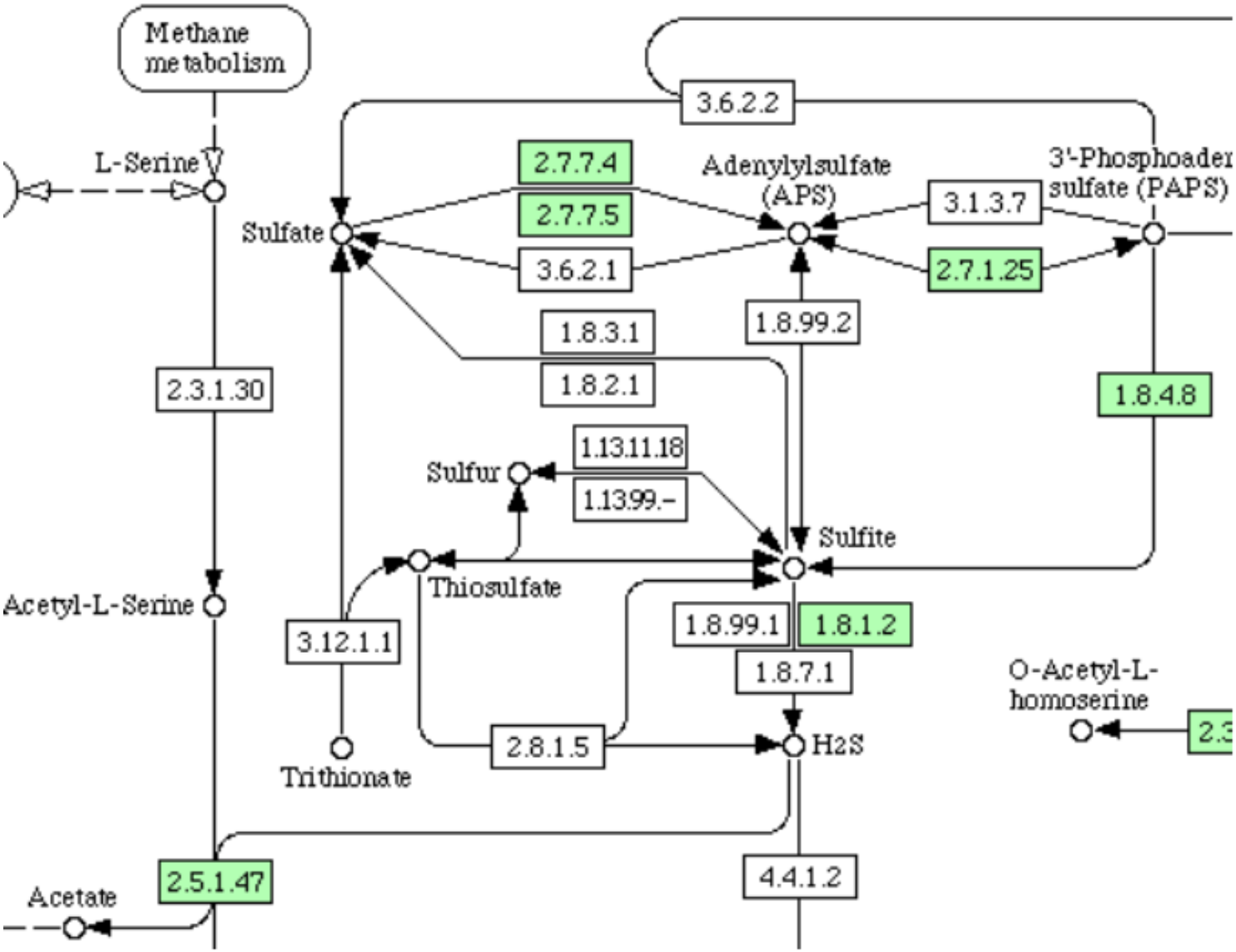


Network 1: protein-protein interaction

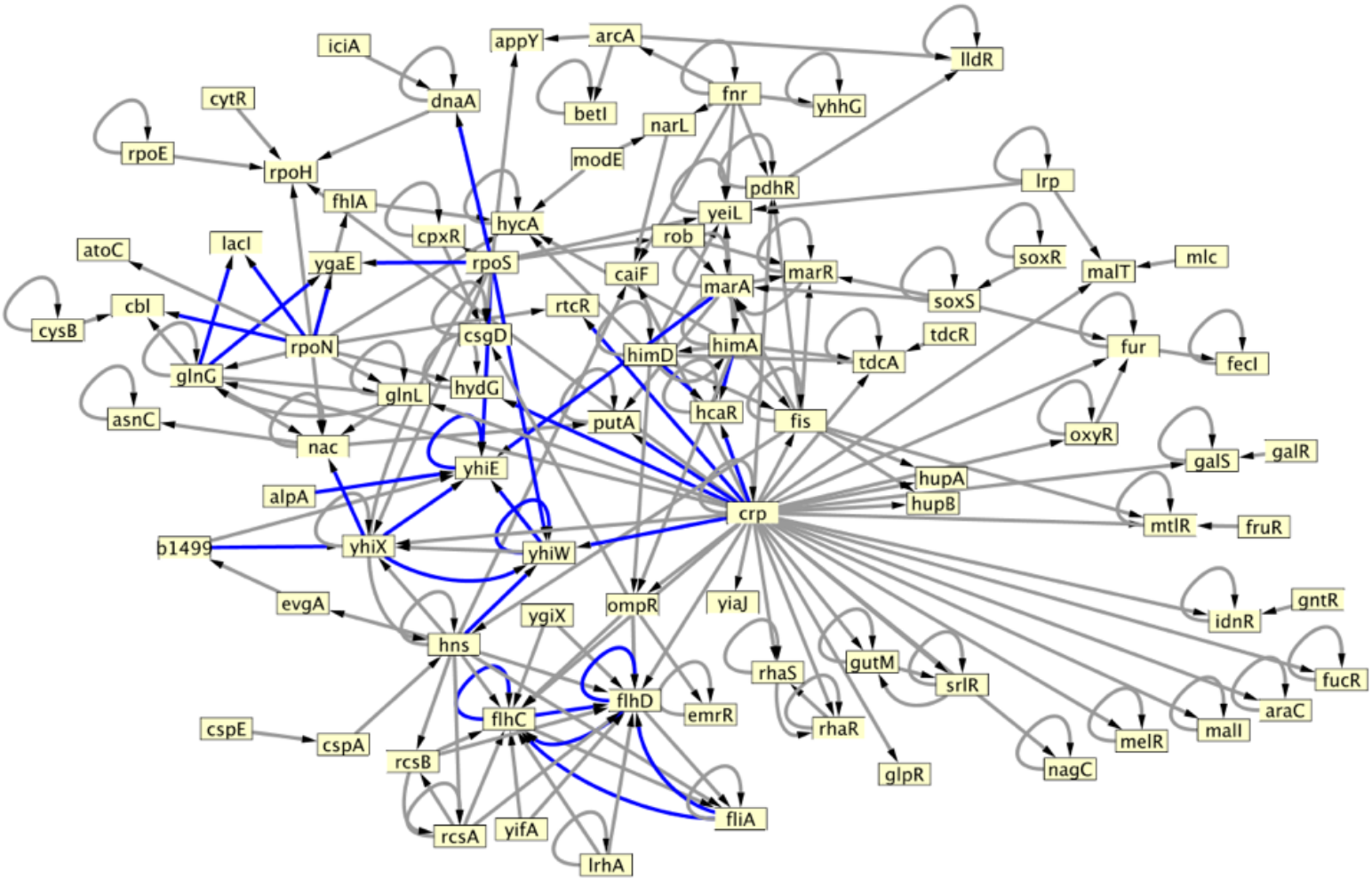




Network 2: metabolic network



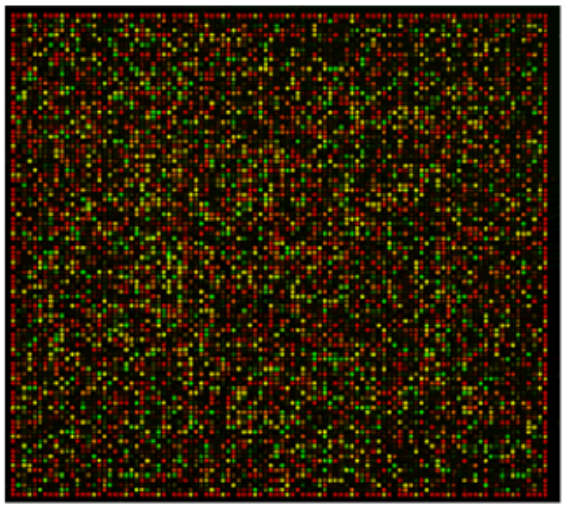
Network 3: gene transcriptional regulatory network



Data available

Biologists have collected a lot of data about proteins. e.g.,

- Gene expression measurements
- Phylogenetic profiles
- Location of proteins/enzymes in the cell



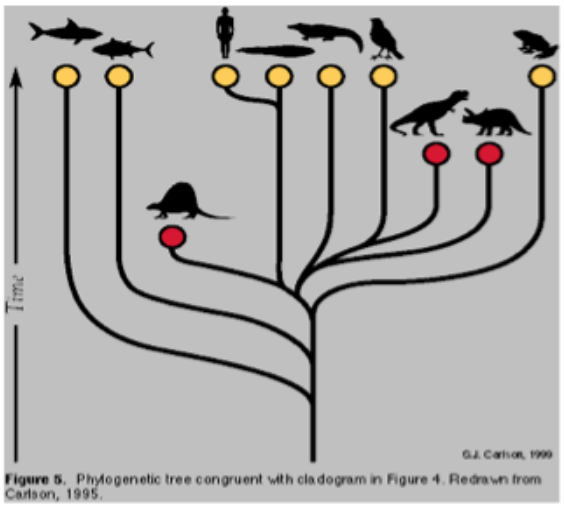
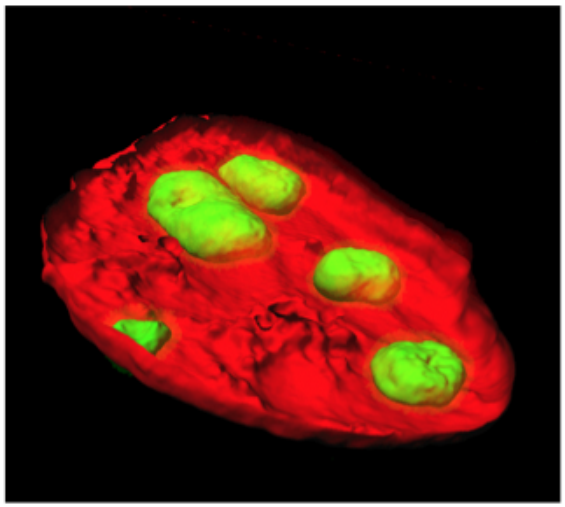
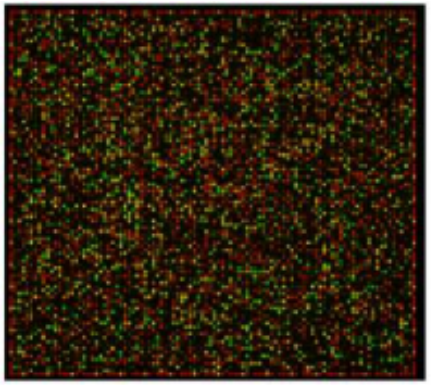
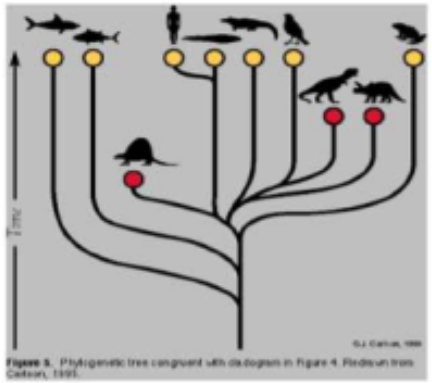


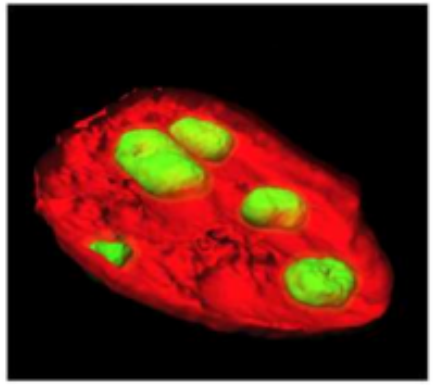
Figure 5. Phylogenetic tree congruent with cladogram in Figure 4. Redrawn from Carlson, 1995.



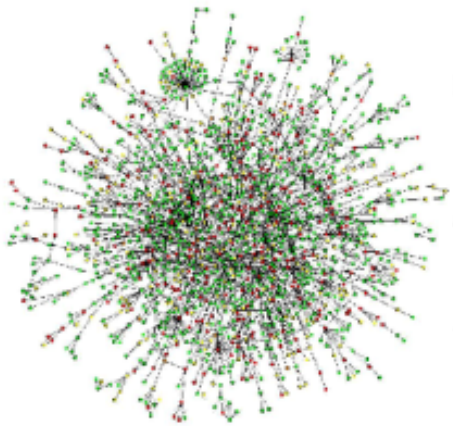
Problem 1 : how to infer relationships between genes from biological data?

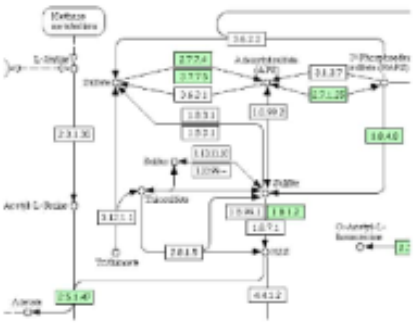


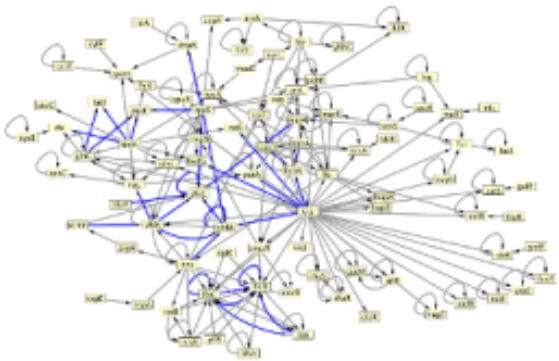




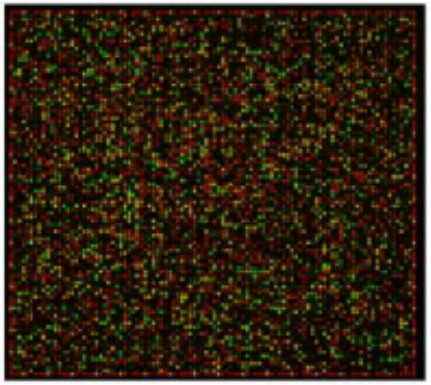
Inference

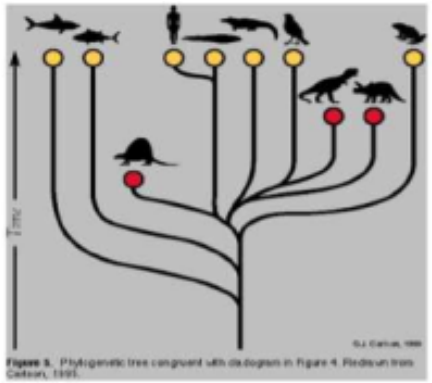


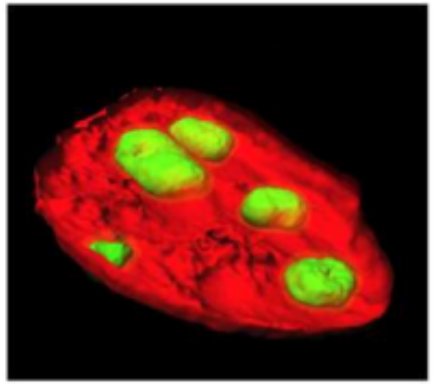




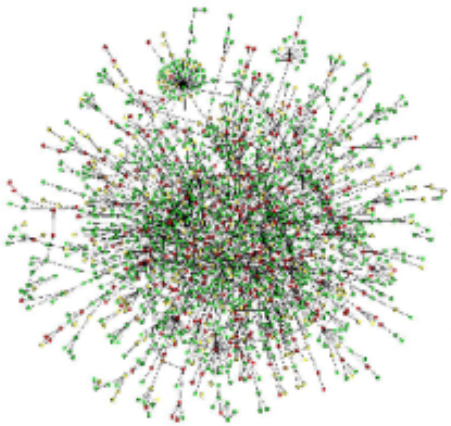
Problem 2 : how to use biological networks to help in the analysis of genomic data?

