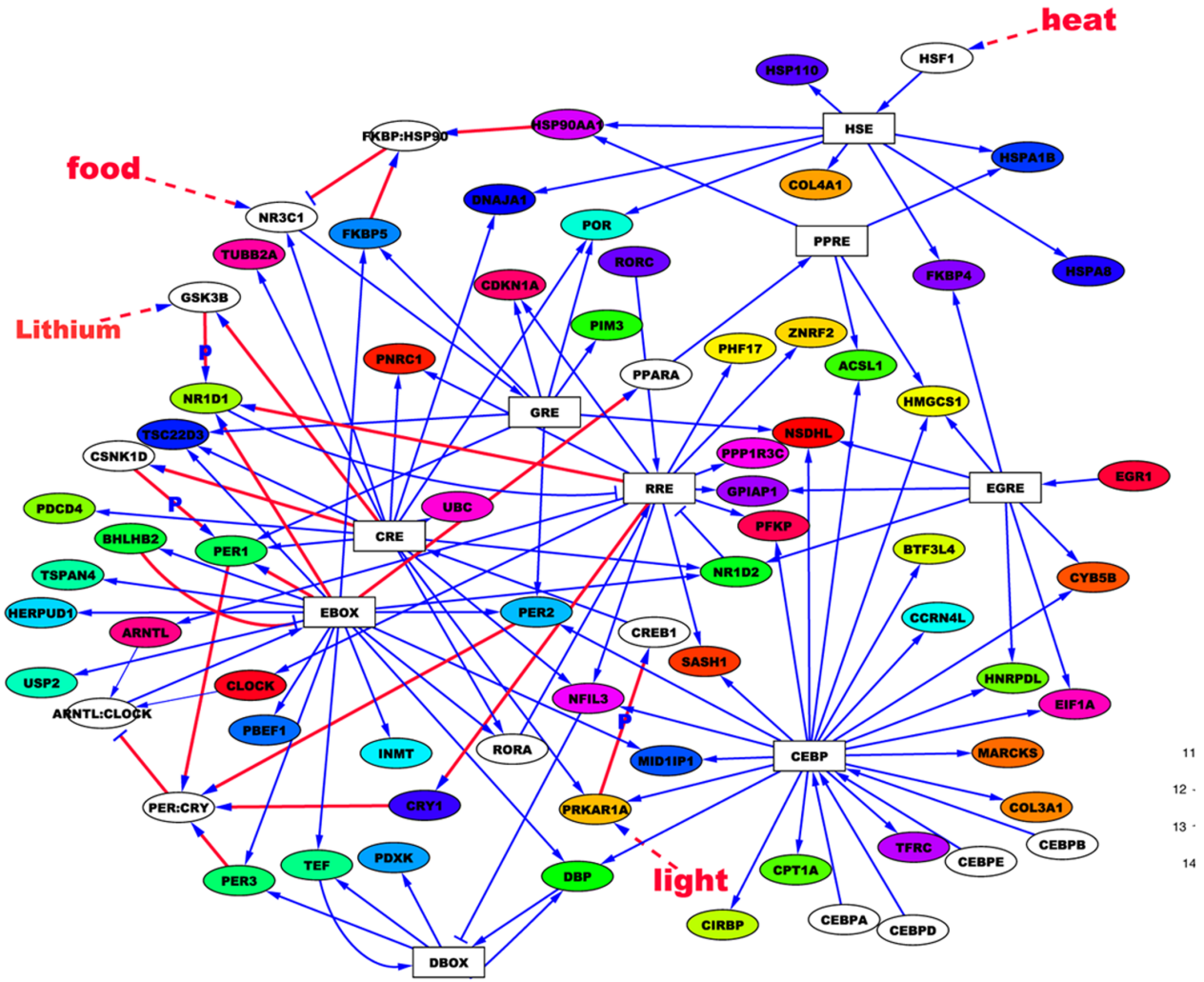
Biological network inference: a challenge to structured data-mining

Florence d'Alché-Buc[1,2]

[1]Visiting: INRIA, LRI umr CNRS 86 23, Université Paris-Sud, Orsay, France
[2]Permanent address: IBISC EA 4526, Université d'Évry Val d'Essonne, Évry cedex, France

Outline

Networks in molecular biology



- Circadian clock in mouse (Fig: Yan et al. Plos Comp. Biol. 2008)

Gene regulatory networks

Circadian clock (CC)

- An example of a gene regulatory network involved in a cellular response to some input signal
- Involved in sustaining 24h-oscillations
- Feedback loop allows for control
- Cellular response to day-night alternance, meals etc...

Gene regulatory networks

Definition of *Transcriptional regulation*

A gene a is said to regulate gene b if a codes for the protein A and A is a transcription factor of gene b.
When A allows for the initiation of the transcription of b, gene a is said to induce gene b. When A blocks the transcription, gene a is said to inhibit gene b.