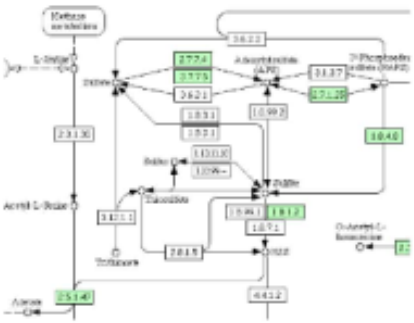


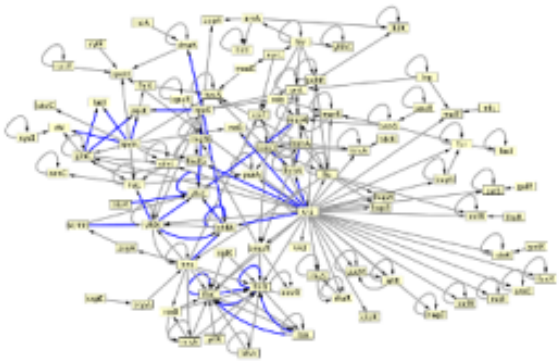




Interpretation







Outline

1. How to infer relationships between genes from biological data?

2. How to use biological networks to help in the analysis of genomic data?

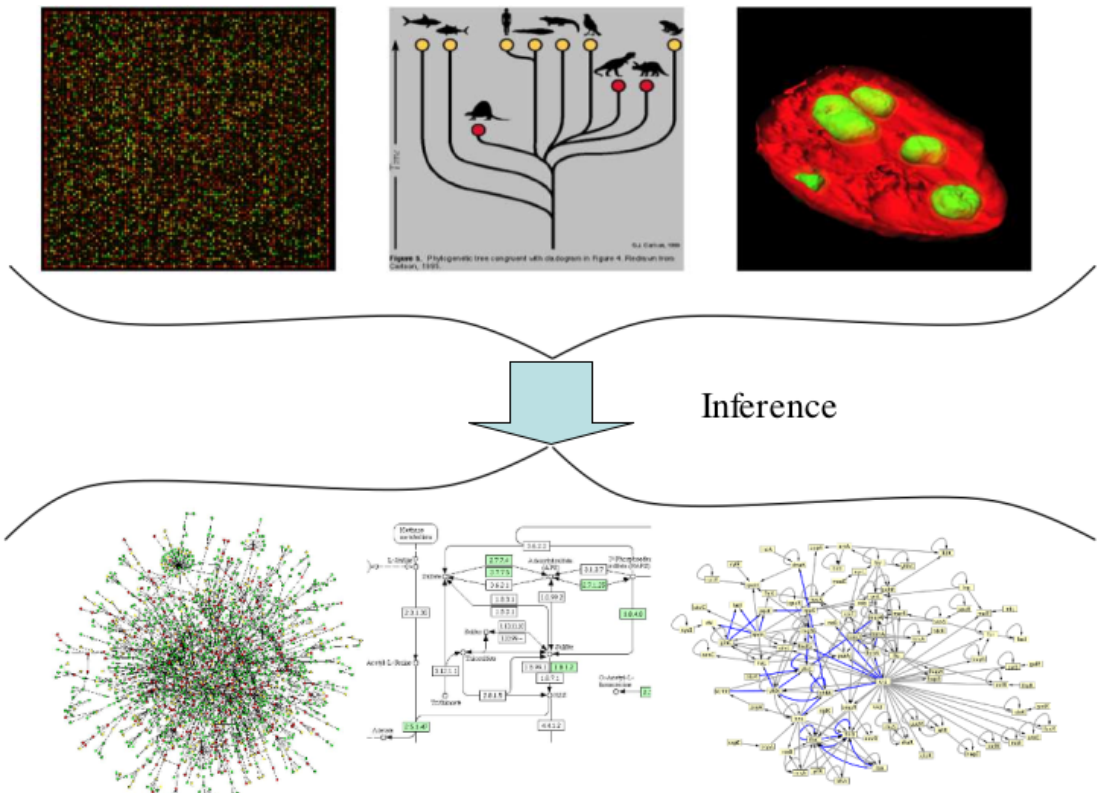
Outline

1. How to infer relationships between genes from biological data?

2. How to use biological networks to help in the analysis of genomic data?

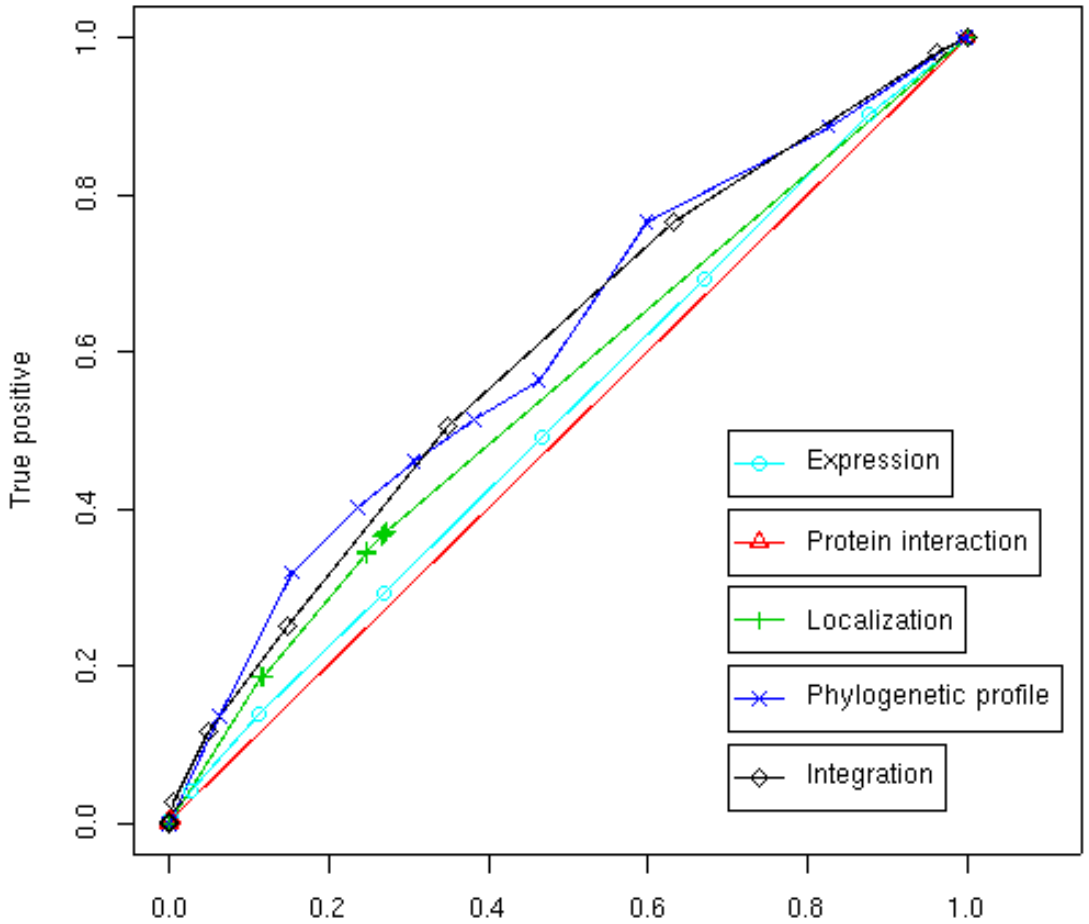
Typical reverse engineering strategies

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



Does it work? Case of metabolic network

- The known metabolic network of the yeast involves 769 proteins.
- Predict edges from distances between a variety of genomic data (expression, localization, phylogenetic profiles, interactions).

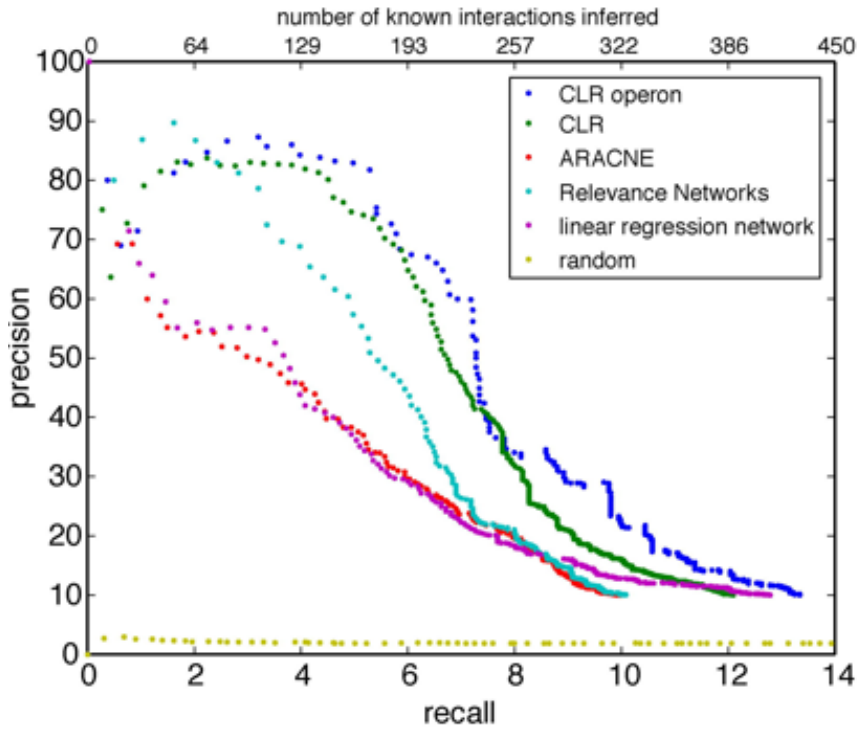


Does it work? Case of regulatory network

OPEN ACCESS Freely available online PLOS BIOLOGY

Large-Scale Mapping and Validation of *Escherichia coli* Transcriptional Regulation from a Compendium of Expression Profiles

Jeremiah J. Faith[1☯], Boris Hayete[1☯], Joshua T. Thaden[2,3], Ilaria Mogno[2,4], Jamey Wierzbowski[2,5], Guillaume Cottarel[2,5], Simon Kasif[1,2], James J. Collins[1,2], Timothy S. Gardner[1,2*]

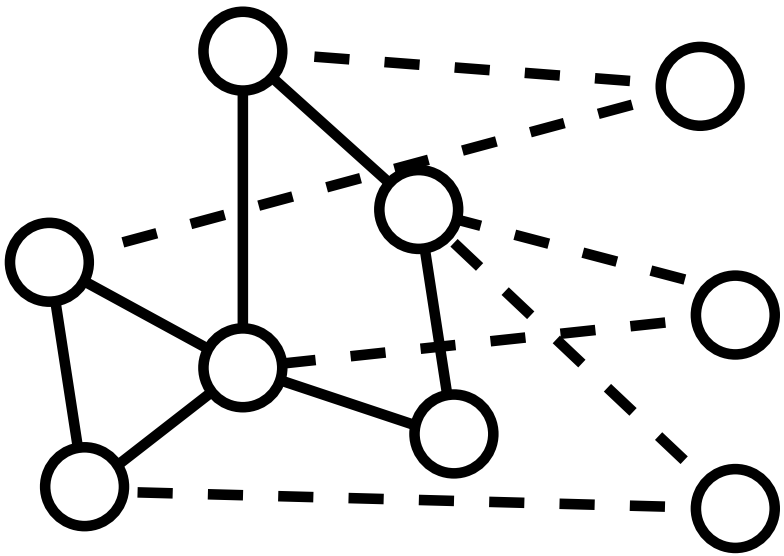


Change of paradigm

Motivation

In actual applications,

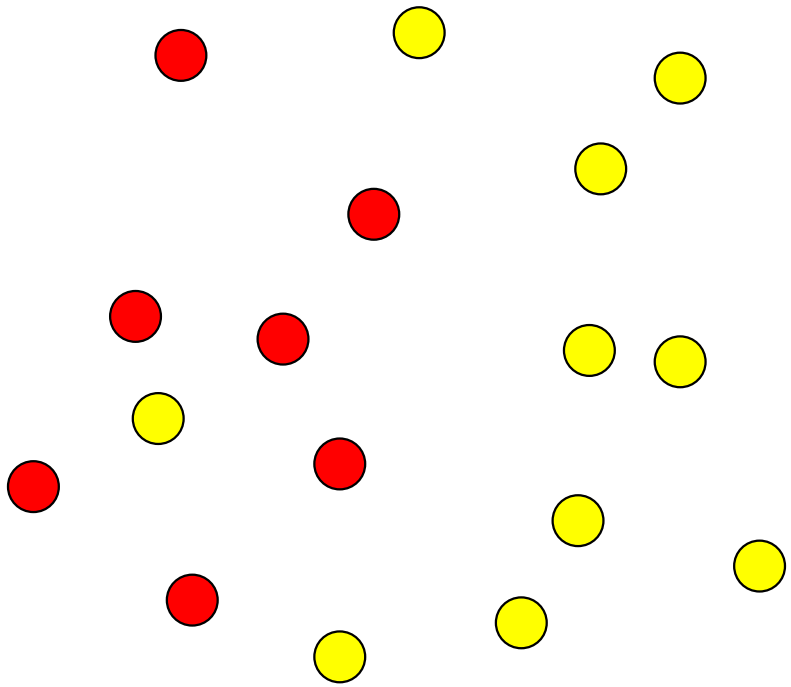
- we know in advance parts of the network to be inferred
- the problem is to add/remove nodes and edges using genomic data as side information



Supervised method

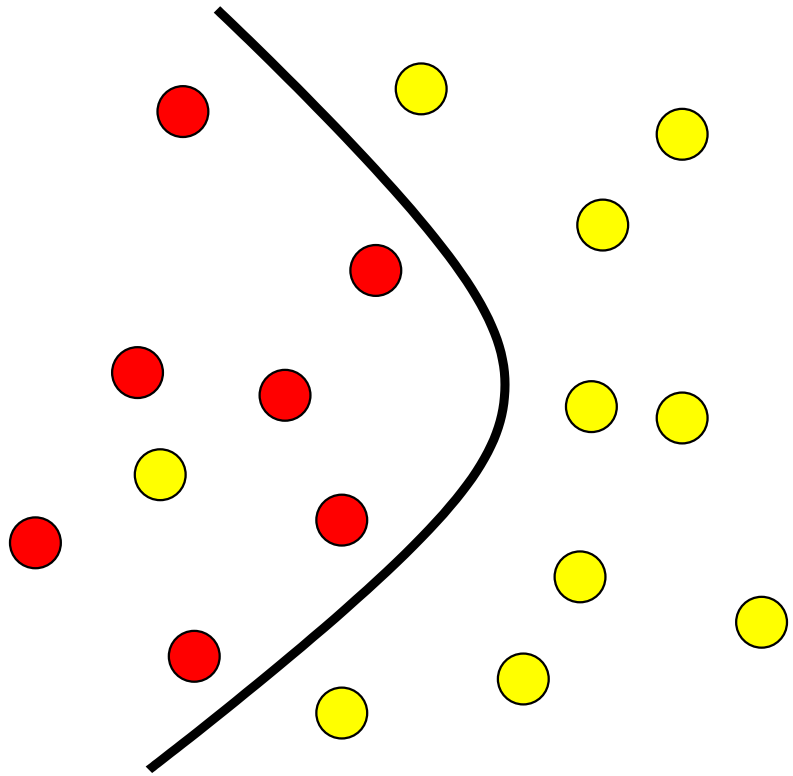
- Given genomic data and the currently known network...
- Infer missing edges between current nodes and additional nodes.

Interlude : Pattern recognition



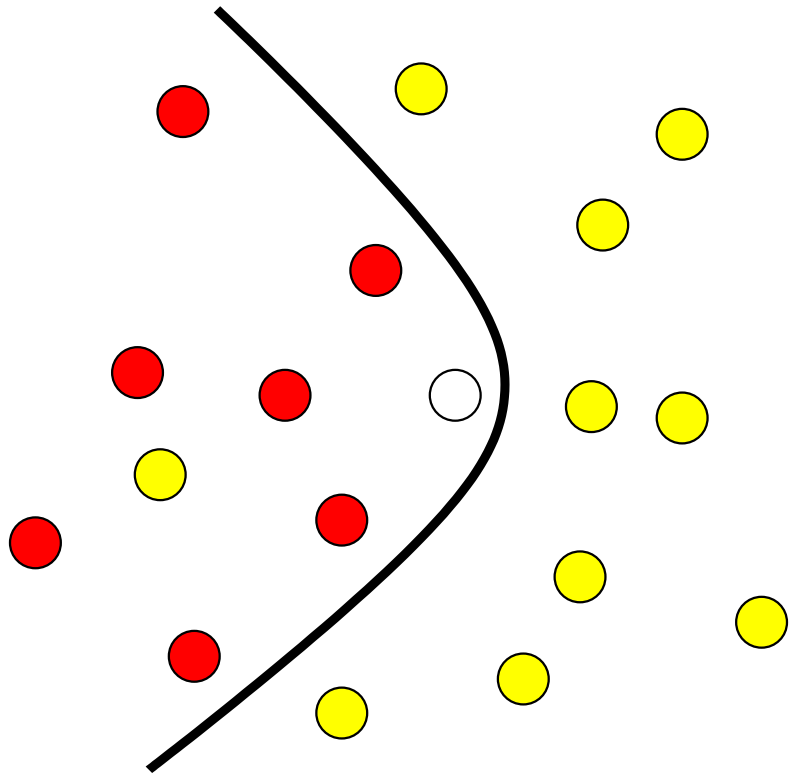
- Given a training set of patterns in two classes, learn to discriminate them
- Many algorithms (ANN, SVM, Decision tress, ...)

Interlude : Pattern recognition



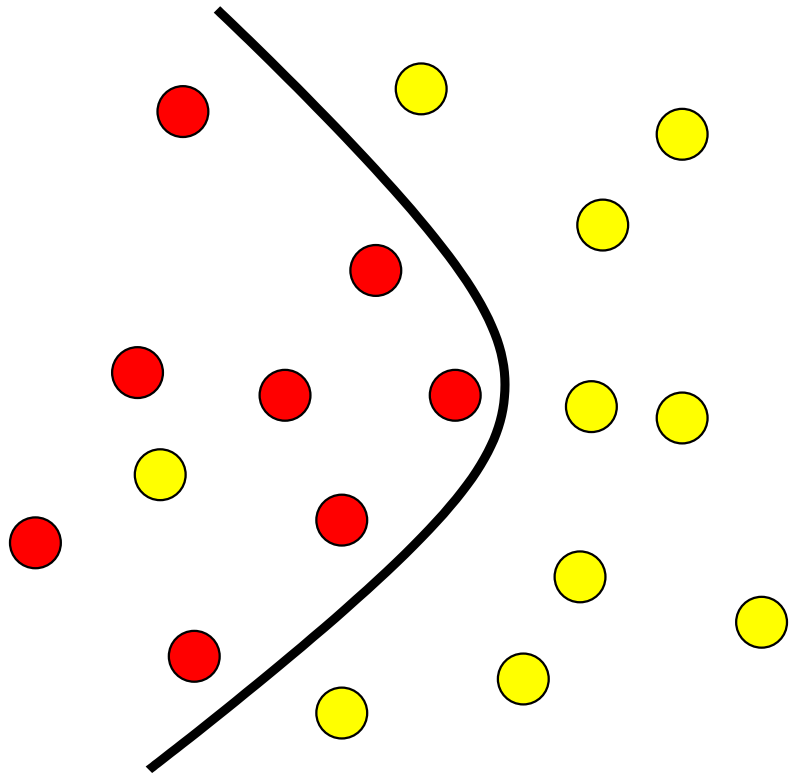
- Given a training set of patterns in two classes, learn to discriminate them
- Many algorithms (ANN, SVM, Decision tress, ...)

Interlude : Pattern recognition



- Given a training set of patterns in two classes, learn to discriminate them
- Many algorithms (ANN, SVM, Decision tress, ...)

Interlude : Pattern recognition



- Given a training set of patterns in two classes, learn to discriminate them
- Many algorithms (ANN, SVM, Decision tress, ...)

Pattern recognition and graph inference

Pattern recognition

Associate a binary label Y to each data X

Graph inference

Associate a binary label Y to each pair of data (X_1, X_2)

Two solutions

- Consider each pair (X_1, X_2) as a single data -> learning over pairs
- Reformulate the graph inference problem as a pattern recognition problem at the level of individual vertices -> local models

Pattern recognition and graph inference

Pattern recognition

Associate a binary label Y to each data X

Graph inference

Associate a binary label Y to each pair of data (X_1, X_2)

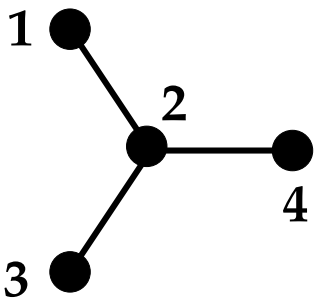
Two solutions

- Consider each pair (X_1, X_2) as a single data -> learning over pairs
- Reformulate the graph inference problem as a pattern recognition problem at the level of individual vertices -> local models

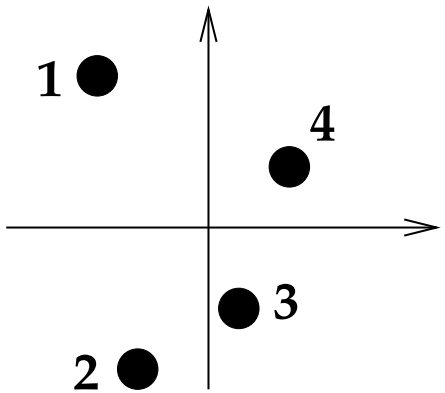
Pattern recognition for pairs

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!



Known graph

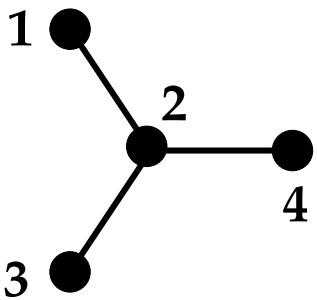


Genomic data

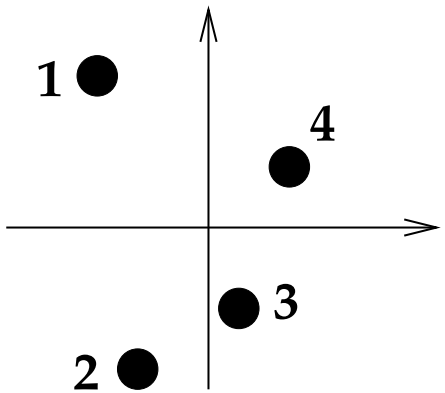
Pattern recognition for pairs

Formulation and basic issue

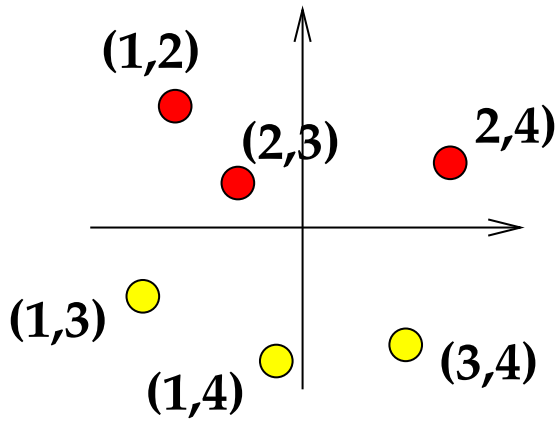
- A pair can be connected (1) or not connected (-1)
- From the known subgraph we can extract examples of connected and non-connected pairs
- However the genomic data characterize individual proteins; we need to work with pairs of proteins instead!



Known graph



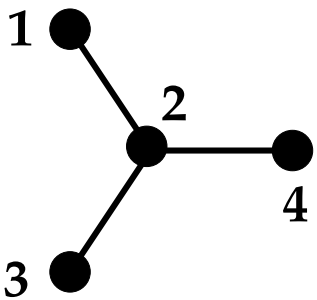
Genomic data



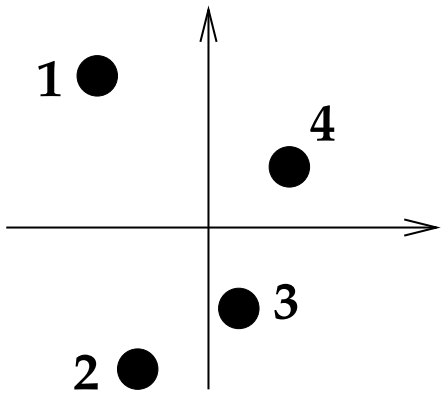
Pattern recognition for pairs

Formulation and basic issue

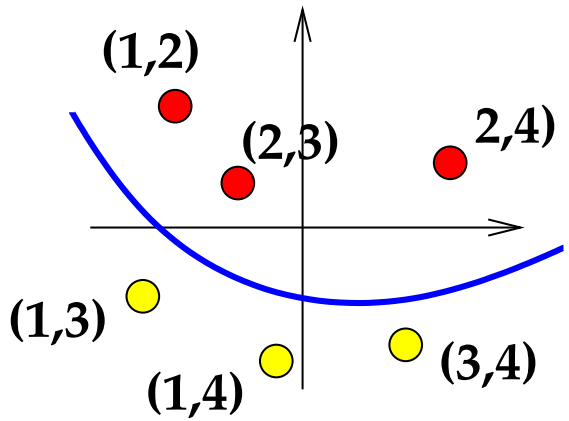
- A pair can be connected (1) or not connected (-1)
- From the known subgraph we can extract examples of connected and non-connected pairs
- However the genomic data characterize individual proteins; we need to work with pairs of proteins instead!



Known graph



Genomic data



Representing a pair

Concatenation?

- A simple idea is to concatenate the vectors u and v to obtain a $2p$-dimensional vector of (u, v):

$$\psi(u, v) = u \oplus v = \begin{pmatrix} u \\ v \end{pmatrix} .$$

- Problem: a linear function then becomes additive...

$$f(u, v) = w^\top \psi(u, v) = w_1^\top u + w^\top v .$$

Representing a pair

Concatenation?

- A simple idea is to concatenate the vectors u and v to obtain a $2p$-dimensional vector of (u, v):

$$\psi(u, v) = u \oplus v = \begin{pmatrix} u \\ v \end{pmatrix} .$$

- Problem: a linear function then becomes additive...

$$f(u, v) = w^\top \psi(u, v) = w_1^\top u + w^\top v .$$

Other representations for pairs

Symmetric tensor product (Ben-Hur and Noble, 2006)

$$\psi(u, v) = (u \otimes v) + (v \otimes u) \ .$$

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

Metric learning (V. et al, 2007)

$$\psi(u, v) = (u - v)^{\otimes 2} \ .$$

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 $D - C$.

Other representations for pairs

Symmetric tensor product (Ben-Hur and Noble, 2006)

$$\psi(u, v) = (u \otimes v) + (v \otimes u) \ .$$

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

Metric learning (V. et al, 2007)

$$\psi(u, v) = (u - v)^{\otimes 2} \ .$$

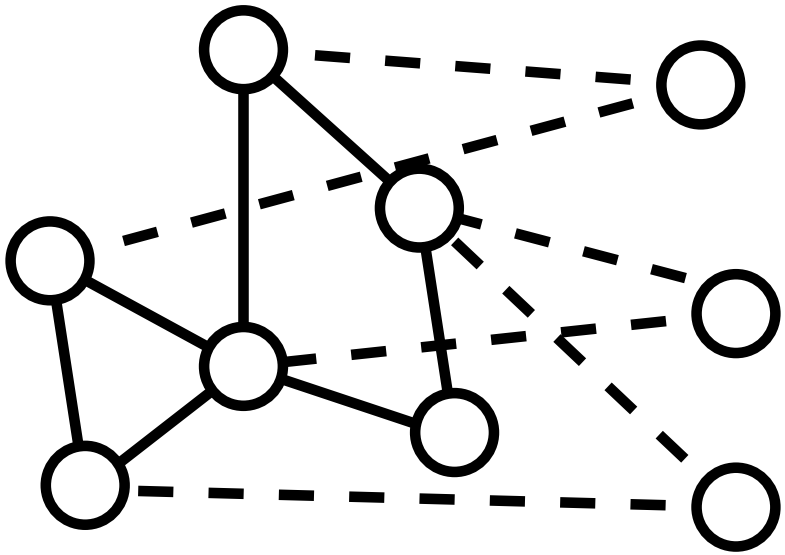
Intuition: a pair (A, B) is similar to a pair (C, D) if:

- $A - B$ is similar to $C - D$, or...
- $A - B$ is similar to $D - C$.

Supervised inference with local models

The idea (Bleakley et al., 2007)

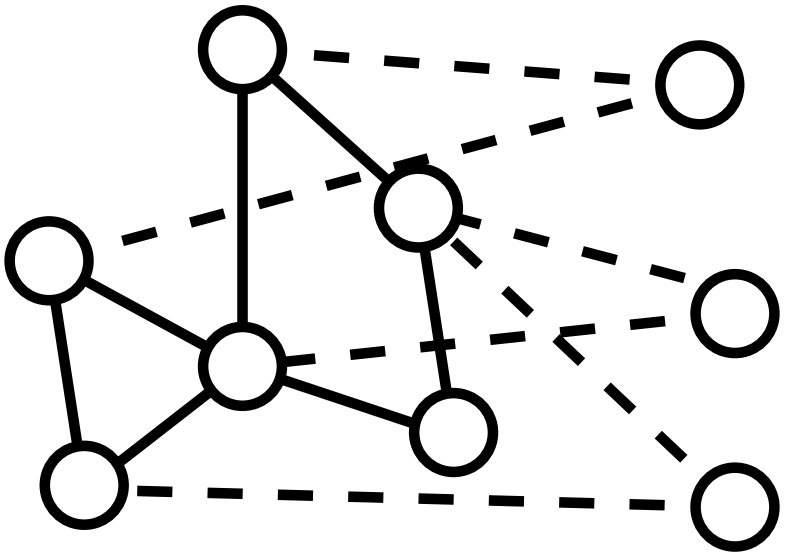
- Motivation: define specific models for each target node to discriminate between its neighbors and the others
- Treat each node independently from the other. Then combine predictions for ranking candidate edges.

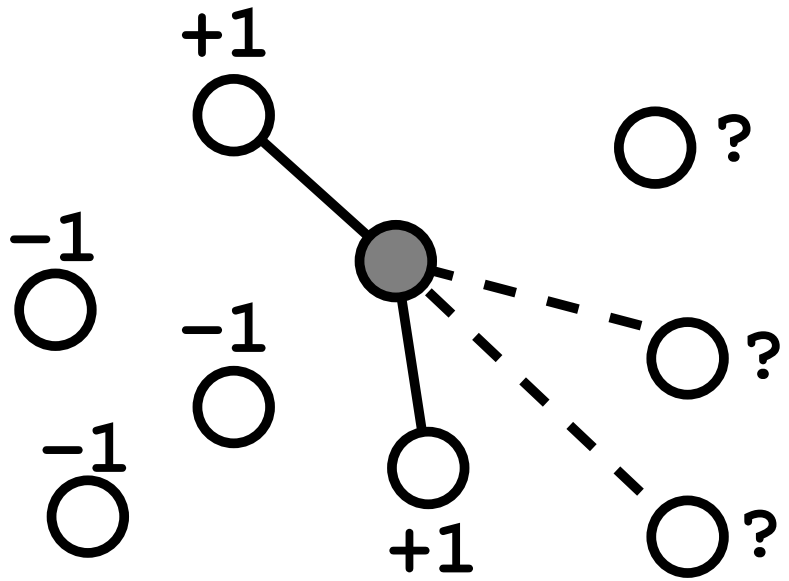


Supervised inference with local models

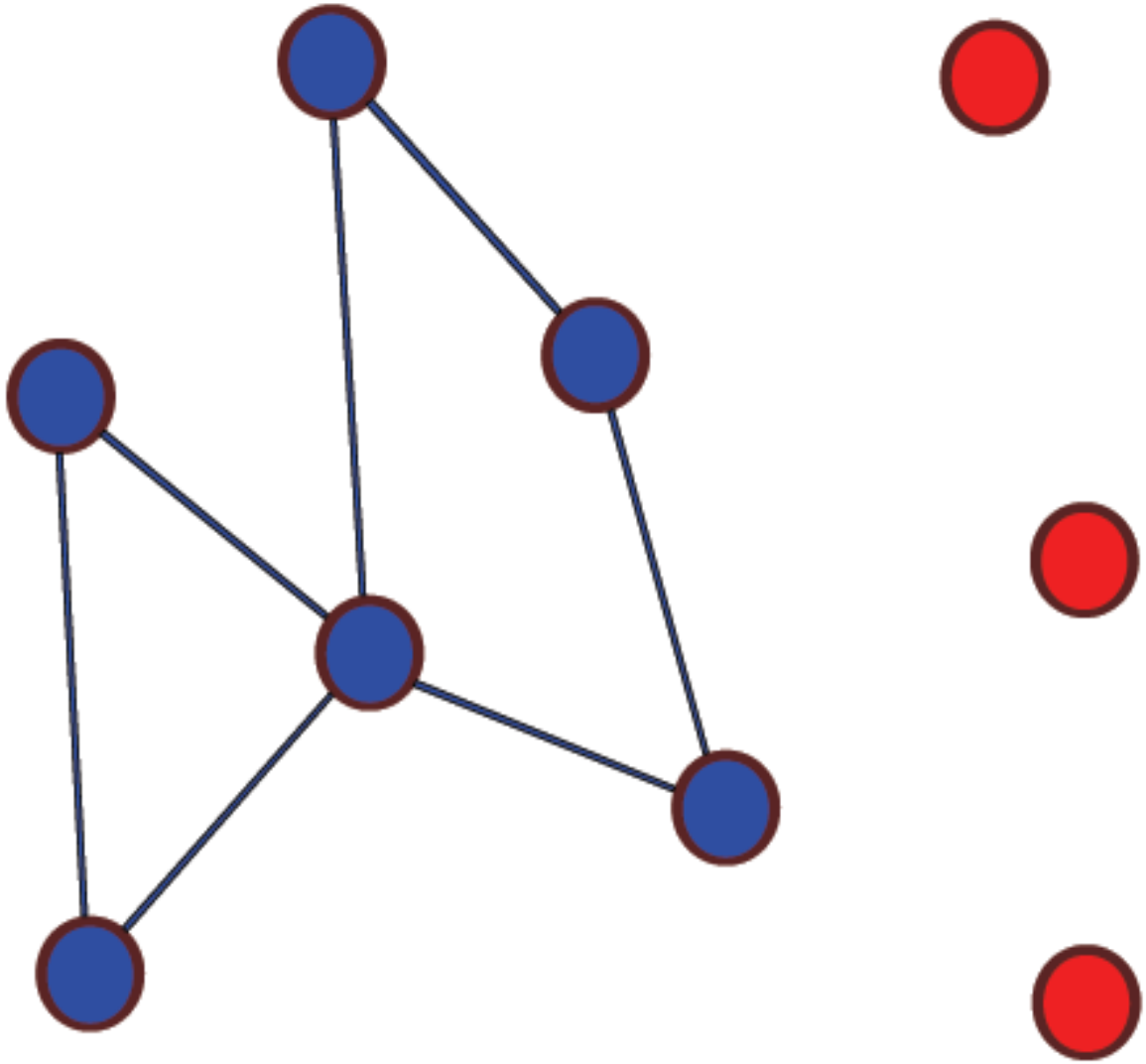
The idea (Bleakley et al., 2007)

- 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.