Gene regulatory networks

Definition of *Transcriptional regulation*

A gene a is said to regulate gene b if a codes for the protein A and A is a transcription factor of gene b.
When A allows for the initiation of the transcription of b, gene a is said to induce gene b. When A blocks the transcription, gene a is said to inhibit gene b.

A simple definition of a *gene regulatory network*

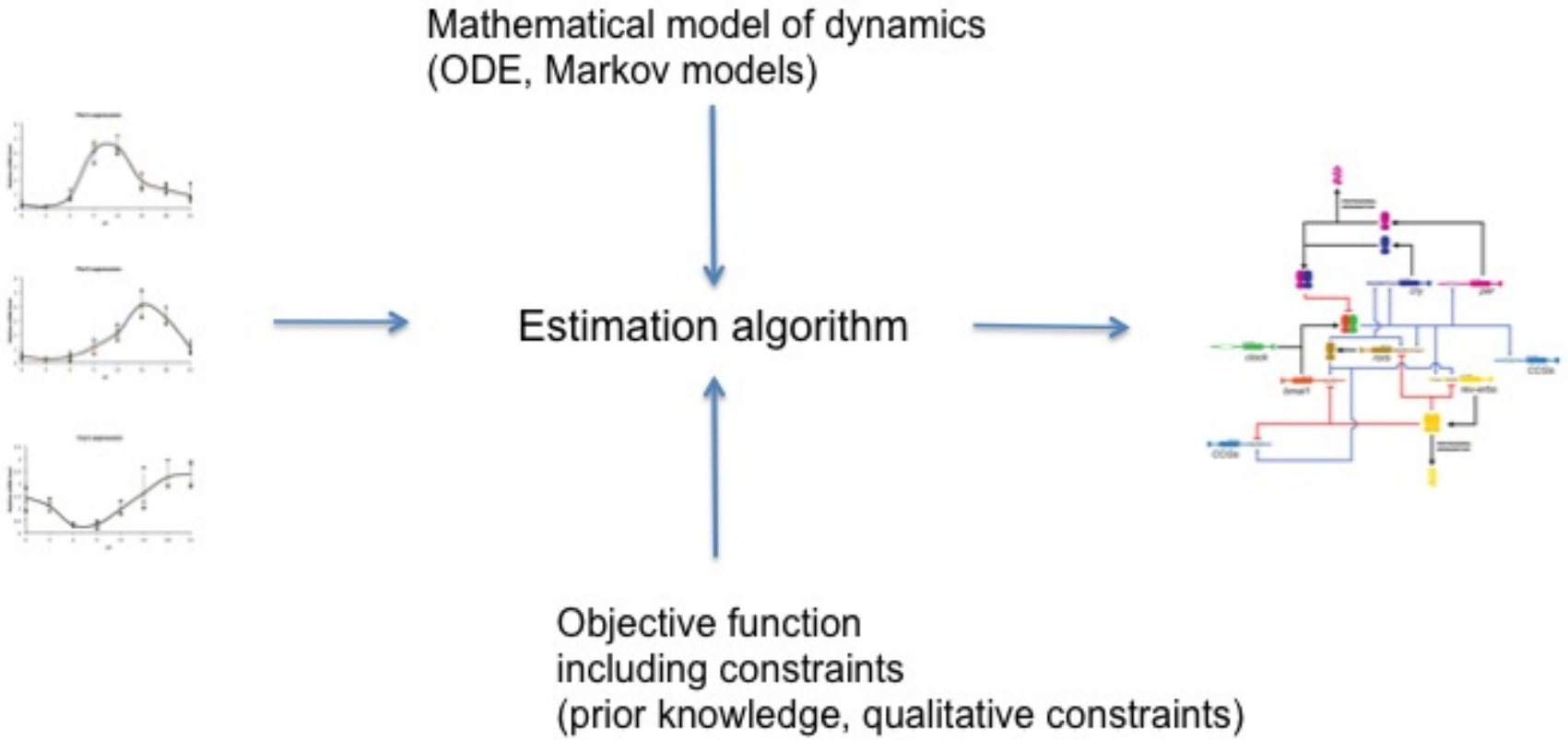
A gene (transcriptional) regulatory network is a dynamical system whose state variables are the mRNA's concentrations (possibly the proteins concentrations) and evolve through time.

Reverse-engineering of gene regulatory networks

- Identify complex regulatory mechanisms at work in the cell
- Motivations: better understanding, predictive models for therapeutical targetting, biomarkers, personalized medecine, ...
- **Main tasks related to data-mining**
  - Parameter estimation in gene regulatory network models
  - Network inference

Biological network Inference

Reverse-modeling of signalling and gene regulatory networks



ANR ODESSA- midterm report – 20/09/2011

High throughput measurement techniques

Experimental techniques

- DNA chips
- Next Generation Sequencing methods
  - RNA Seq
  - Chip-seq

Data

- Gene expression level of tissue at a given time point after a long-term "run": steady state
- Time-course of gene expression: expression levels measured at a given time point for a given organism (one time-point: one organism)
- Perturbation data: Knock-out or knock-down of a gene and measurement of the gene expression level

Difficulties and limitations

- Intrinsic noise, measurement on a cell population, measurement noise
- Limited size of data [thousands of genes, tens of measurement] $\neq$ BIG DATA
- Missing knowledge about the timing of regulation
- Nonlinearity of the behaviors
- Missing observations
- Other actors: role of chromatine, other kinds of regulations

Hopes

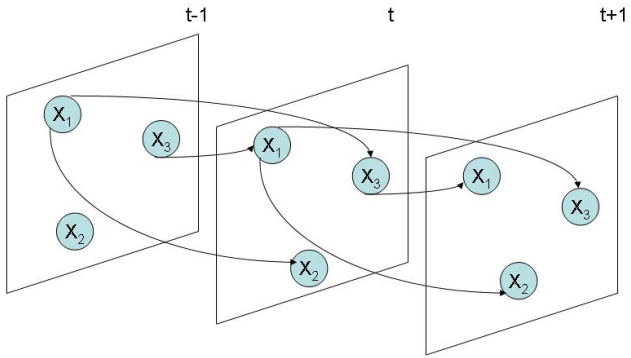
- Success of various clustering methods
- Reproducibility of data and behaviors
- Many other sources of knowledge/data: known functions of proteins, list of transcription factors, protein-protein interactions, metabolism (when relevant),...
- Multiple-view data
- Progress of experimental measurements and cost reduction

Gene regulatory network inference as a learning task

- **Dimension reduction**: clustering, biclustering
- **Supervised link prediction** : SIRENE (Mordelet et al. 2008)
- **Model-free approaches to estimate the network structure**: ARACNE (Margolin et al. 2006)
- **Model-based and unsupervised approaches**: Bayesian networks (Pe'er et al. 2001, Segal et al. 2003, ...), graphical Gaussian models (Strimmer et al. 2006)

Network inference from time series

- **Measurements of coupling**: Kramer at al. 2009, mutual information : Zoppoli et al. 2010
- **Differential equations**: linear equations (Chen 1999), non-linear: S-systems (Voit et al. 2006)
- **Autoregressive models**
  - $\mathbf{x}_{t+1} = h(\mathbf{x}_t) + \epsilon_t$, t > 0, $\mathbf{x}_t$: state vector, h $\in \mathcal{H}$, ϵ_t : iid gaussian noise
- **Dynamic Bayesian models**
  - $x^i_{t+1} = h_i(Pa(i,t)) + \epsilon_t$, t > 0, Pa(i,t): parent state variables at time t



Dynamic bayesian network without hidden variables

Network inference with autoregressive models

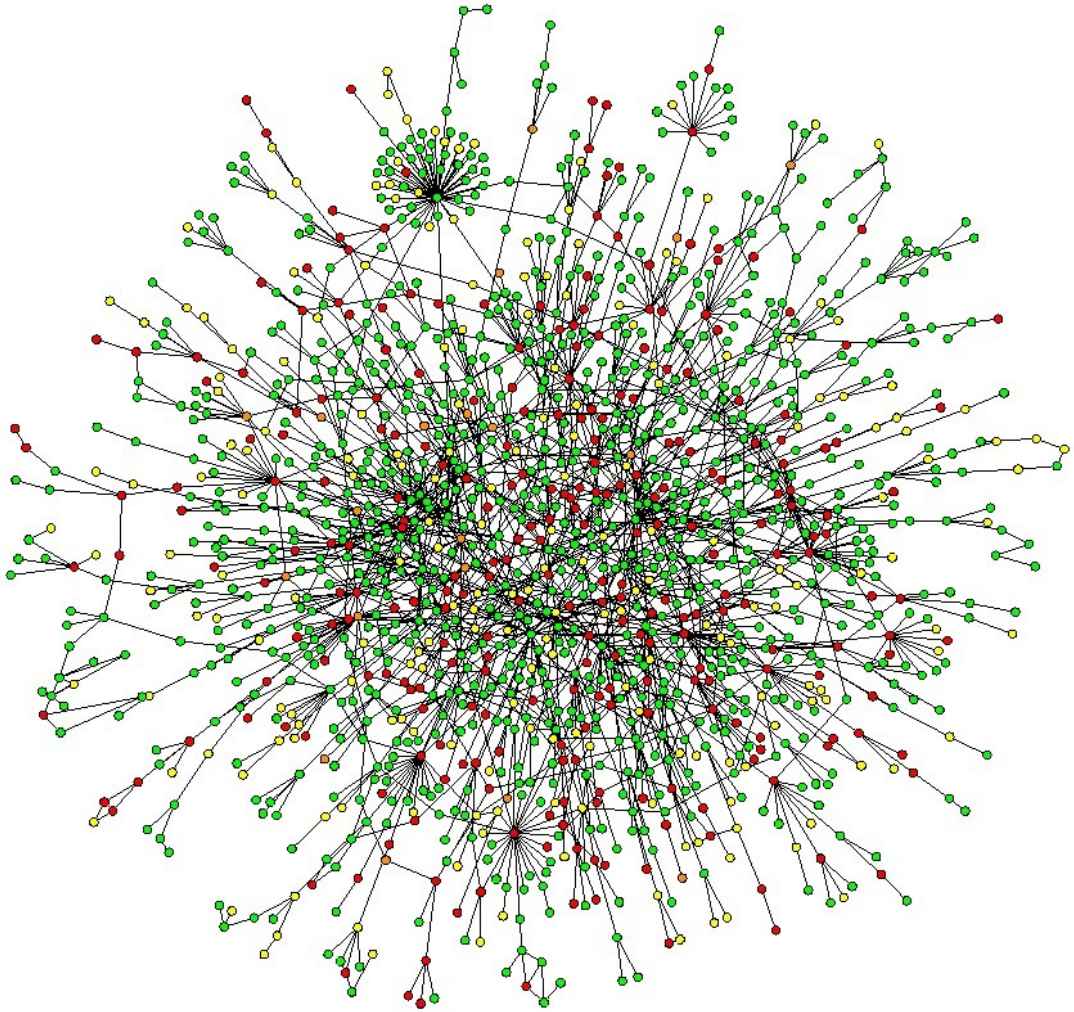
- **Sparse linear models**
  - Linear autoregressive models (Opgen-Rhein and Strimmer, 2007; Fujita et al. 2007, Shimamura et al. 2009)
  - Granger causality (Shojaie and Michailidis, 2010 and 11)
  - State-space models (Perrin et al. 2003, Rangel et al. 2004)
  - Several order autoregressive models (Lozano 2009, Bolstad et al. 2011)
  - Time-varying models (Lebre et al. 2011)