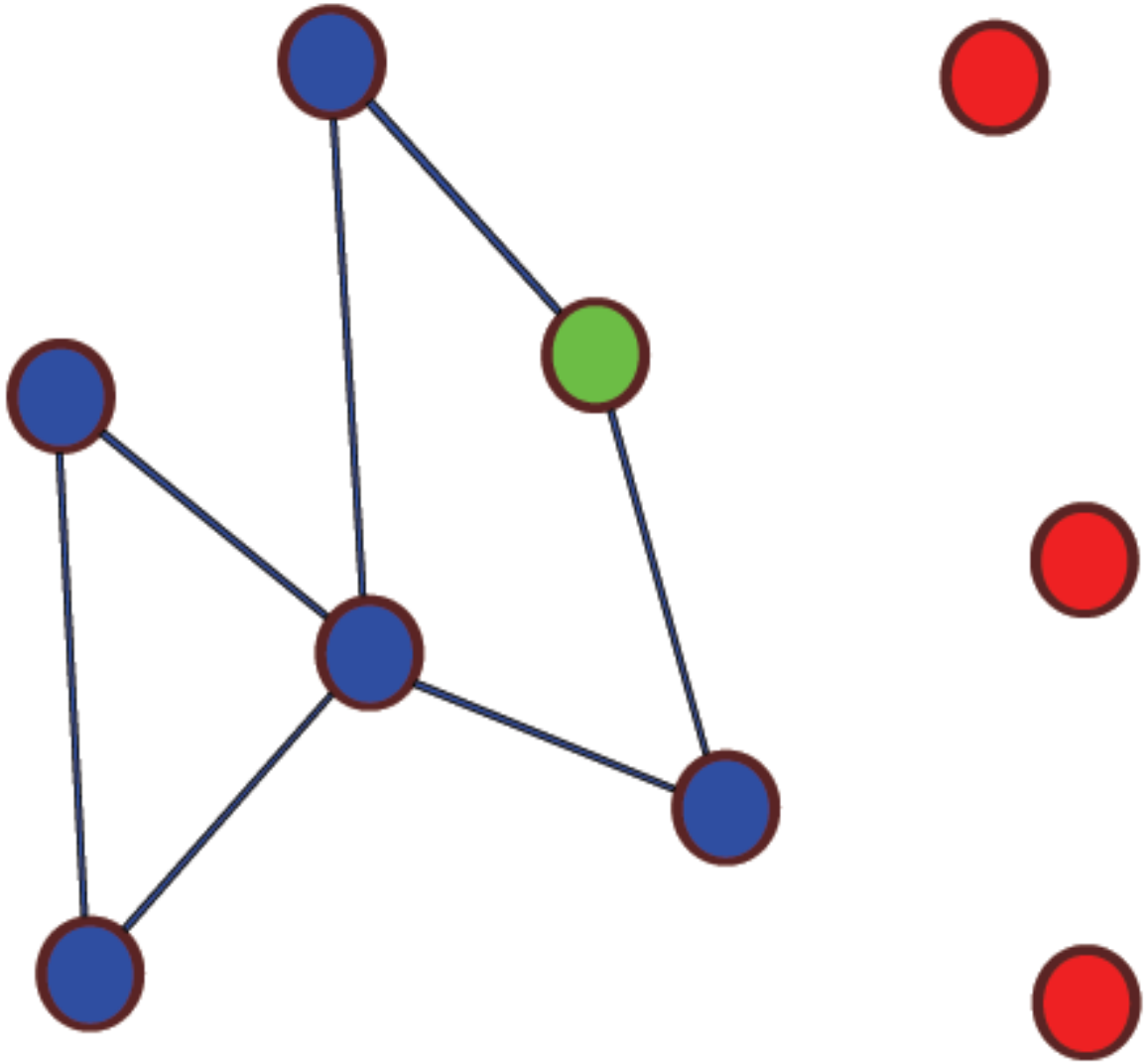




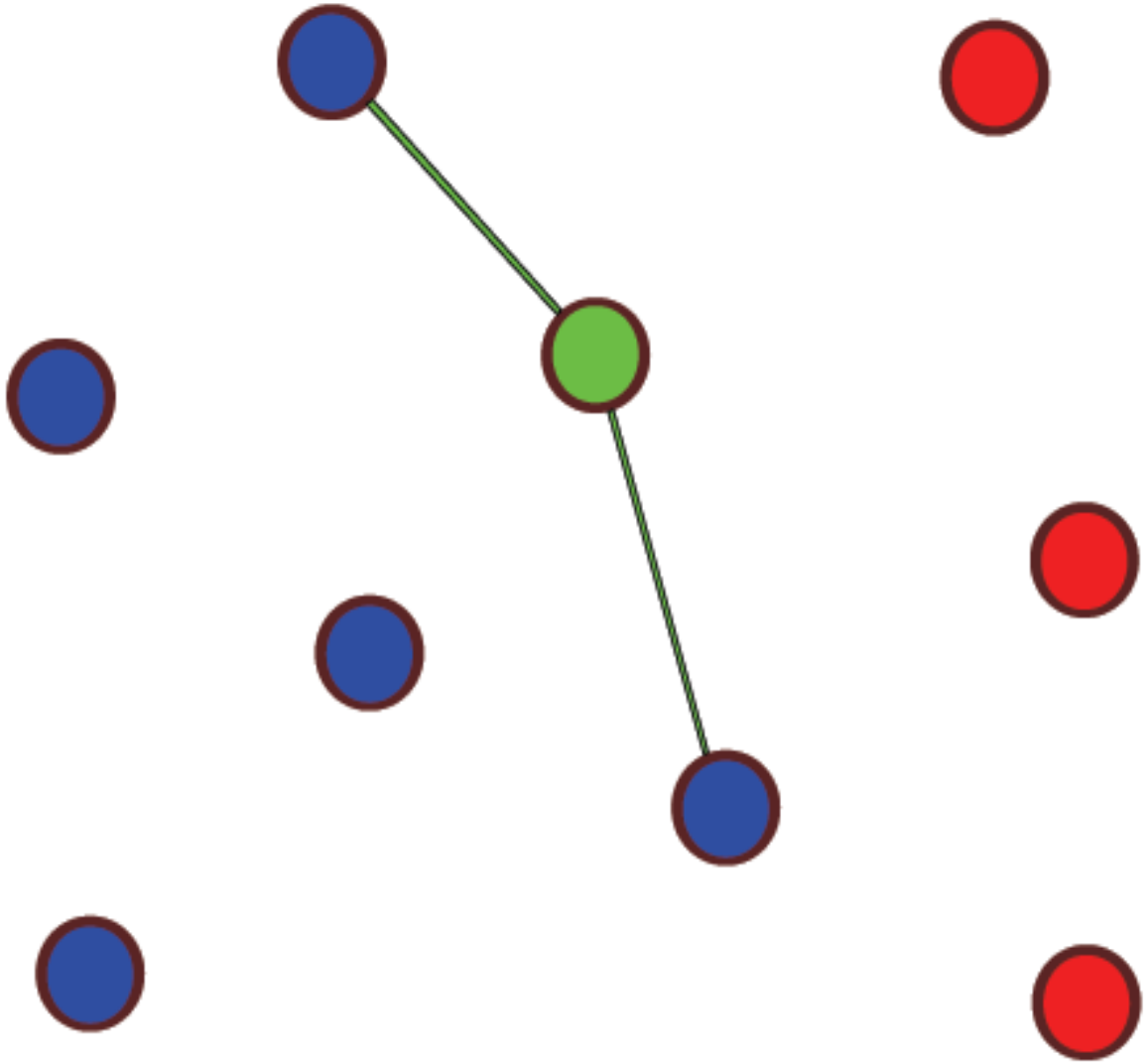
The LOCAL model



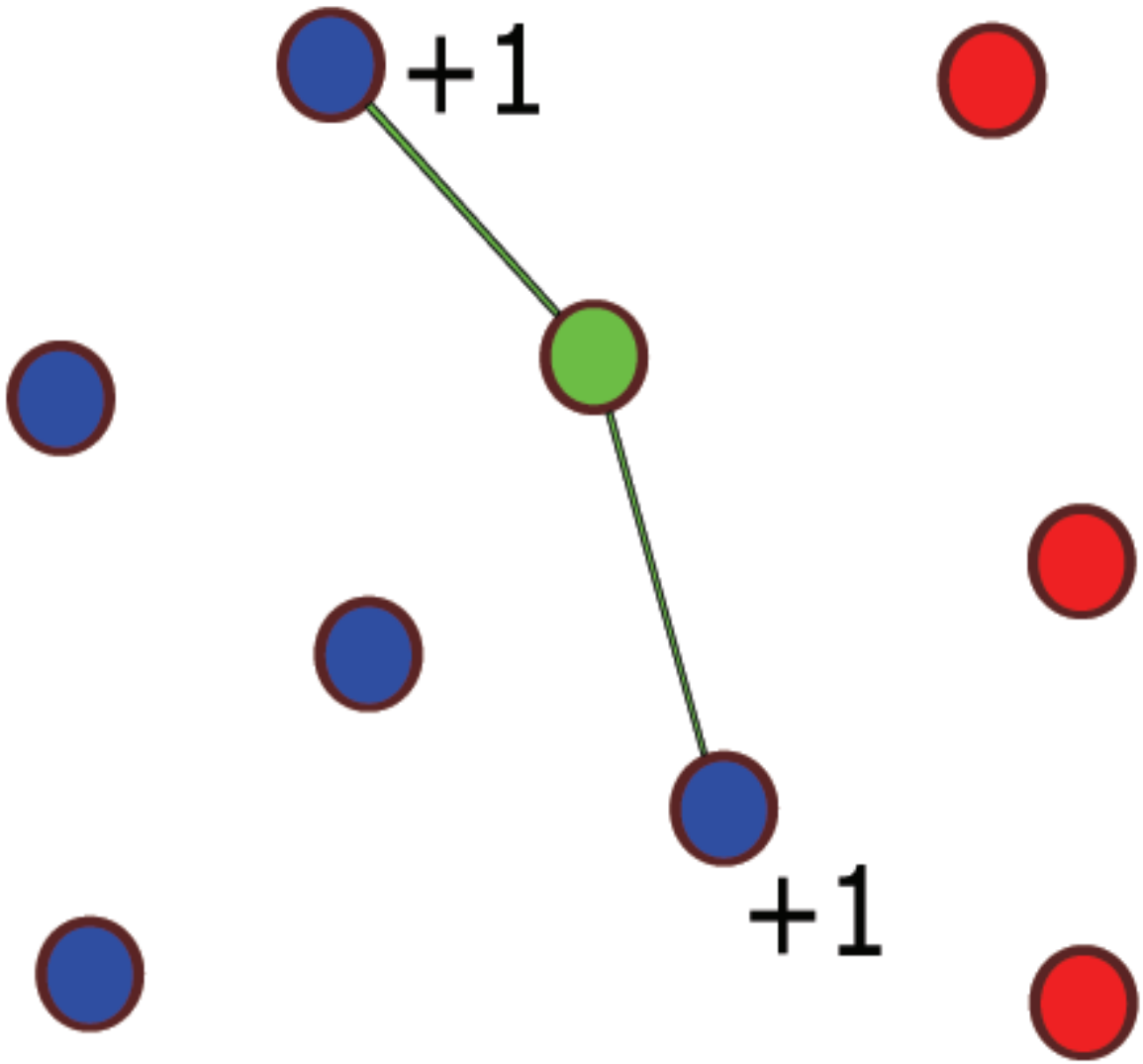
The LOCAL model



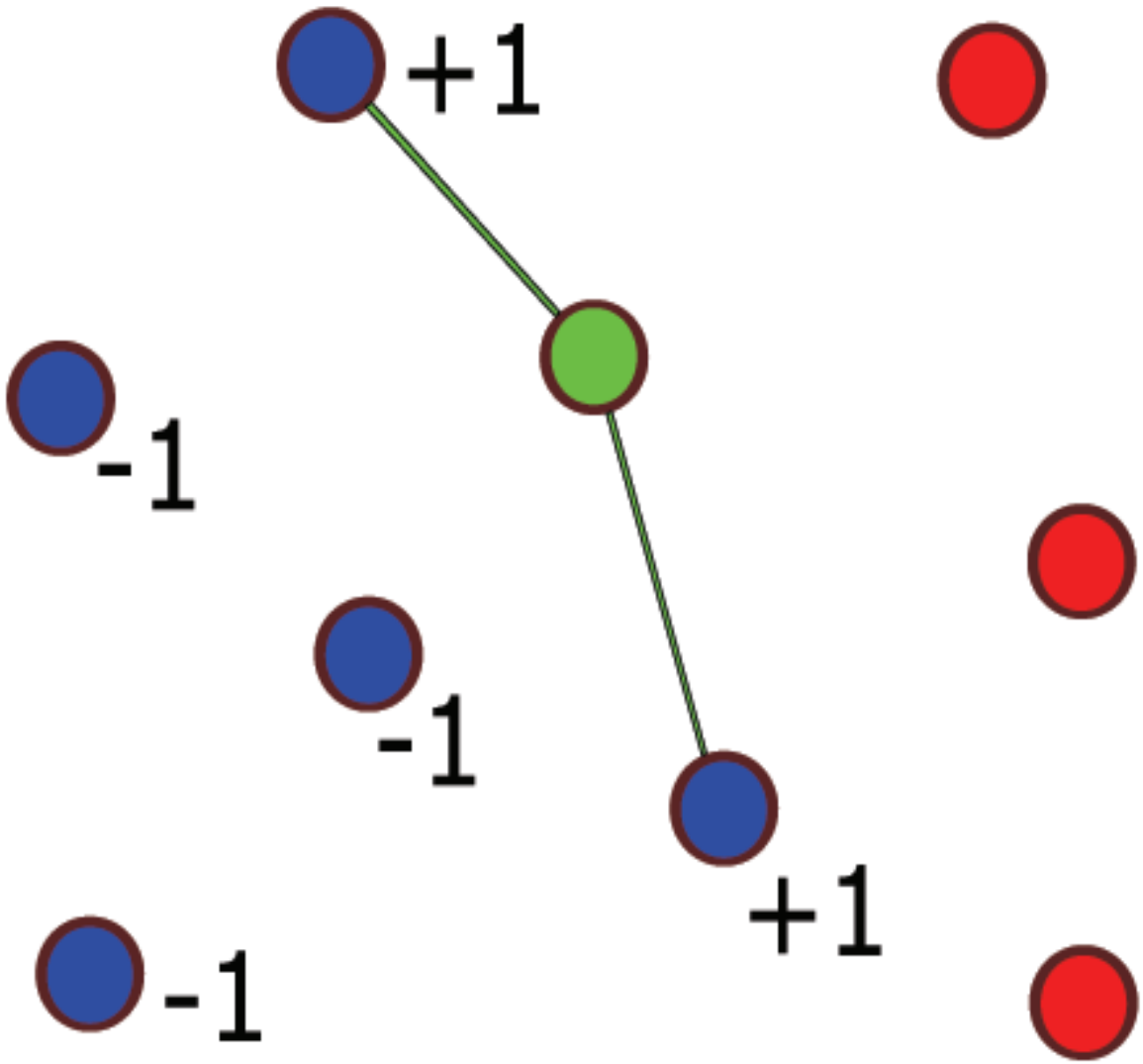
The LOCAL model



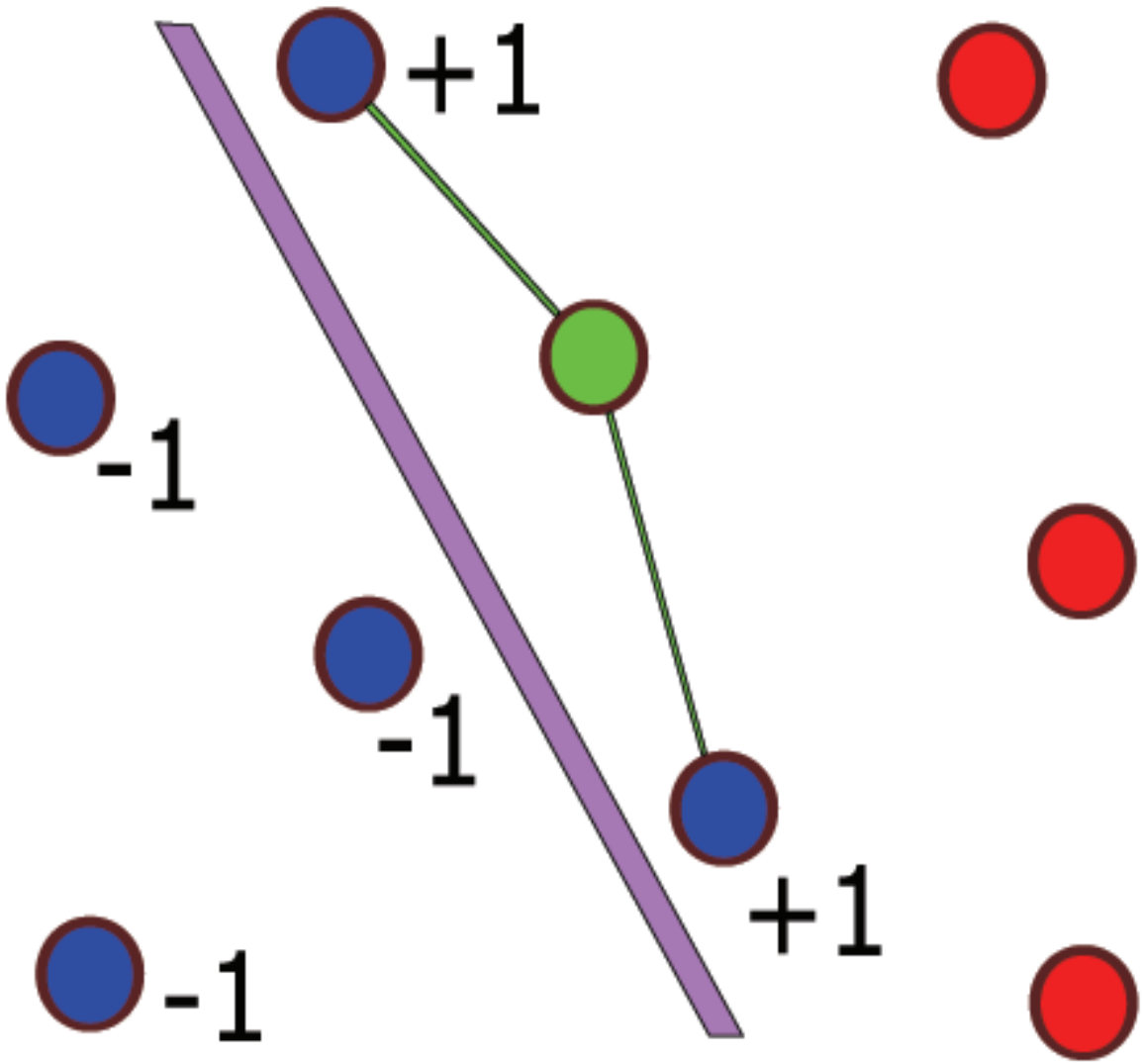
The LOCAL model



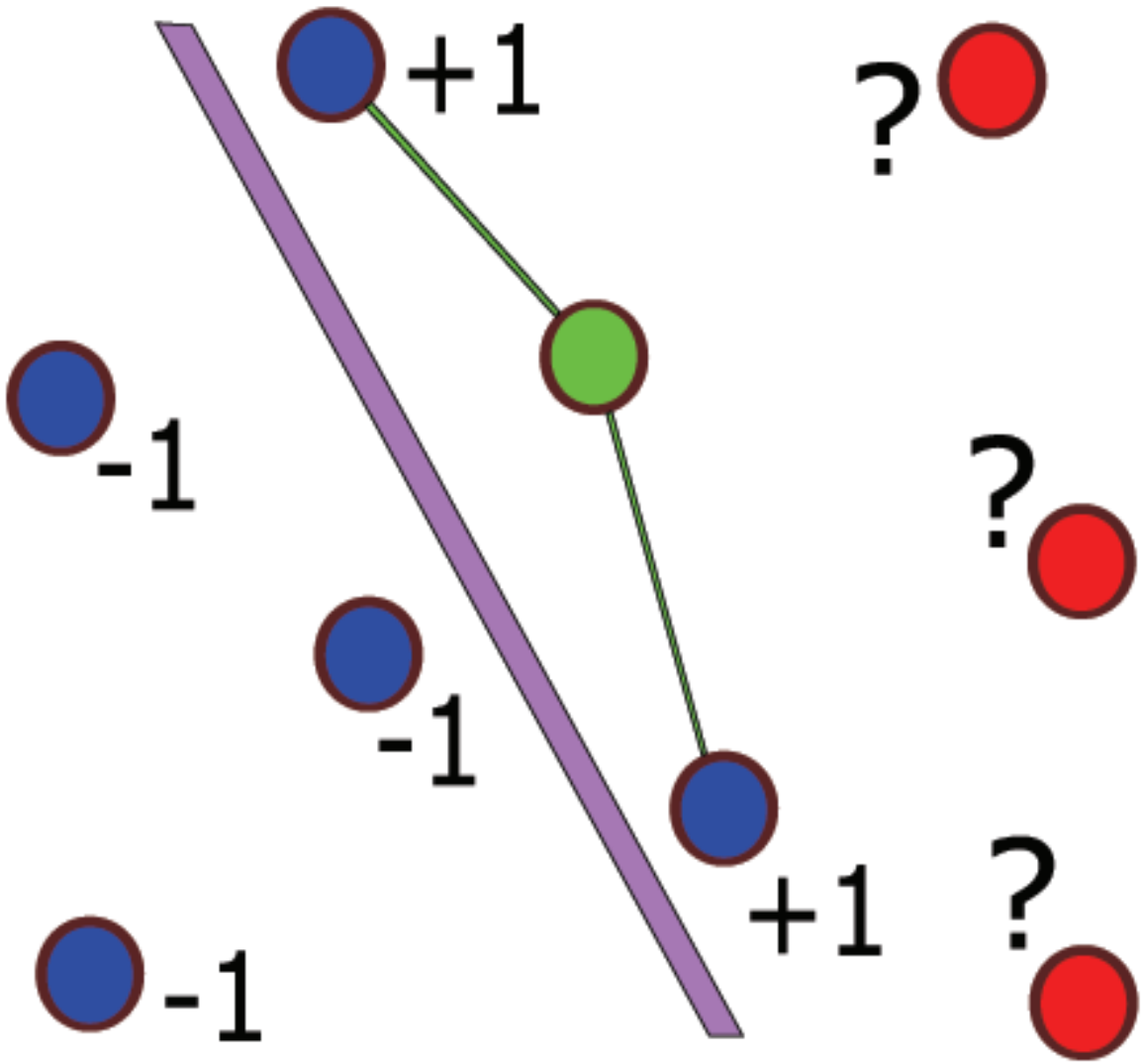
The LOCAL model



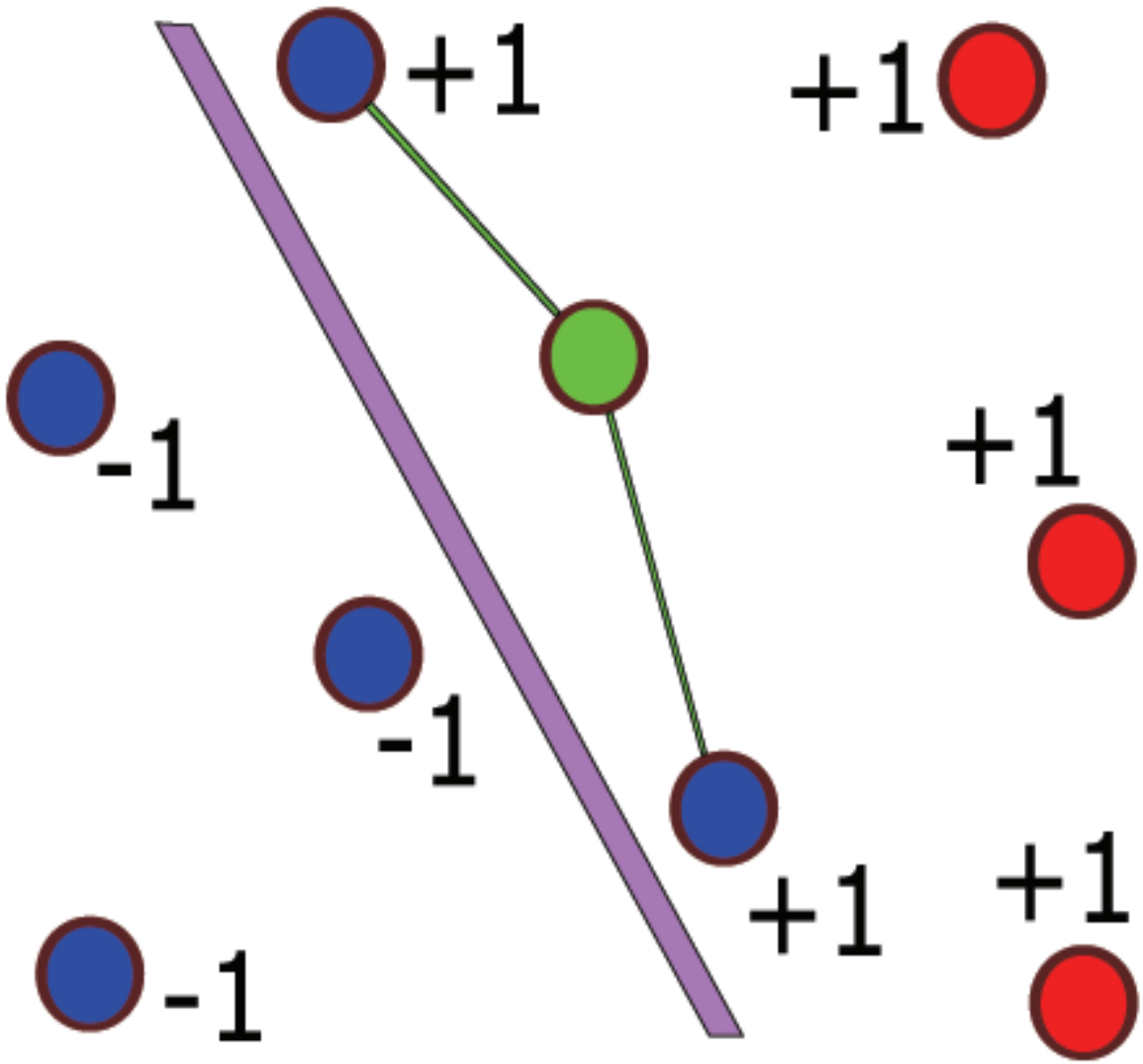
The LOCAL model



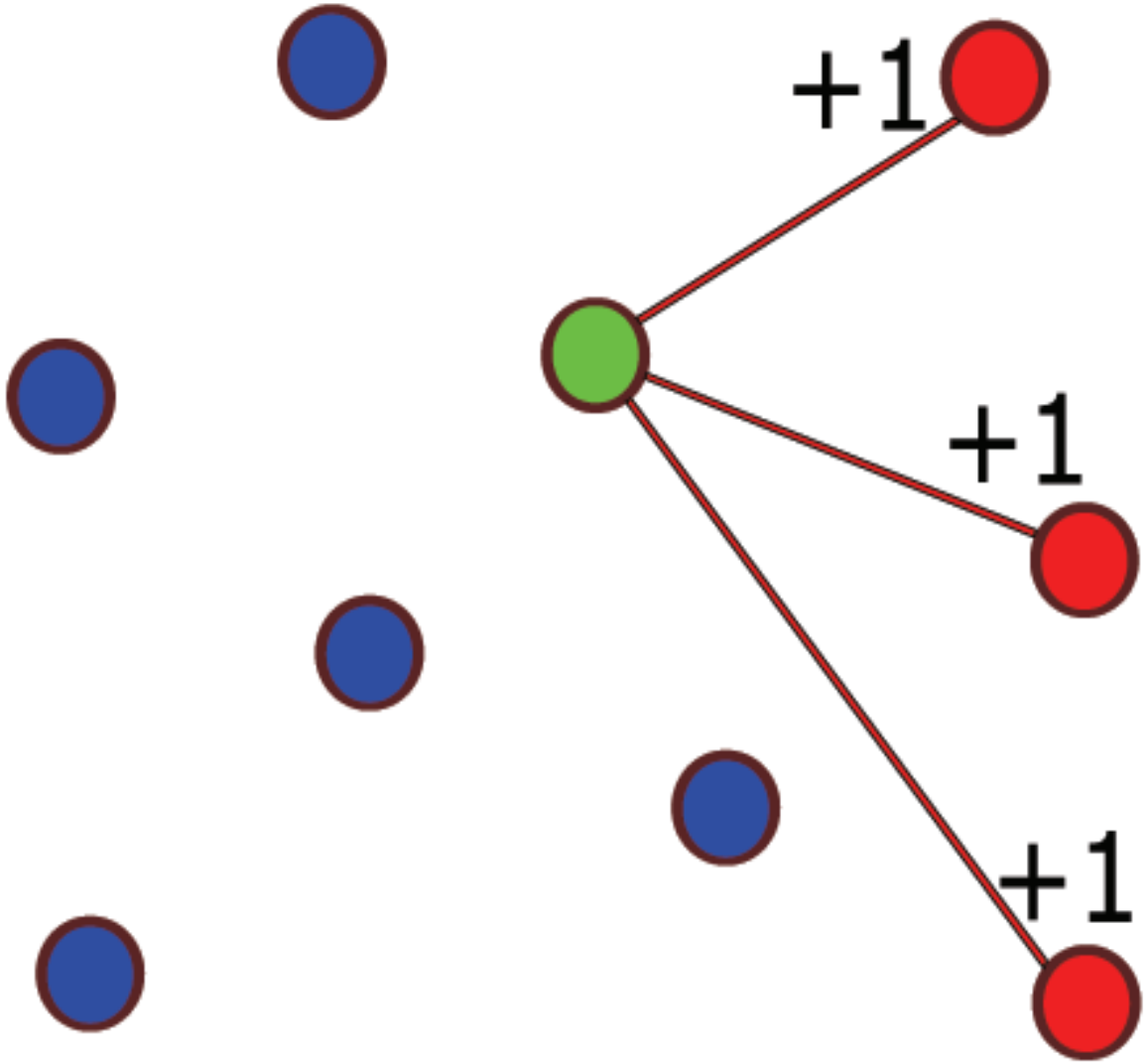
The LOCAL model



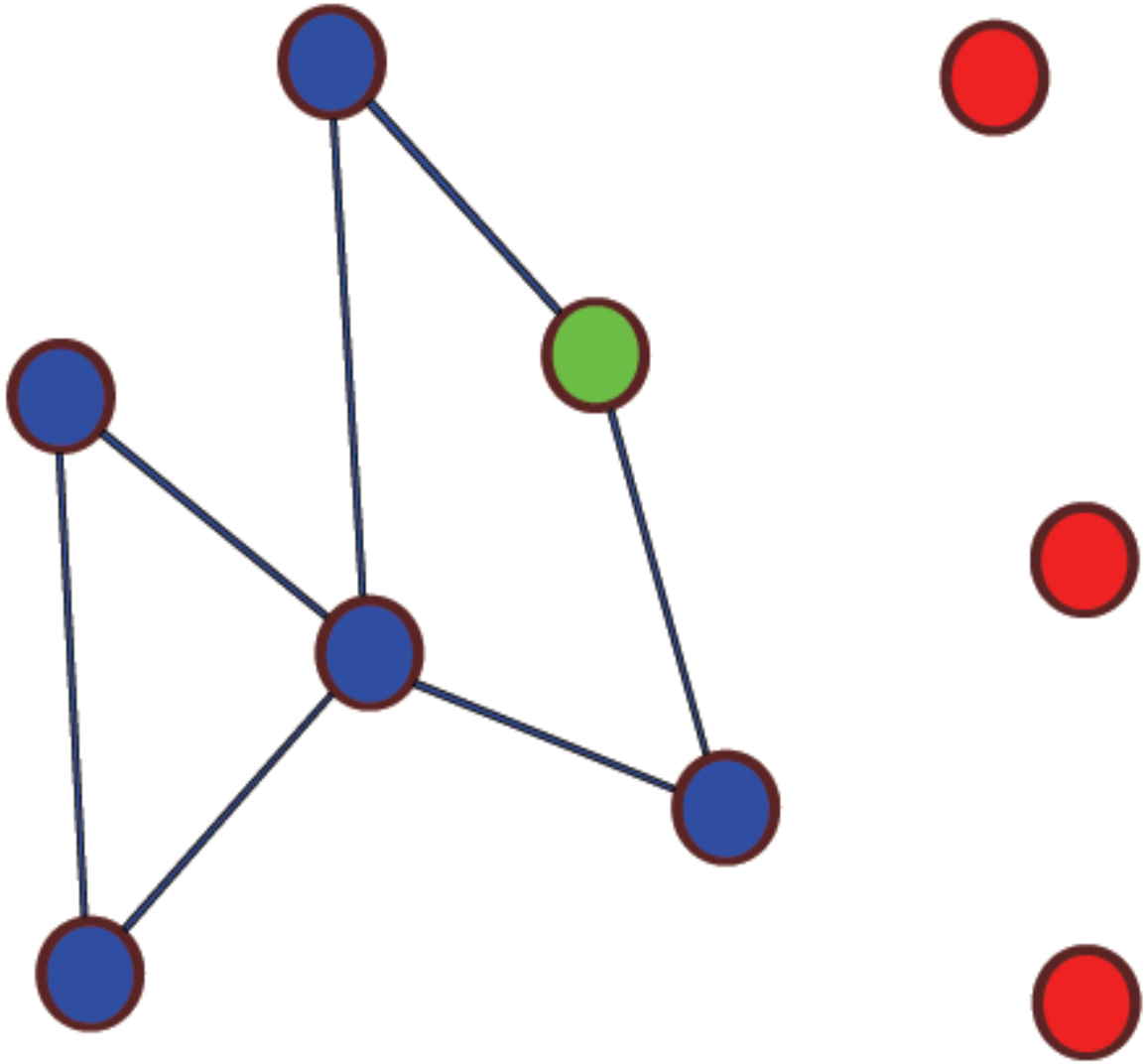
The LOCAL model



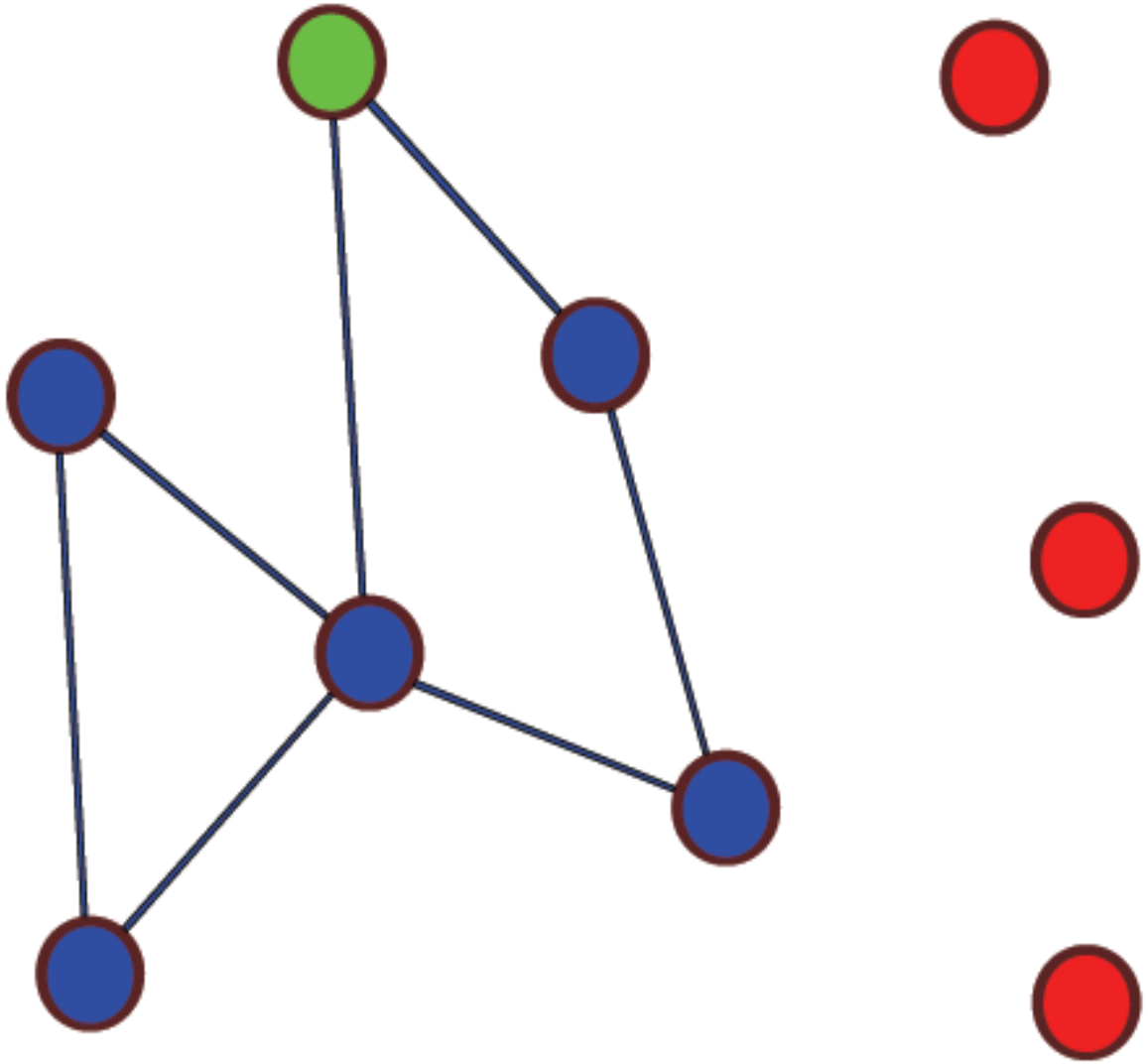
The LOCAL model



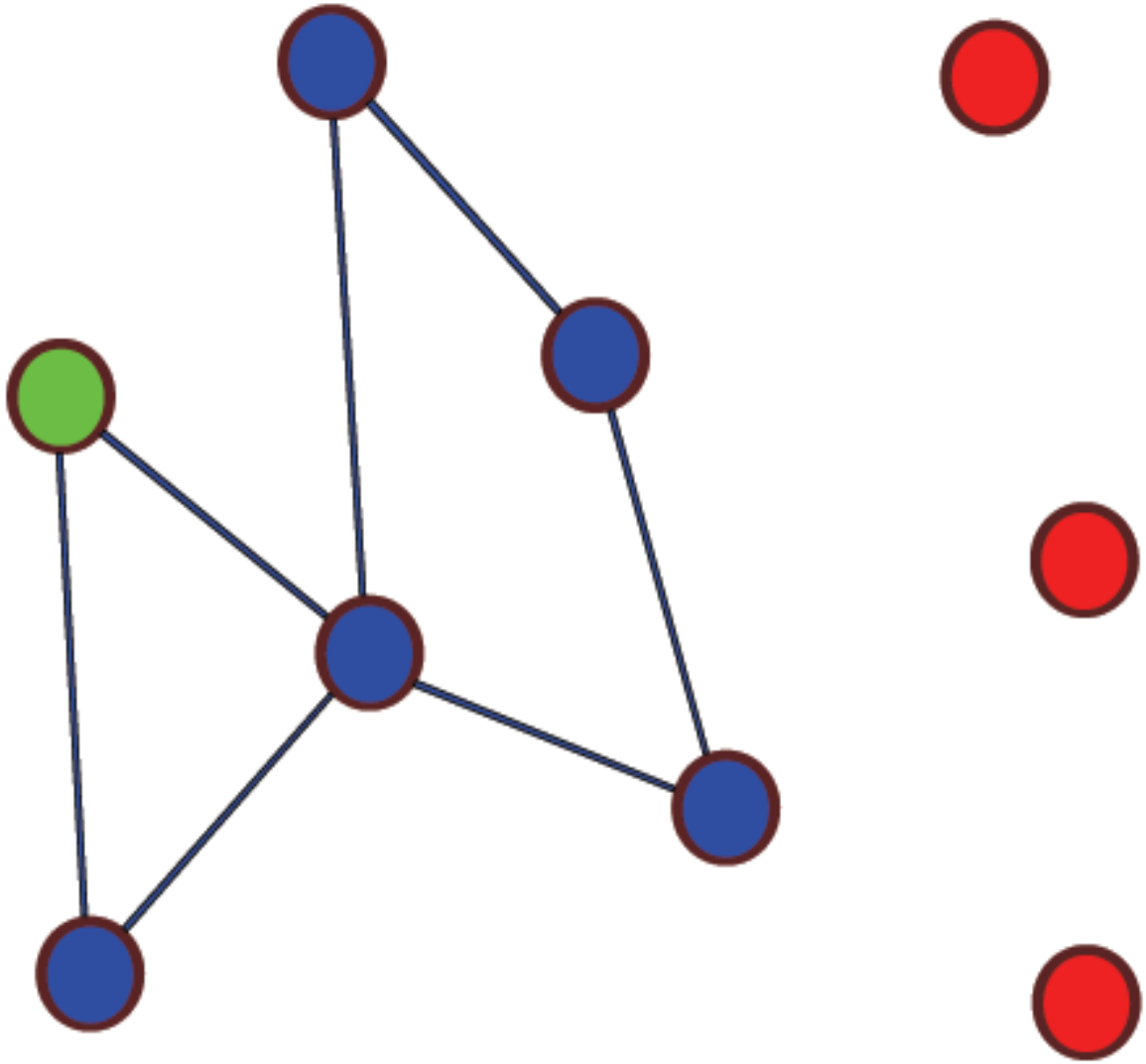
The LOCAL model



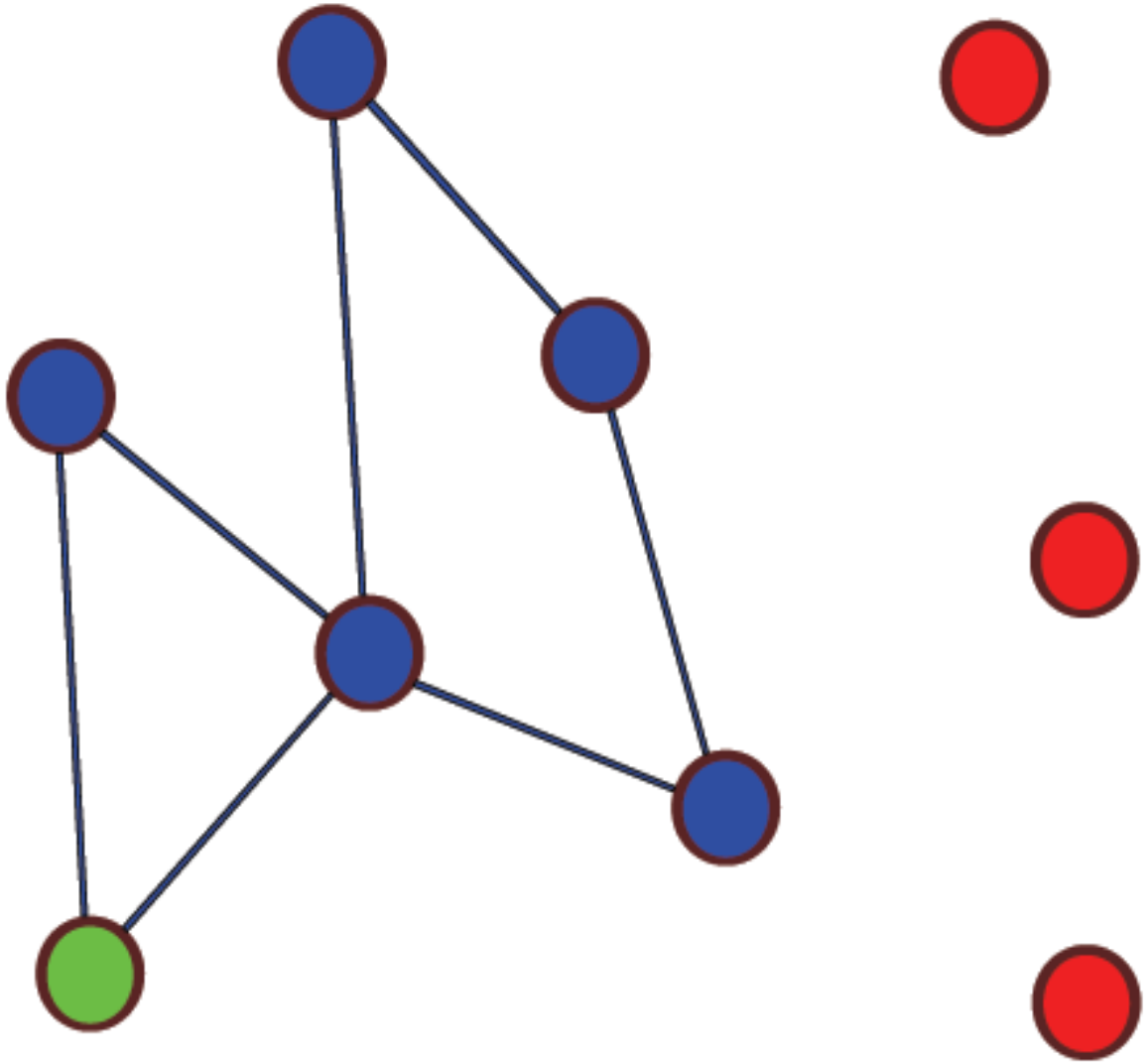
The LOCAL model



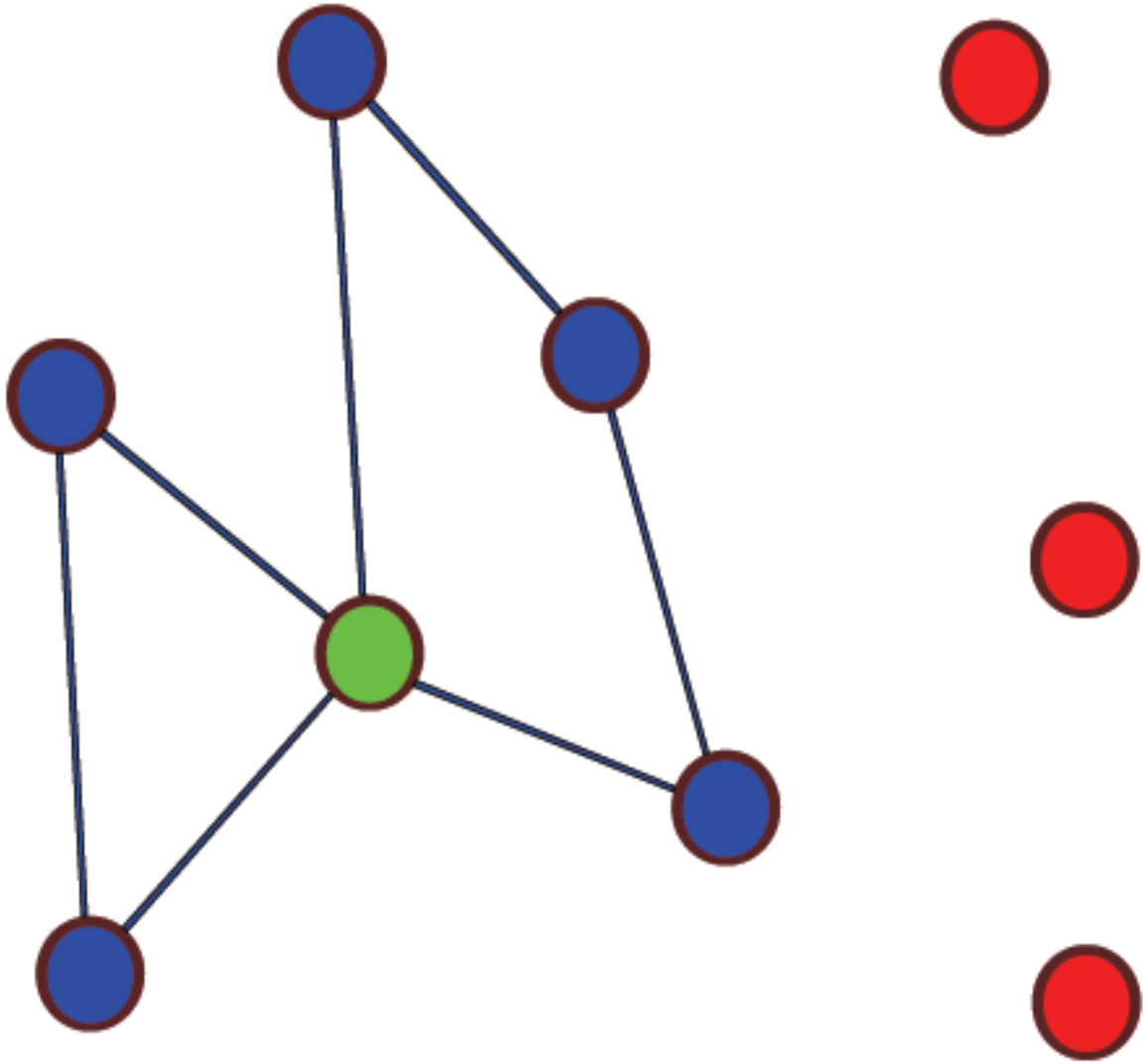
The LOCAL model



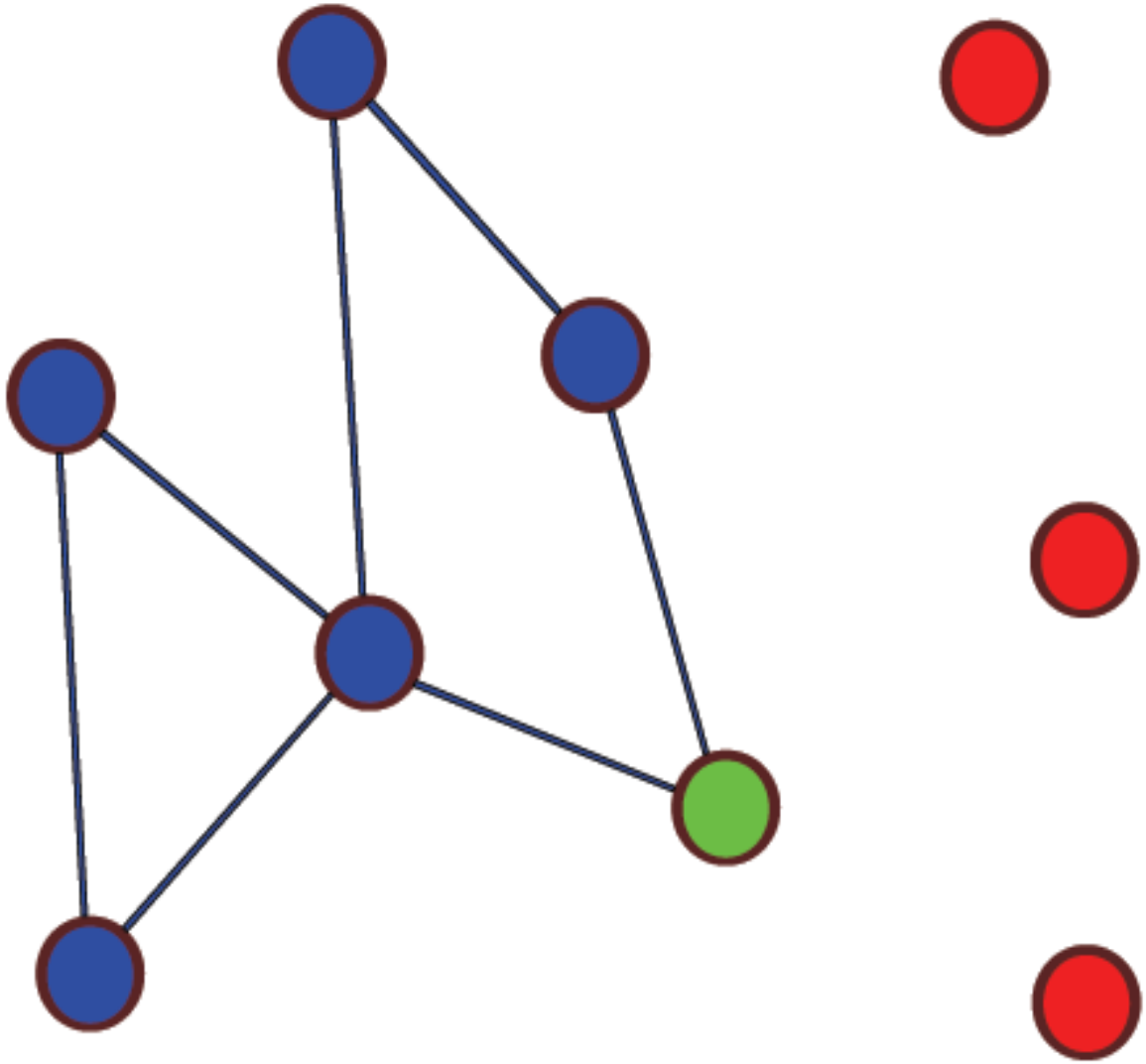
The LOCAL model



The LOCAL model



The LOCAL model



A few remarks about the local approach

- Weak hypothesis:
  - if A is connected to B,
  - if C is similar to B,