Network inference with nonlinear autoregressive models

- **Nonlinear nonparametric models**
  - Dynamic Bayesian Networks (Imoto et al. 2002, Husmeier et al. 2005, Li et al. 2007, Bansal et al. 2007)
  - Gaussian processes for network inference (Aijo and Lahdesmaki 2009)

Outline

Protein-protein interaction network



- Protein-protein interaction network in yeast

Experimental detection of protein-protein interactions

- *in vivo* large scale systems:
  - *Y2H* high false positive rate
- *in vitro* small scale methods: costly and laborious
  - protein-arrays
  - co-immunoprecipitations
  - FRET, NMR

Data

Combine indirect information on pairs of proteins with direct information on ppi data

- over-represented domains or motifs pairs
- structural information, primary sequences
- subcellular localization
- biological functions
- co-expression of genes
- conservation of pairs of sequences

Difficulties

- Generally no structural information
- Very few labeled edges and no negative labels
- Relevant features ? Context ?
- Source of knowledge: some information used as input feature have been inferred from the outputs

Hopes

- Global improvement of datasets and databases
- Better encoding of structured knowledge
- Transfer learning, multi-task learning

Existing approaches for link prediction (1): supervised classification

- Pairwise SVM [Ben-Hur and Noble 2005]
- Random forest, mixture of feature experts (Qi 2008), ensemble methods with original evolutionary features (De Vienne and AzŐ, 2012)
- Supervised Learning of a kernel or a similarity
  - With KCCA [Yamanishi et al. 2004], with metric learning [Yamanishi and Vert 2005]
  - With output kernel regression tree [Geurts et al. 2006,07], with output kernel gradient boosting [Geurts et al. 2007]
- Supervised classification linked to a node
  - local classifiers [Bleakley et al. 2007]

Existing approaches (2): semi-supervised and transductive learning

- Kernel Matrix completion
  - Using EM [Tsuda et al. 2003] and [Kato et al. 2005]
- Transductive or semi-supervised learning
  - Link Propagation [Kashima et al. 2009]
  - Mixture of Wishart Matrices [Dit-Yeung 2009]
  - Training set expansion [Yip and Gerstein 2009]

Outline

Extend linear autoregressive models

Network inference with linear models

- Estimate B in $\mathbf{x}_{t+1} = B\mathbf{x}_t + \epsilon_t$ with a **sparsity** constraint
- Threshold B to get an estimation of the true adjacency matrix A:

Operator-valued kernels based model

- Kernel-based models for vector prediction use operator-valued kernels (Micchelli and Pontil 2005, Caponnetto et al. 2008)
- Representer theorem for semi-supervised learning ($**$) give:
- $h(\mathbf{x}) = \sum_\ell K(\mathbf{x}_t, \mathbf{x}_\ell)\mathbf{c}_\ell$

Extend linear autoregressive models

Network inference with operator-valued kernels*

- $\mathbf{x}_{t+1} = h(\mathbf{x}_t) + \epsilon_t$
- $h(\mathbf{x}_t) = \sum_{i=1}^{N-1} K(\mathbf{x}_t, \mathbf{x}_\ell)\mathbf{c}_\ell$
- Use the following estimate:
  - $\hat{A}_{ij} = sgn\left(\frac{1}{N+1}\sum_{t=1}^{N+1}\frac{\partial h(x_t)_i}{\partial (x_t)_j} - \theta\right)$
- *: collaboration with Nehemy Lim and George Michailidis

Which matrix-valued kernel?