  - then A is likely to be connected to C.
- Computationally: much faster to train N local models with N training points each, than to train 1 model with N^2 training points.
- Caveats:
  - each local model may have very few training points
  - no sharing of information between different local models

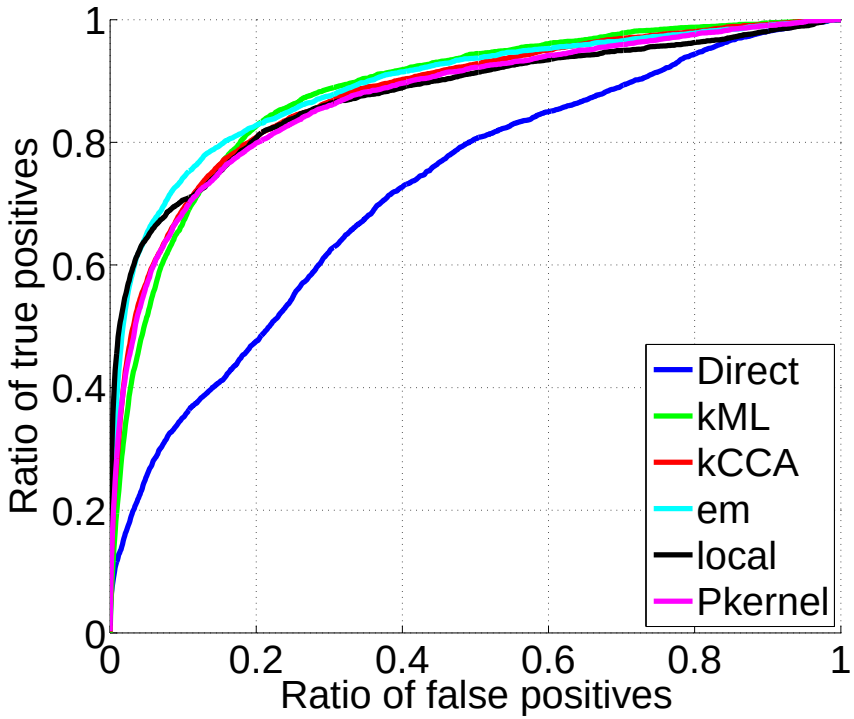
A few remarks about the local approach

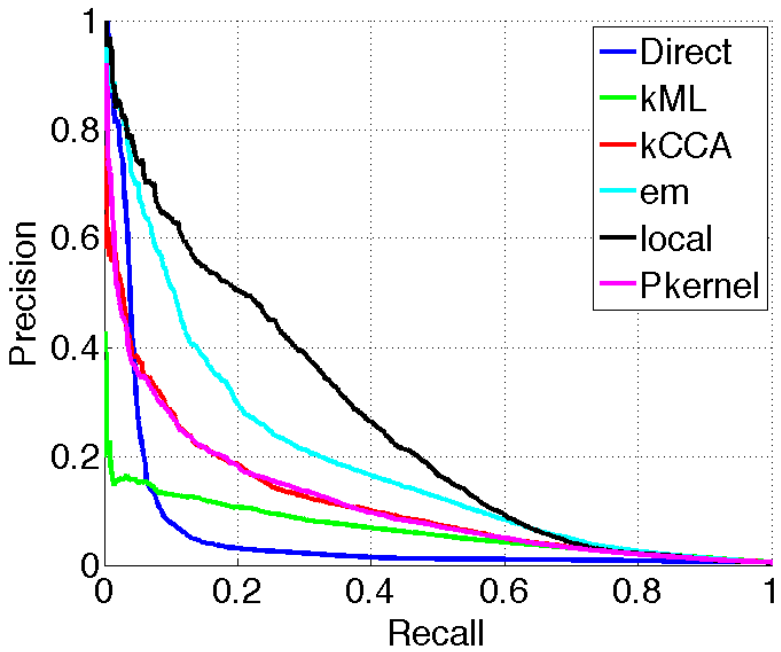
- Weak hypothesis:
  - if A is connected to B,
  - if C is similar to B,
  - then A is likely to be connected to C.
- Computationally: much faster to train N local models with N training points each, than to train 1 model with N^2 training points.
- Caveats:
  - each local model may have very few training points
  - no sharing of information between different local models

A few remarks about the local approach

- Weak hypothesis:
  - if A is connected to B,
  - if C is similar to B,
  - then A is likely to be connected to C.