- Let us take k_{γ_1} as the (scalar) Gaussian kernel:
- $\forall \ (x, y) \in \mathbb{R} \times \mathbb{R}, k_{\gamma_1}(\mathbf{x}, \mathbf{y}) = exp(-\gamma_1 ||\mathbf{x} - \mathbf{y}||^2)$.
- and the matrix-valued kernel: $K_{\gamma_2}(\mathbf{x}, \mathbf{y})_{ij} = exp(-\gamma_2 (x^i - y^j)^2)$
- Finally, $K(\mathbf{x}, \mathbf{y}) = k_{\gamma_1}(\mathbf{x}, \mathbf{y}) B \circ K_{\gamma_2}(\mathbf{x}, \mathbf{y})$
- **Important**: Sparsity of B controls sparsity of the Jacobian

Learning h

Learning algorithm

- B is estimated as well as $\mathbf{c}'s$
- Boosting algorithm (build $H(\mathbf{x}_t) = \sum_m h_m(\mathbf{x}_t)$)
  - Base model: a model h defined on a random subspace
  - Construction through ℓ_2-boosting

Inference of a synthetic network in yeast [IRMA, Cantone, 2009]

	Switch-off		Switch-on	
	AUROC	AUPR	AUROC	AUPR
OKVAR-Boost	0.807	0.807	**1**	**1**
LASSO	0.500	0.253	0.583	0.474
Äijö	**0.875**	**0.848**	0.838	0.836

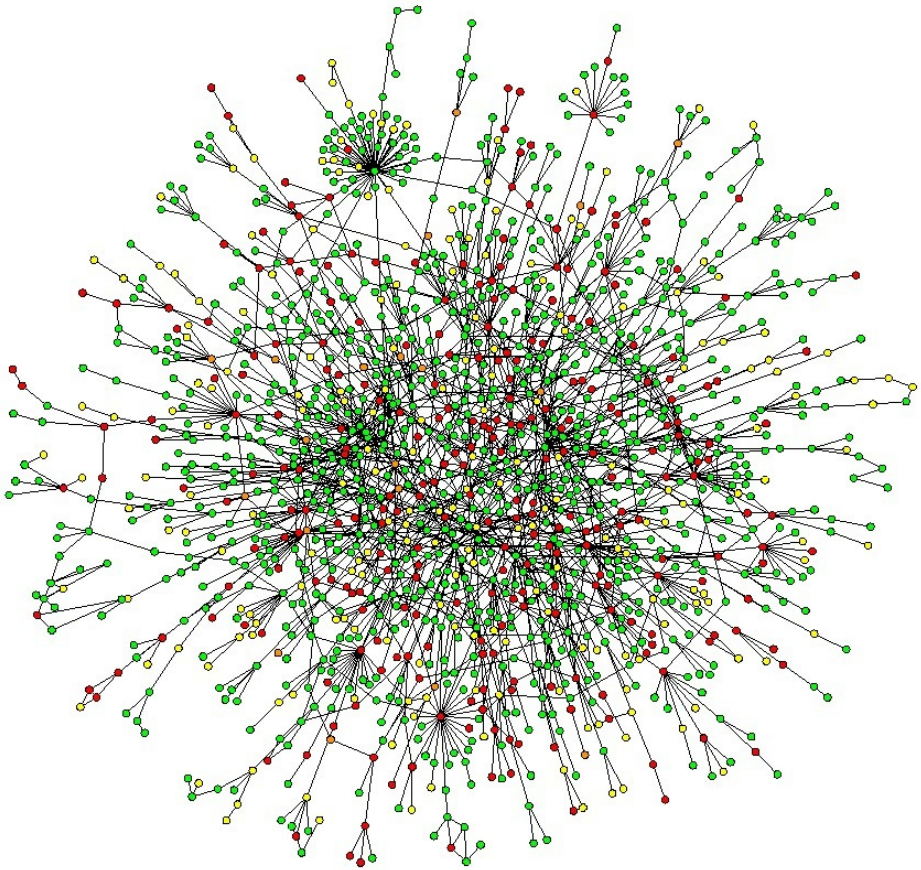
Results on 10-size networks (DREAM3 challenge)

Size-10	Ecoli1 AUROC	Ecoli1 AUPR	Ecoli2 AUROC	Ecoli2 AUPR	AURC
OKVAR + True B	0.932	0.712	0.814	0.754	0.856
OKVAR-Boost (1 TS)	0.665 $\pm$ 0.088	0.272 $\pm$ 0.081	0.629 $\pm$ 0.095	0.466 $\pm$ 0.065	0.663 $\pm$ 0.03
OKVAR-Boost (4 TS)	**0.853**	**0.583**	**0.749**	**0.536**	**0.689**
LASSO	0.500	0.119	0.547	0.531	0.528
Team 236	0.621	0.197	0.650	0.378	0.646
Team 190	0.573	0.152	0.515	0.181	0.631