- Computationally: much faster to train N local models with N training points each, than to train 1 model with N^2 training points.
- Caveats:
  - each local model may have very few training points
  - no sharing of information between different local models

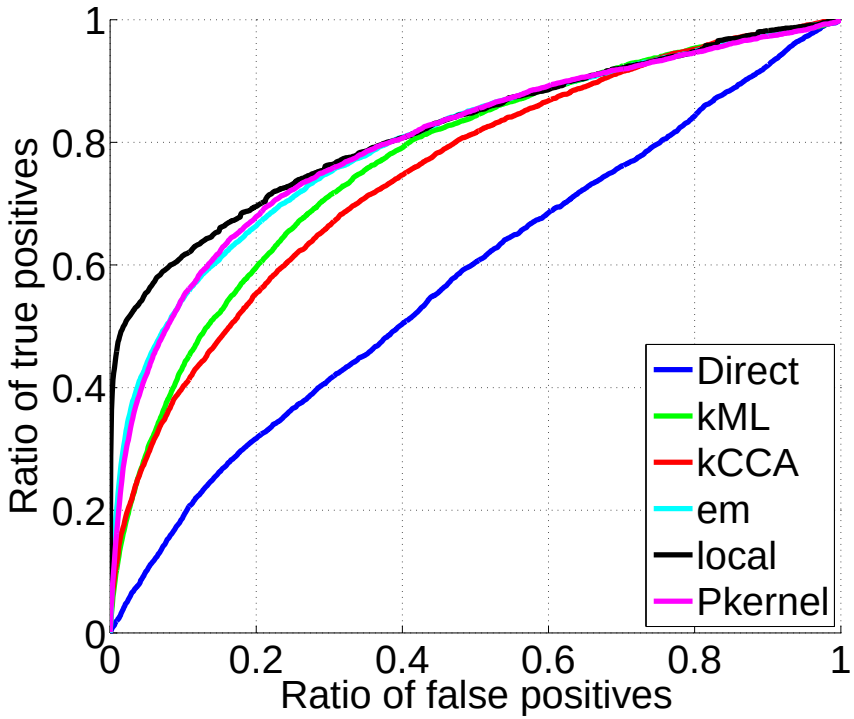
Results: protein-protein interaction (yeast)

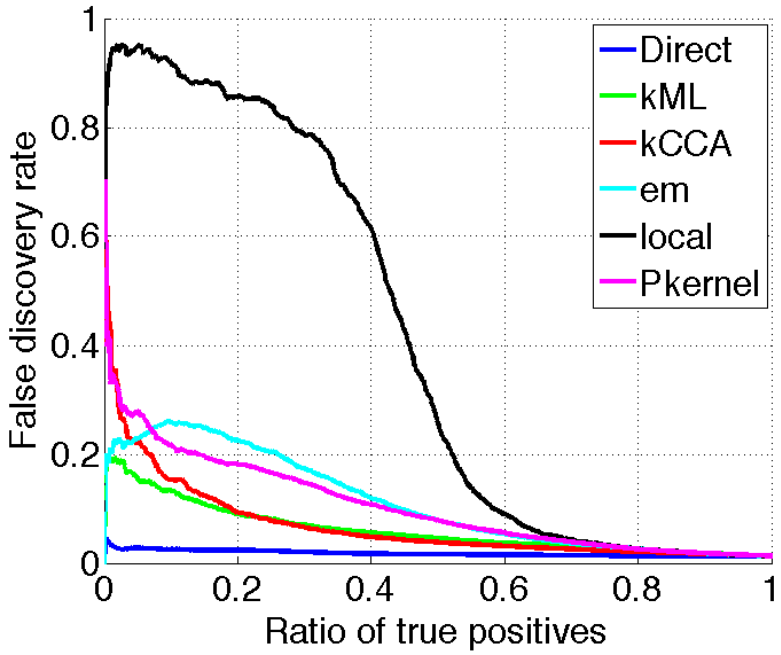




(from Bleakley et al., 2007)

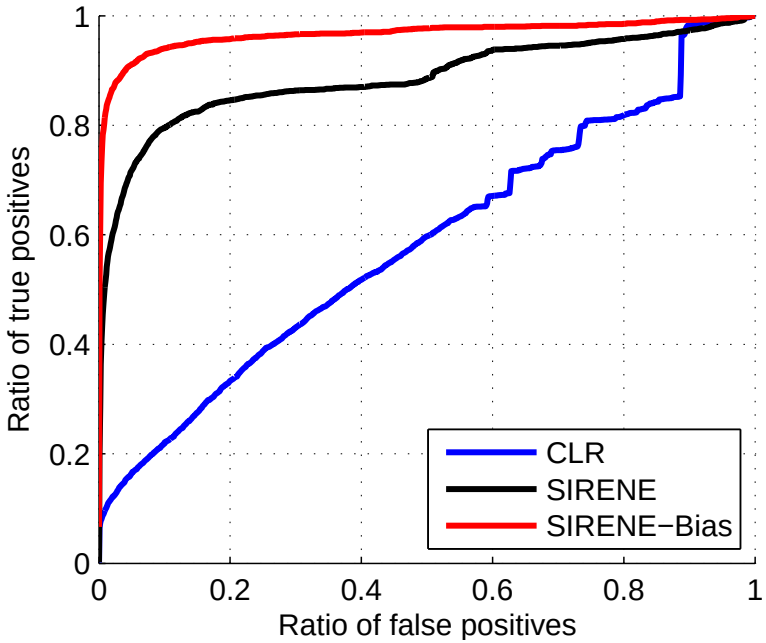
Results: metabolic gene network (yeast)

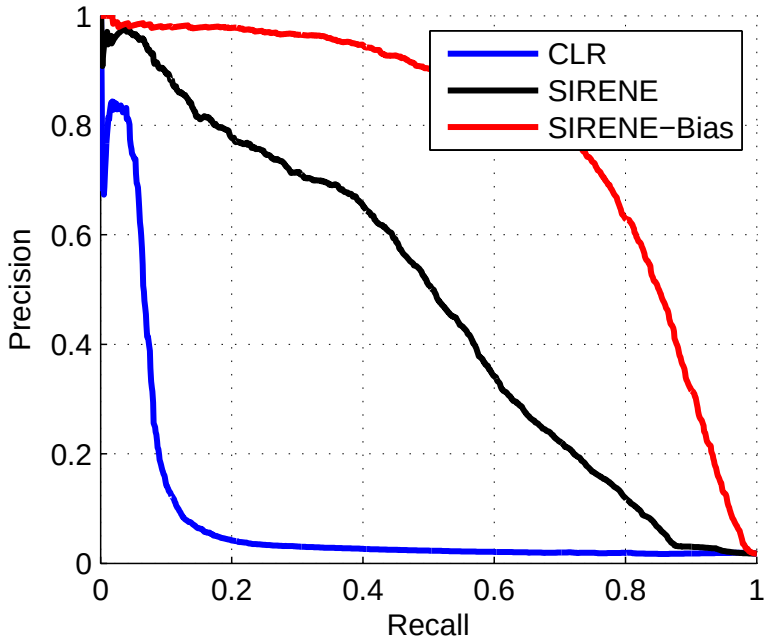




(from Bleakley et al., 2007)

Results: regulatory network (E. coli)





Method	Recall at 60%	Recall at 80%
SIRENE	**44.5%**	**17.6%**
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)

Applications: missing enzyme prediction

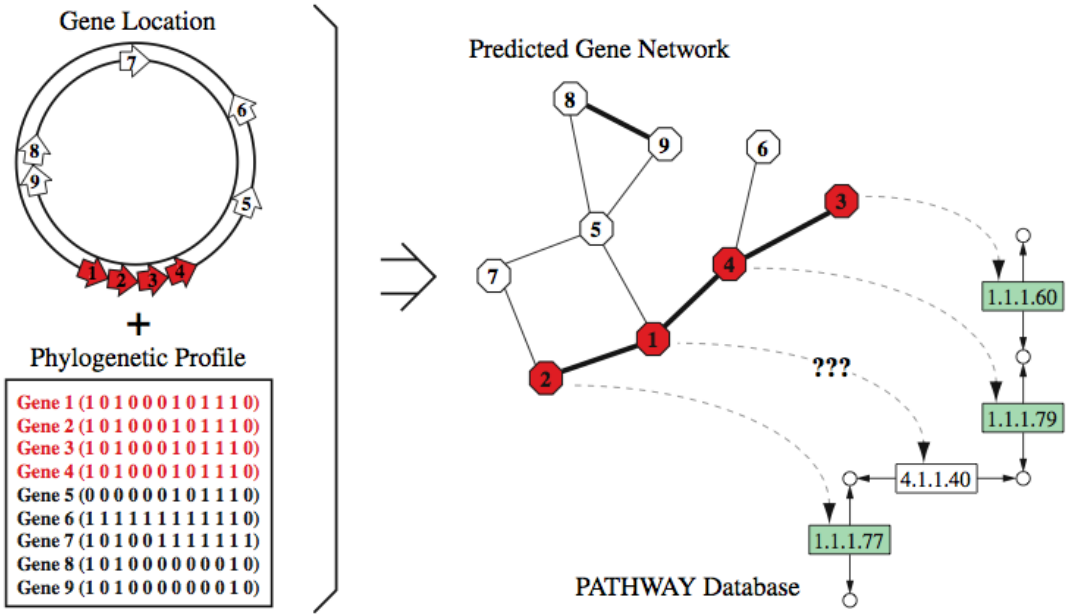
the FEBS Journal

Prediction of missing enzyme genes in a bacterial metabolic network

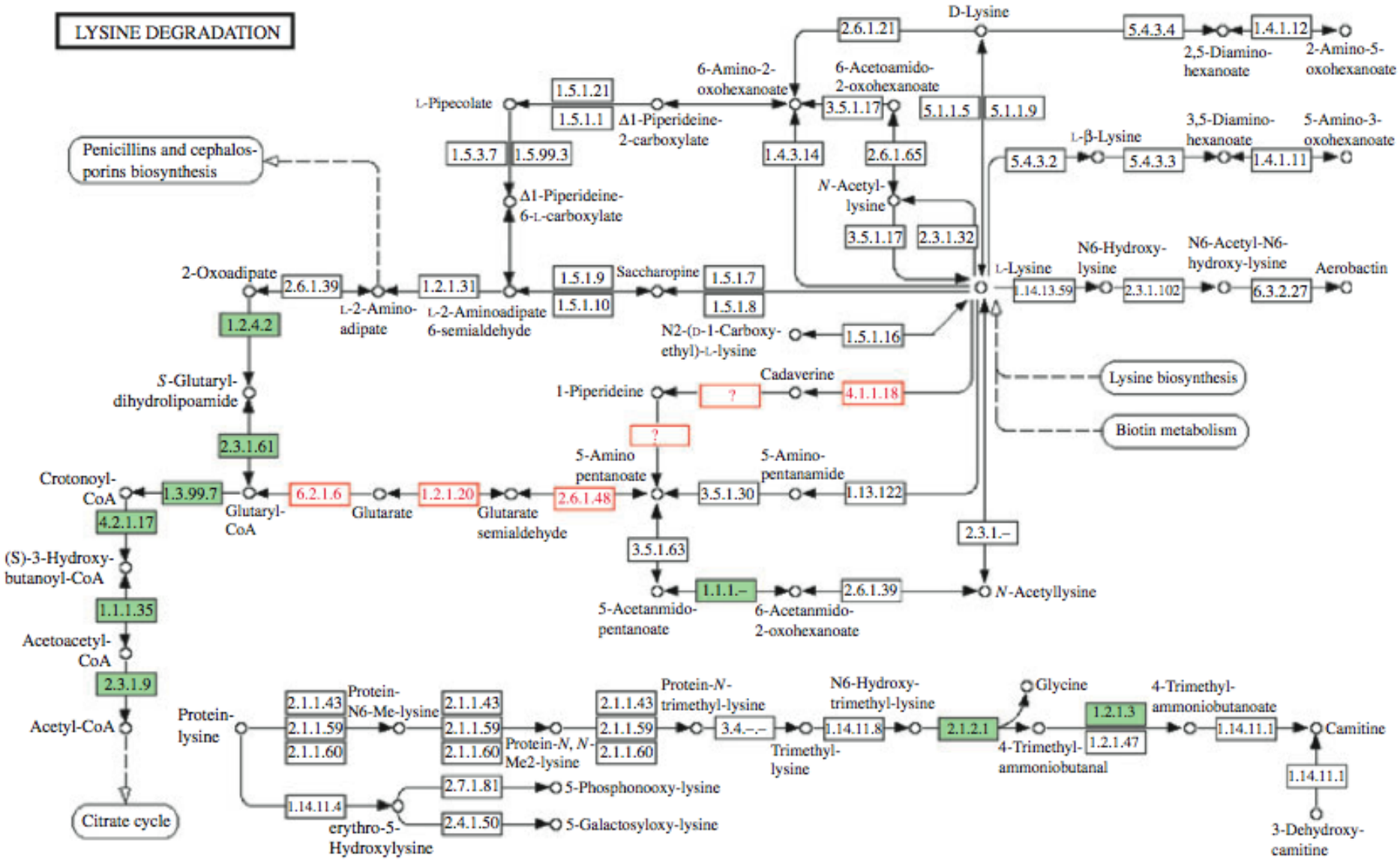
Reconstruction of the lysine-degradation pathway of *Pseudomonas aeruginosa*

Yoshihiro Yamanishi[1], Hisaaki Mihara[2], Motoharu Osaki[2], Hisashi Muramatsu[3], Nobuyoshi Esaki[2], Tetsuya Sato[1], Yoshiyuki Hizukuri[1], Susumu Goto[1] and Minoru Kanehisa[1]

1 Bioinformatics Center, Institute for Chemical Research, Kyoto University, Japan
2 Division of Environmental Chemistry, Institute for Chemical Research, Kyoto University, Japan
3 Department of Biology, Graduate School of Science, Osaka University, Japan



Applications: missing enzyme prediction



Applications: missing enzyme prediction

900 DOI 10.1002/pmic.200600862 *Proteomics* 2007, *7*, 900–909

RESEARCH ARTICLE

Prediction of nitrogen metabolism-related genes in *Anabaena* by kernel-based network analysis

Shinobu Okamoto[1], Yoshihiro Yamanishi[1], Shigeki Ehira[2], Shuichi Kawashima[3], Koichiro Tonomura[1]** and Minoru Kanehisa[1]*

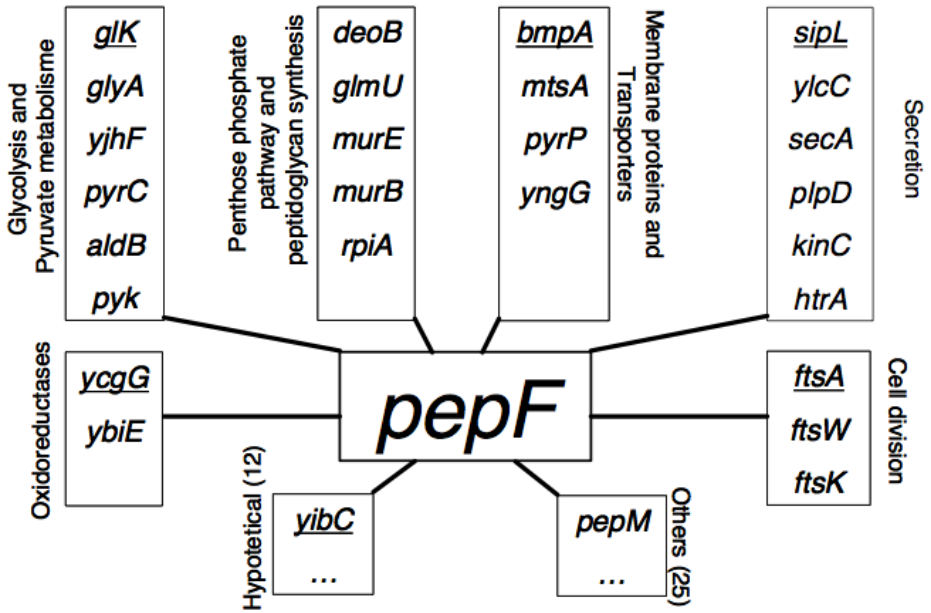
[1] Bioinformatics Center, Institute for Chemical Research, Kyoto University, Uji, Japan
[2] Department of Biochemistry and Molecular Biology, Faculty of Science, Saitama University, Saitama, Japan
[3] Human Genome Center, Institute of Medical Science, University of Tokyo, Meguro, Japan

Applications: function annotation

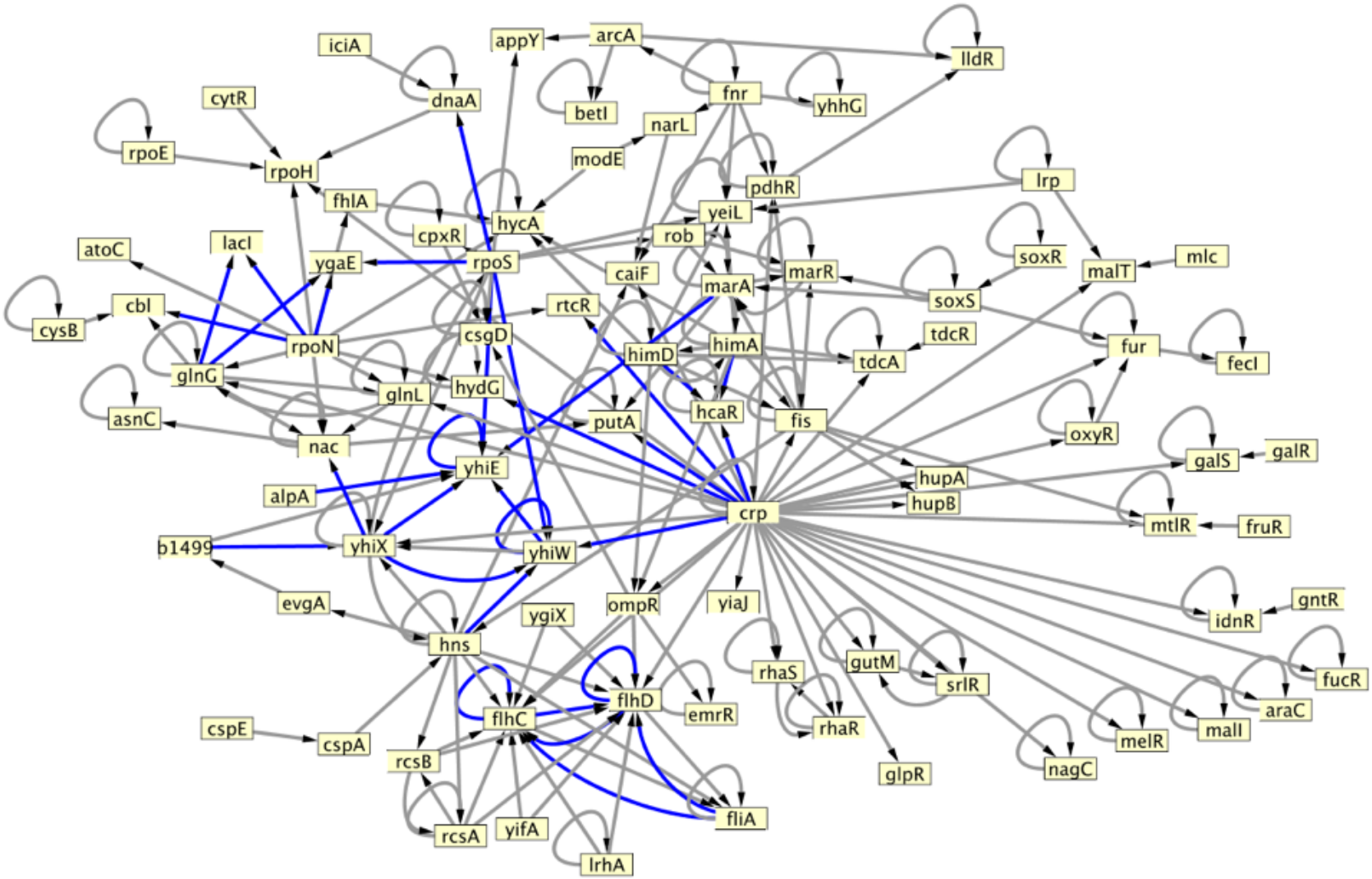
Determination of the role of the bacterial peptidase PepF by statistical inference and further experimental validation

Liliana LOPEZ KLEINE[1,2], Alain TRUBUIL[1], Véronique MONNET[2]

[1]Unité de Mathématiques et Informatiques Appliquées. INRA Jouy en Josas 78352, France.
[2]Unité de Biochimie Bactérienne. INRA Jouy en Josas 78352, France.



Application: predicted regulatory network (E. coli)



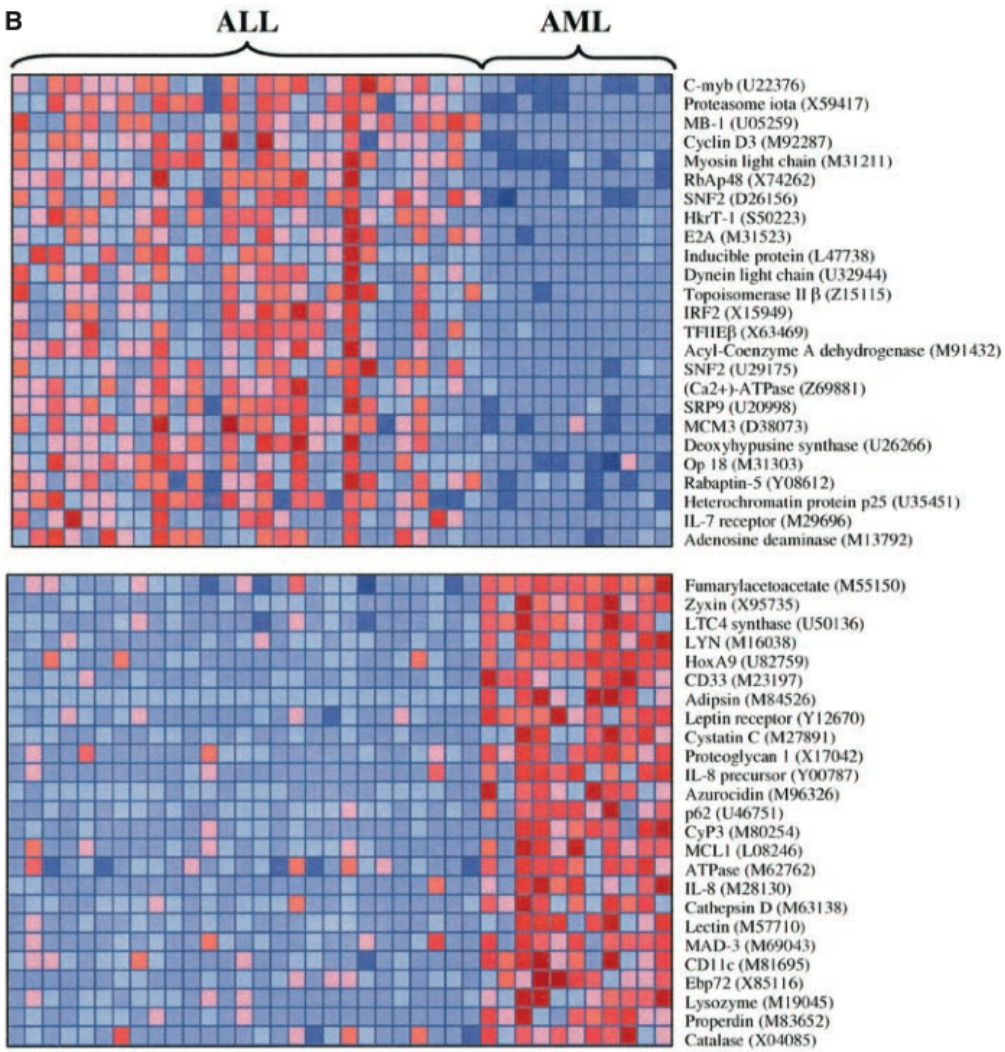
Prediction at 60% precision, restricted to transcription factors (from Mordelet and V., 2008).

Outline

1. How to infer relationships between genes from biological data?

2. How to use biological networks to help in the analysis of genomic data?

Tissue classification from microarray data



Goal

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

Issue

20K+ genes but only <100 tumours

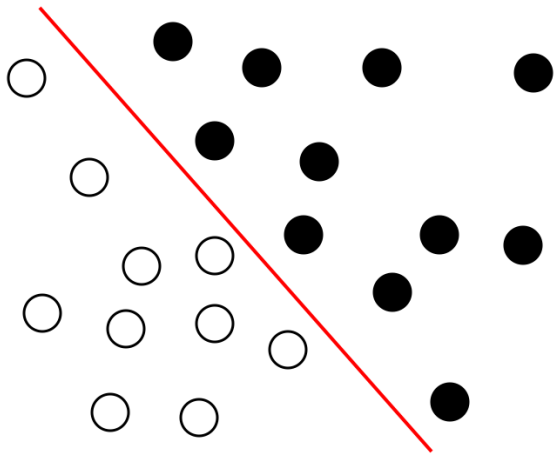
Linear classifiers and signatures

The model

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

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

- Interpretability: the weight β_i quantifies the influence of feature i (but...)



Linear classifiers

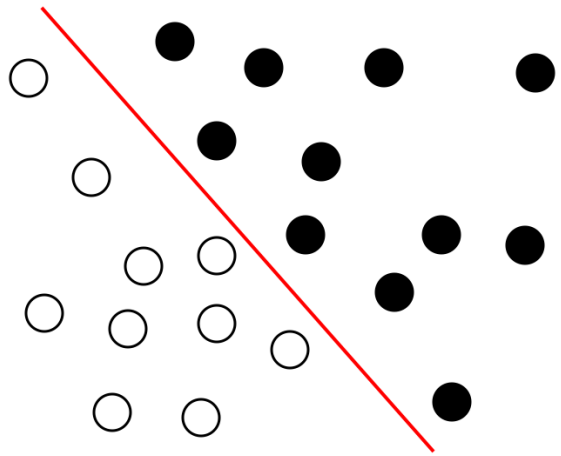
Training the model

- 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 β, e.g.:

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



Classical penalties

- Feature selection (NP-hard, many greedy variants exist):

$$\Omega_{\text{Best subset selection}}(\beta) = \| \beta \|_0 = \sum_{i=1}^{p} 1(\beta_i > 0) \, .$$

- Small weights (SVM, ridge regression, ...):

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

- Sparsity-inducing convex priors (computationnally tractable + feature selection):

$$\Omega_{\text{LASSO}}(\beta) = \| \beta \|_1 = \sum_{i=1}^{p} | \beta_i | \, .$$

Classical penalties

- Feature selection (NP-hard, many greedy variants exist):

$$\Omega_{\text{Best subset selection}}(\beta) = \| \beta \|_0 = \sum_{i=1}^{p} 1(\beta_i > 0) \, .$$

- Small weights (SVM, ridge regression, ...):

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

- Sparsity-inducing convex priors (computationnally tractable + feature selection):

$$\Omega_{\text{LASSO}}(\beta) = \| \beta \|_1 = \sum_{i=1}^{p} | \beta_i | \, .$$

Classical penalties

- Feature selection (NP-hard, many greedy variants exist):

$$\Omega_{\text{Best subset selection}}(\beta) = \| \beta \|_0 = \sum_{i=1}^{p} 1(\beta_i > 0) \, .$$

- Small weights (SVM, ridge regression, ...):

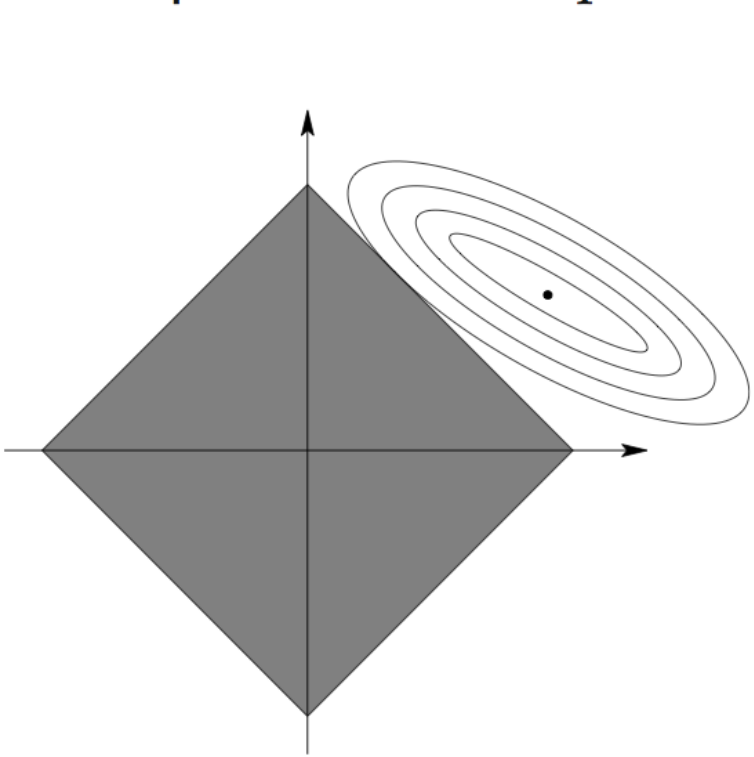
$$\Omega_{\text{ridge}}(\beta) = \| \beta \|_2^2 = \sum_{i=1}^{p} \beta_i^2 \, .$$

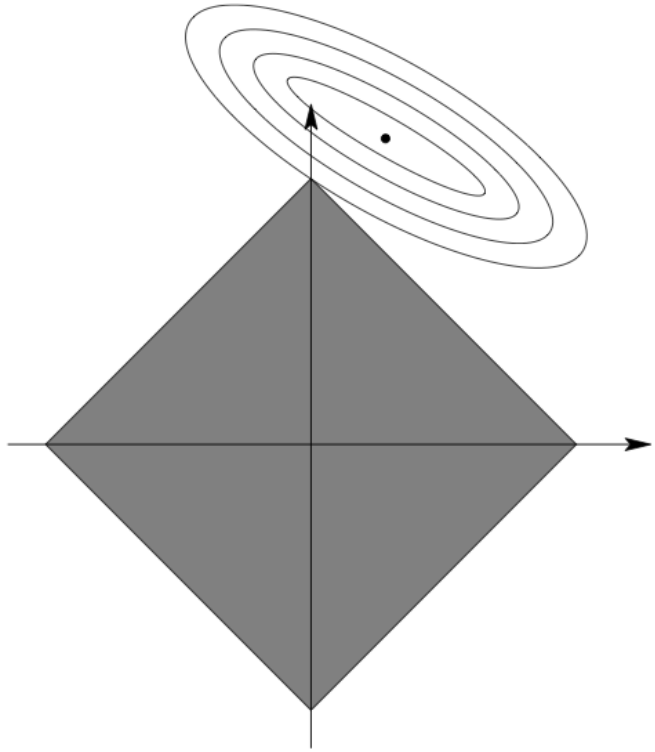
- Sparsity-inducing convex priors (computationnally tractable + feature selection):

$$\Omega_{\text{LASSO}}(\beta) = \| \beta \|_1 = \sum_{i=1}^{p} | \beta_i | \, .$$

Why LASSO leads to sparse solutions

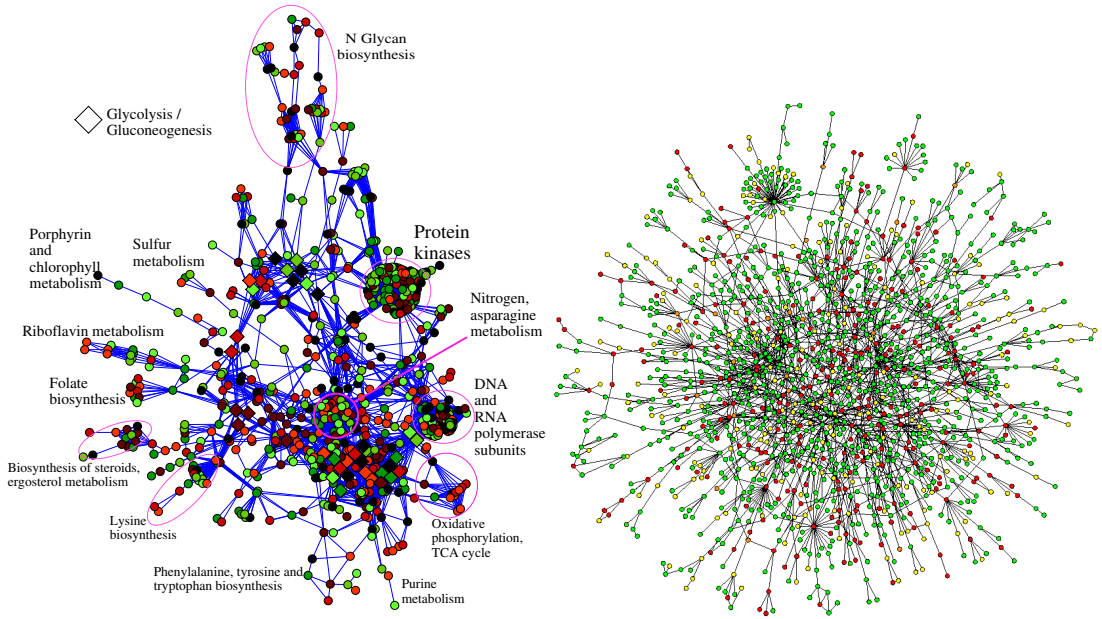
Geometric interpretation with $p = 2$





How protein networks can help us

- Basic biological functions usually involve the coordinated action of several proteins:
  - Formation of protein complexes
  - Activation of metabolic, signalling or regulatory pathways
- Many pathways and protein-protein interactions are already known
- Hypothesis: the signature should be "coherent" with respect to this prior knowledge