Table: AUROC and AUPR for OKVAR-Boost ($\lambda_1 = 1, \lambda_2 = 10$ selected by *Block-Stability*), LASSO, Team 236 and Team 190 (DREAM3 challenge) run on DREAM3 size-10 networks. OKVAR-Boost results using respectively one time series (OKVAR-Boost (1 TS)) (Average $\pm$ Standard Deviations) and the four available time series (OKVAR-Boost (4 TS)) are from consensus networks. The numbers in **boldface** are the maximum values of each column. (* Consensus thresholds for Yeast2 and Yeast3 are different due to

Outline

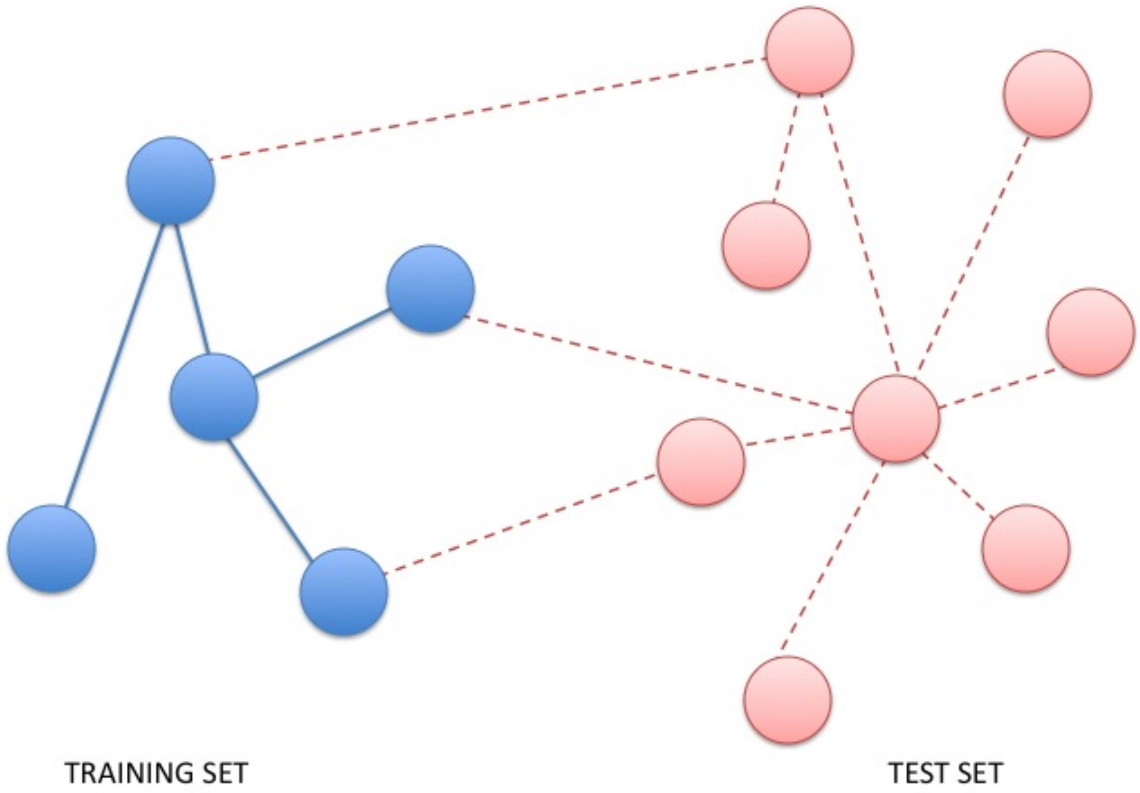
Protein-protein interaction network inference



- $\mathcal{V}$: set of nodes (corresponding to proteins)
- An edge between nodes v and v' means a physical interaction between proteins v and v'

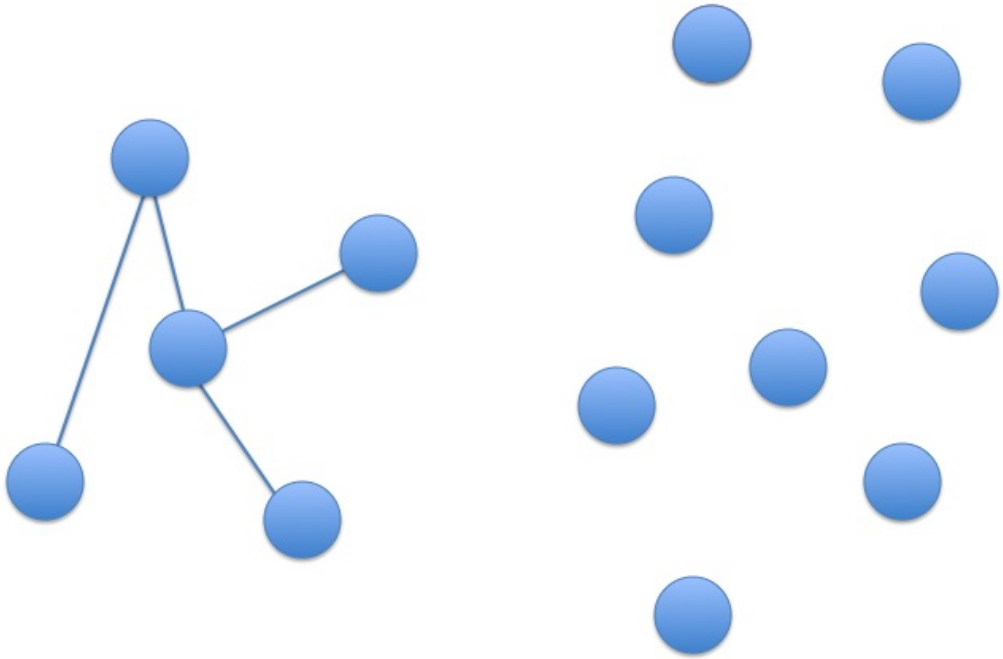
Supervised link prediction

- Edges are known for the ℓ first nodes ($\mathcal{V}_\ell$)
- **Goal**: learning a predictor $f : \mathcal{V} \times \mathcal{V} \to \{0, 1\}$ from:
  - descriptions of proteins in $\mathcal{V}_\ell$ (localization, sequence ...),
  - the adjacency matrix A_ℓ of the training subgraph.



Semi-supervised link prediction

- Let us use unlabeled data !
- Let $\mathcal{V}_{\ell+u} = \{v_1, ..., v_{\ell+u}\}$: ℓ fully labeled nodes, u unlabeled nodes
- We assume that the description of $v_{\ell+1}, ..., v_{\ell+u}$ is known
- **Goal**: learning a predictive model $f : \mathcal{V} \times \mathcal{V} \rightarrow \{0, 1\}$ from descriptions of proteins in $\mathcal{V}_{\ell+u}$ and A_ℓ, with $\ell << u$.



Building a classifier f by learning similarity κ_y

- κ_y: similarity between two proteins as nodes in the known graph,

Similarity-based model:

$$f_\theta(v, v') = sgn(\hat{\kappa_y}(v, v') - \theta)$$

- Learning a proxy of κ_y and choosing θ = learning the classifier f_θ

Reminder: scalar-valued kernel

Definition

A symmetric function k from $\mathcal{V} \times \mathcal{V}$ to $I\!R$ is said to be a definite positive kernel if and only if: For any positive integer n, for any set of n objects $(v_1, ..., v_n) \in \mathcal{V}$, for any real $c_1, ..., c_n$,

$$\sum_{i,j} c_i c_j k(v_i, v_j) \geq 0 \tag{1}$$

Theorem

For any positive definite kernel k on $\mathcal{V} \times \mathcal{V}$, there exists an Hilbert space $\mathcal{F}$ with and a feature mapping $\phi : \mathcal{V} \to \mathcal{F}$ such that for all (v, v'): $k(v, v') =< \phi(v), \phi(v') >_{\mathcal{F}}$

Now, let us introduce scalar-valued kernels (1)

Assumptions about the outputs

- Let us assume we only know for training data, the value of the $\ell \times \ell$ Gram matrix K_y of an output kernel: $(K_y)_{ij} = k_y(v_i, v_j)$, $k_y : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$
- For instance, K_y is a diffusion kernel matrix

Output Gram matrix: diffusion kernel

Only $\kappa_y(v_i, v_j) =< y(v_i), y(v_j) >_{\mathcal{F}_y}$ for $i, j = 1, \ldots, \ell$ are known.

- Here we use the **diffusion kernel** [Kondor & Lafferty, 2002)] : The Gram matrix K_{Y_ℓ} with $K_{i,j} = \kappa_y(v_i, v_j)$ is given by:

$$K_{Y_\ell} = \exp(-\beta L),$$

where the graph Laplacian L is defined by :

$$L = D_\ell - A_\ell,$$

with A_ℓ the adjacency matrix and D_ℓ the diagonal matrix of vertices degrees.