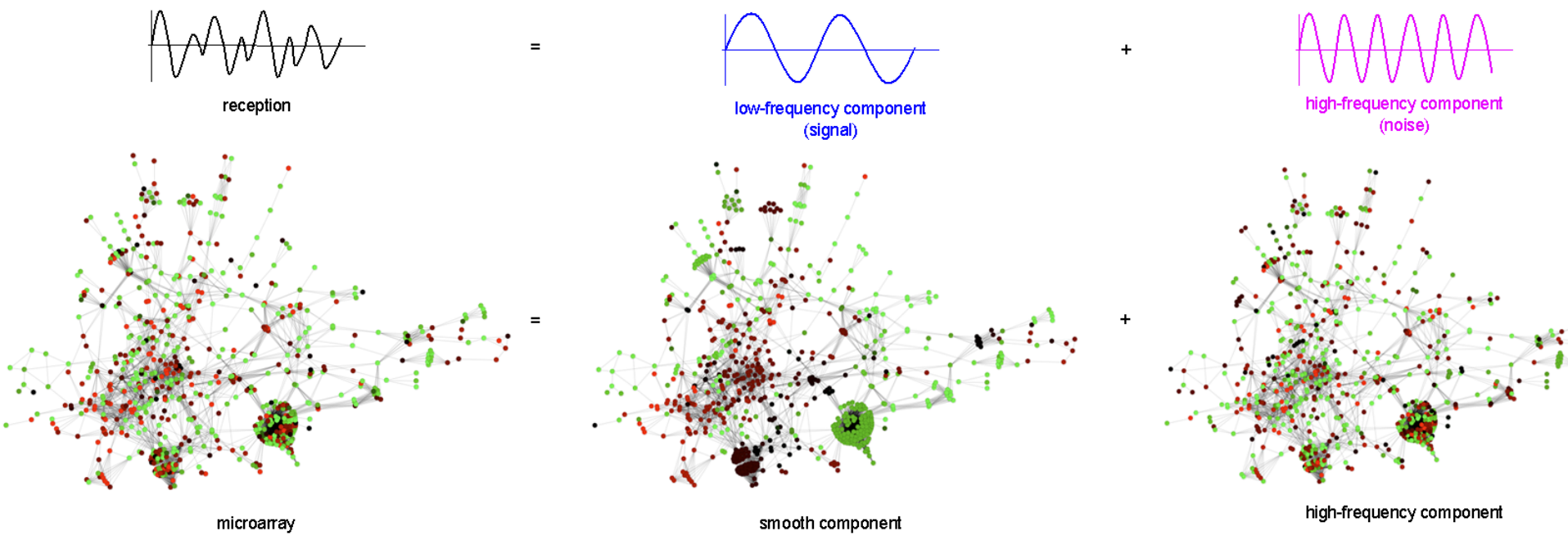


Example: smooth signature

- Hypothesis: adjacent genes should have similar weights in the signature
- Penalty function (Rapaport et al., 2007):

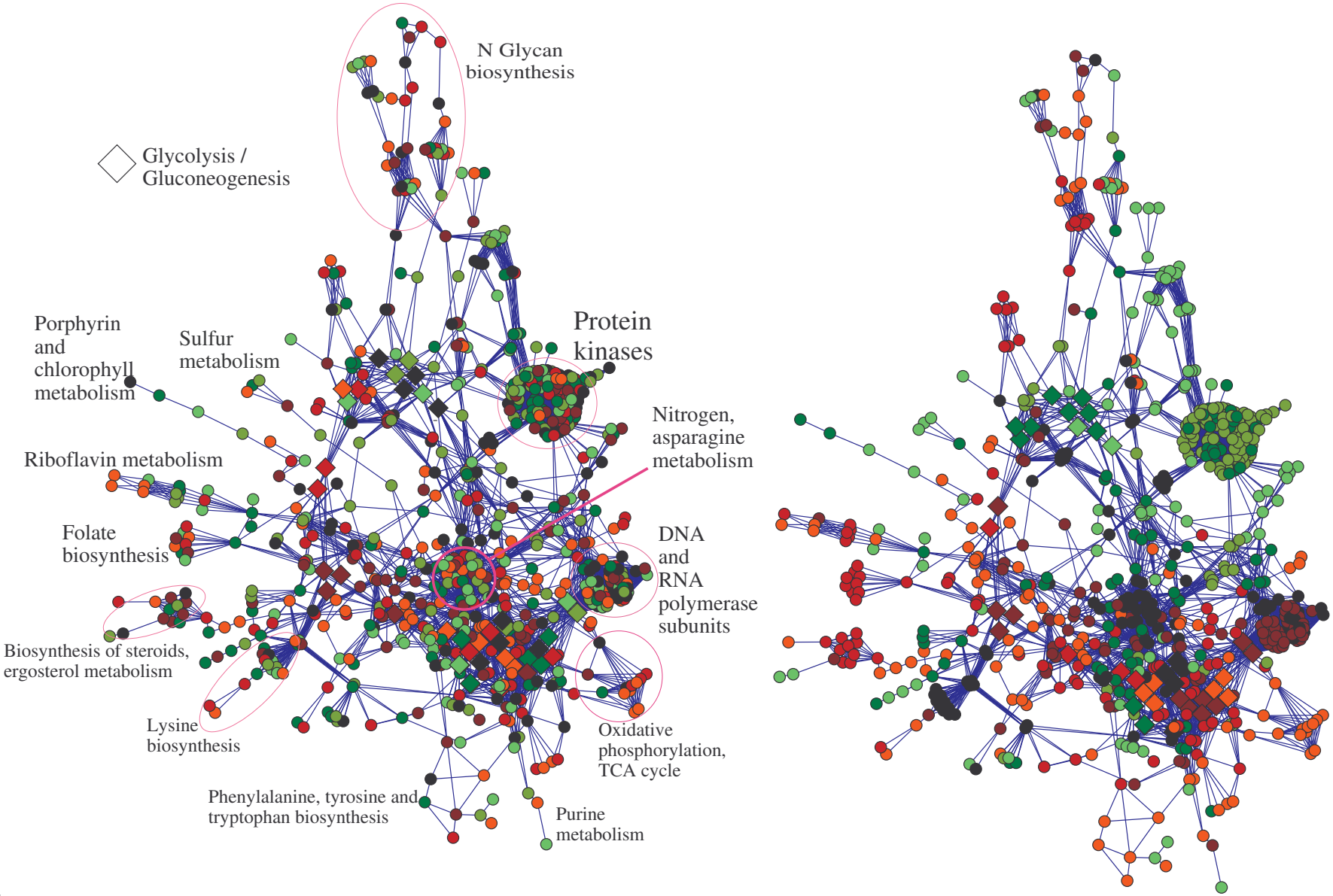
$$\Omega_{smooth}(\beta) = \sum_{i \sim j} (\beta_i - \beta_j)^2$$

Equivalent formulation

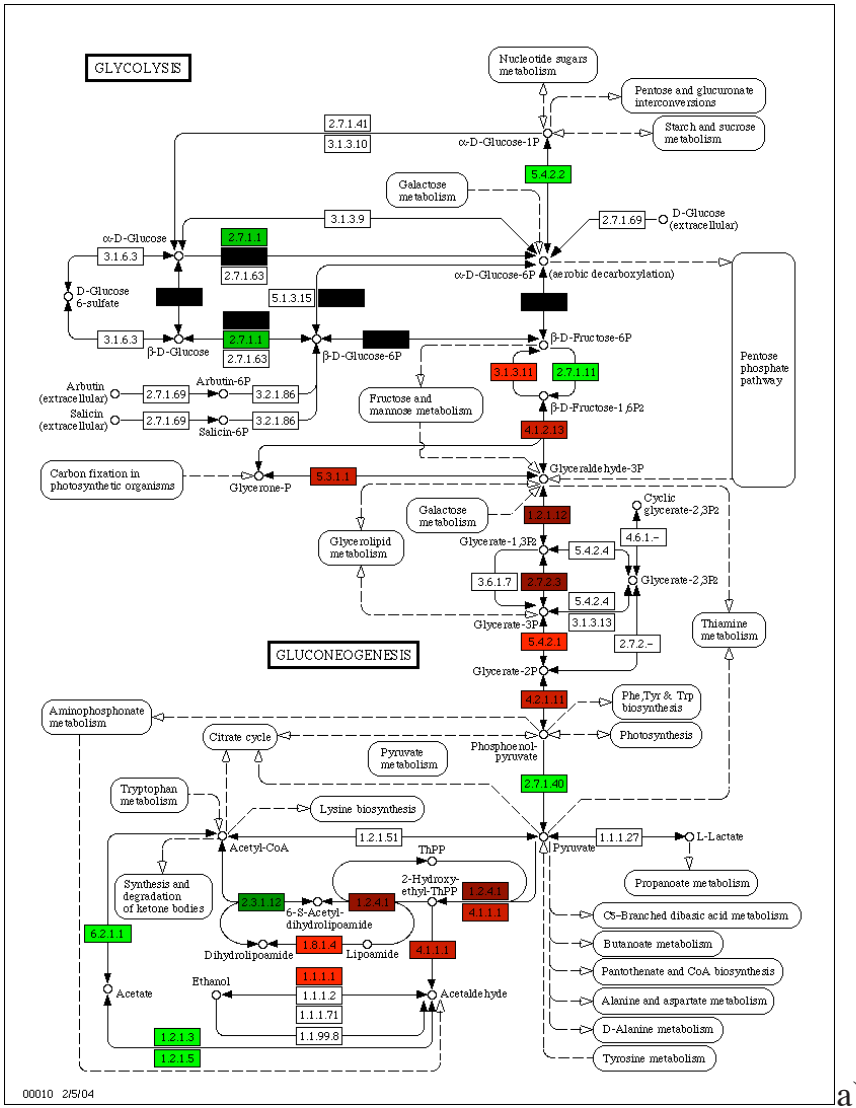


1. Use the gene network to extract the "important information" in gene expression profiles by Fourier analysis on the graph
2. Learn a linear classifier on the smooth components with classical ridge penalty.

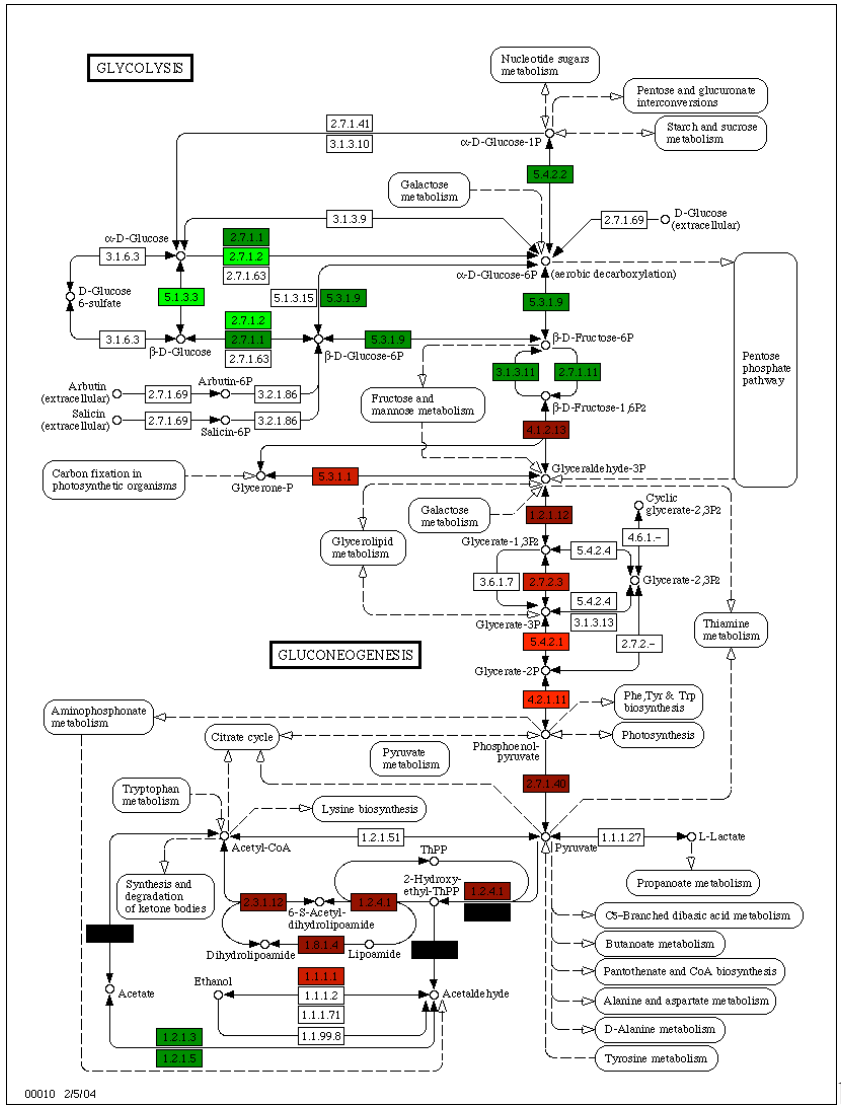
Illustration (yeast, high vs. low irradiation doses



Signatures



a)



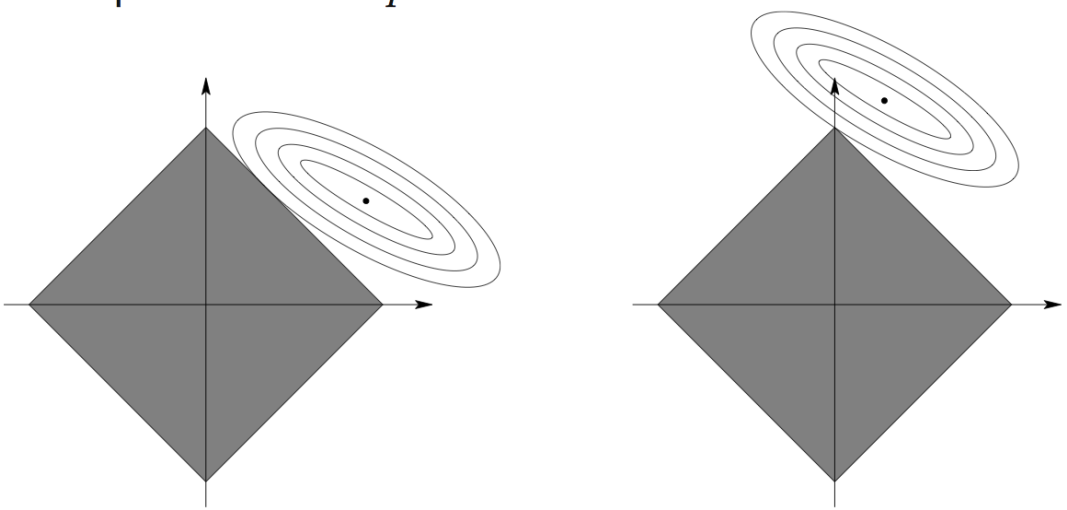
b)

Example: smooth and sparse signature

- Hypothesis:
  - the signature should be sparse (gene selection)
  - connected genes should have the same weight
- Penalty function (Rapaport et al., 2008):

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

Geometric interpretation with $p = 2$



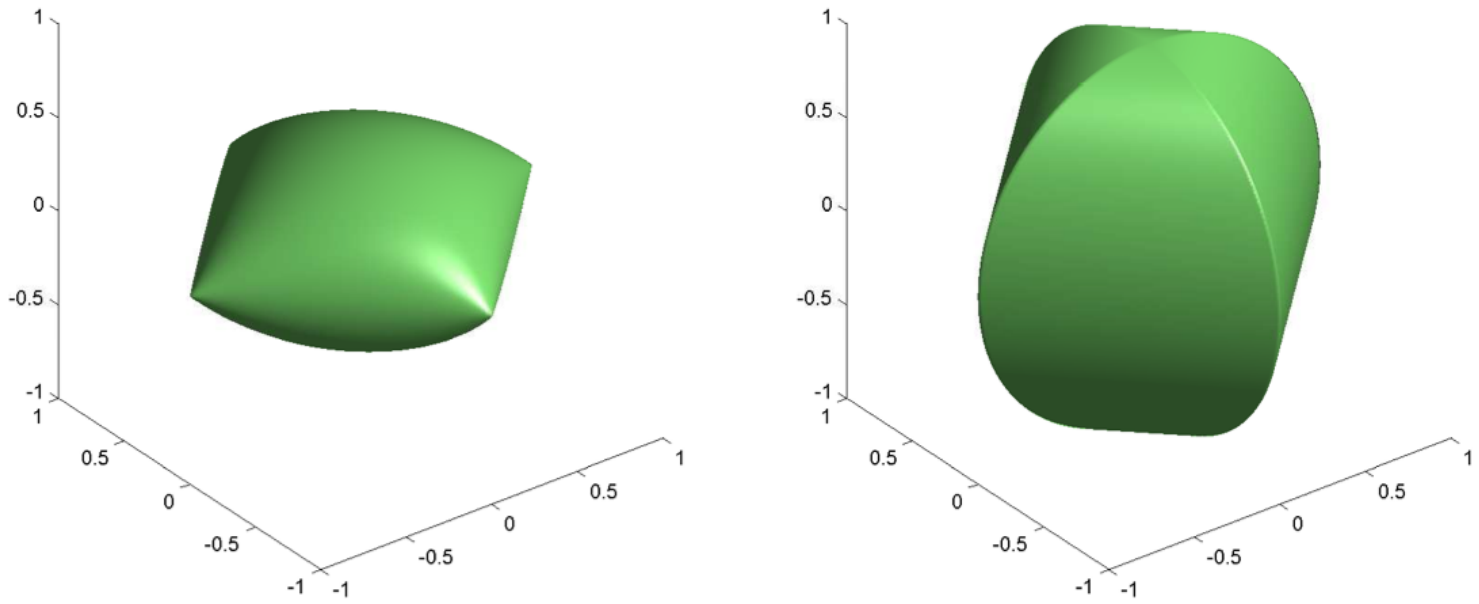
Example: sparse pathway signature

- Hypothesis:
  - the signature should be sparse (gene selection)
  - selected genes should form dense connected components (without any constraint of their relative weights)
- Penalty function (Jacob et al., 2009):

$$\Omega_{intersection}(\beta) = \sum_{i \sim j} \sqrt{\beta_i^2 + \beta_j^2}\,,$$

$$\Omega_{union}(\beta) = \sup_{\alpha \in \mathbb{R}^p : \forall i \sim j, \|\alpha_i^2 + \alpha_j^2\| \leq 1} \alpha^\top \beta\,.$$

Graph LASSO leads to structured sparsity



Groups $(1,2)$ *and* $(2,3)$*. Left:* $\Omega_{intersection}(\beta)$*. Right:* $\Omega_{union}(\beta)$*. Vertical axis is* β_2*.*

Preliminary results

Breast cancer data

- Gene expression data for 8,141 genes in 295 breast cancer tumors.
- Canonical pathways from MSigDB containing 639 groups of genes, 637 of which involve genes from our study.

METHOD	ℓ_1	$\Omega^{\mathcal{G}}_{\text{OVERLAP}}(.)$
ERROR	0.38 ± 0.04	0.36 ± 0.03
♯ PATH.	$148, 58, 183$	$6, 5, 78$
PROP. PATH.	$0.32, 0.14, 0.41$	$0.01, 0.01, 0.17$

- Graph on the genes.

METHOD	ℓ_1	$\Omega_{graph}(.)$
ERROR	0.39 ± 0.04	0.36 ± 0.01
AV. SIZE C.C.	$1.1, 1, 1.0$	$1.3, 1.4, 1.2$

Conclusion

- A supervised machine learning formulation leads to promising results on the problem of inferring unknown relationships between genes and proteins.
- Conversely, biological networks can help fighting the curse of dimensionality for classification of high-dimensional genomic data
- All this is progressing very quickly these days!

People I need to thank

- Graph inference : Yoshihiro Yamanishi, Minoru Kanehisa (Univ. Kyoto), Jian Qian, Bill Noble (Univ. Washington), Kevin Bleakley, Gerard Biau (Univ. Montpellier), Fantine Mordelet (ParisTech)
- Using graphs : Franck Rapaport, Emmanuel Barillot, Andrei Zinovyev (Institut Curie), Laurent Jacob (ParisTech), Guillaume Obozinski (Berkeley / INRIA)

Acknowledgement

This presentation is supported by a JSPS Invitation Fellowship Program for Research in Japan, hosted by Tatsuya Akutsu (Kyoto University)