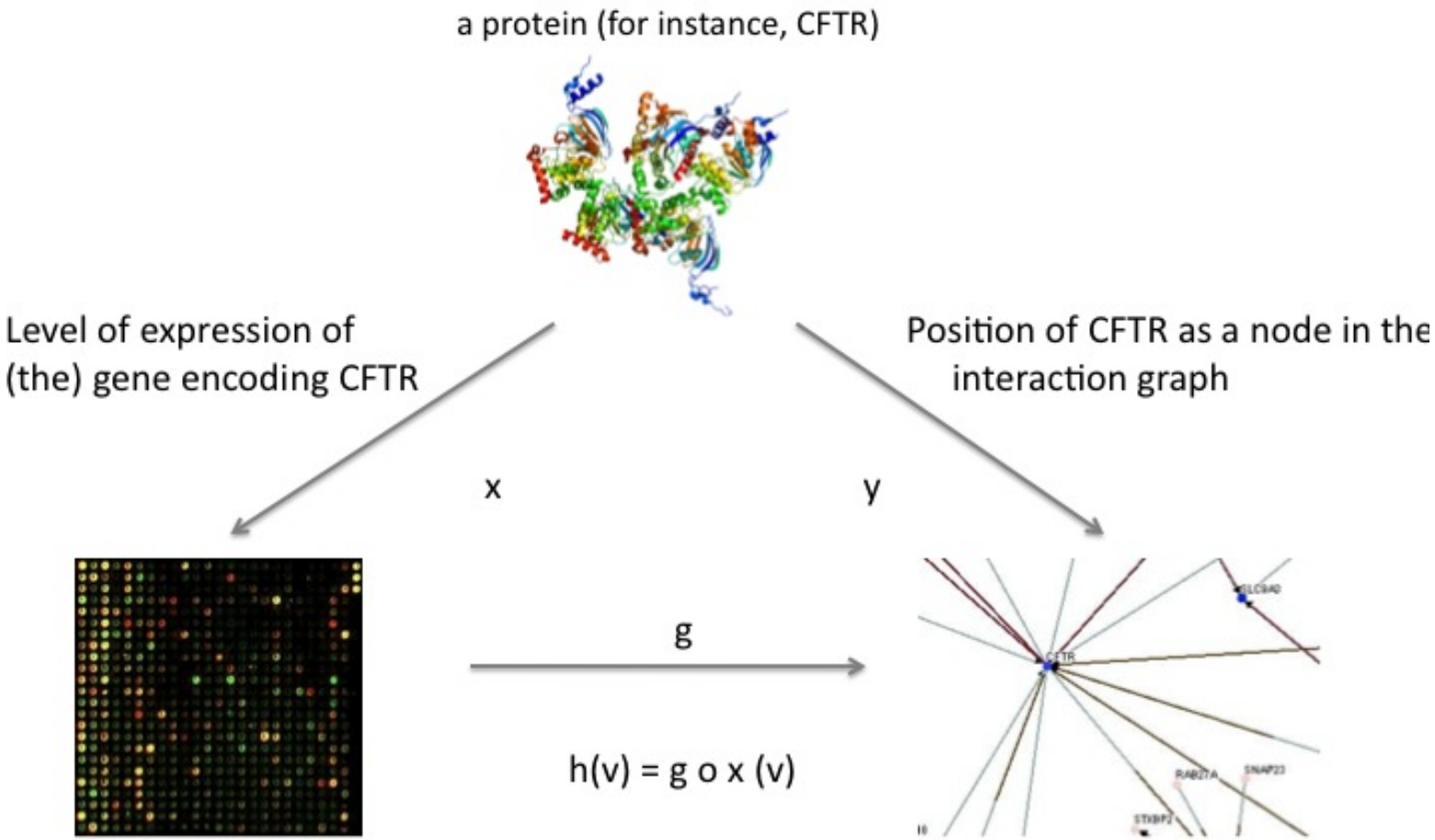
Building the classifier f by learning an output kernel k_y

$$\forall (v, v') \in \mathcal{V} \times \mathcal{V}, k_y(v, v') =< y(v), y(v') >_{\mathcal{F}_y}$$

- Let us learn to predict y with a function $h : \mathcal{V} \rightarrow \mathcal{F}_y$
- Then we will get: $\hat{k}_y(v, v') =< h(v), h(v') >_{\mathcal{F}_y}$

=> instead of learning a pairwise classifier, we learn a single variable function with output values in a Hilbert space

How to learn h ?



Which family H of models to build ?

Output kernel tree (OK3)* and extensions

- $h_{tree}(v) = \sum_{m=1}^{M} 1_m(x(v)).\bar{\mathbf{y}}_m$
- where M is the number of leaves in the tree, $1_m(x(v)) = 1$ if $x(v)$ falls into leaf m and 0 otherwise
- $\bar{\mathbf{y}}_m = \frac{1}{N_m} \sum_{i=1}^{n} 1_m(x(v_i))$

- $*$: joint work with Pierre Geurts (Geurts et al. 2006, Geurts et al. 2007) and Louis Wehenkel (Geurts et al. 2007)
- Extension to boosting and random forests

Which family H of models to build for semi-supervised learning ?

Operator-valued kernels based models$**$

- Kernel-based models for vector prediction use operator-valued kernels (Micchelli and Pontil 2005, Caponnetto et al. 2008)
- Representer theorem for semi-supervised learning ($**$) give:
  - $h(v) = \sum_{i=1}^{\ell+u} K_x(v, v_i)\mathbf{c}_i$
- $**$: joint work with Céline Brouard (PhD student) and Marie Szafranski (Brouard et al. 2011)

Operator-valued kernel-based model 1

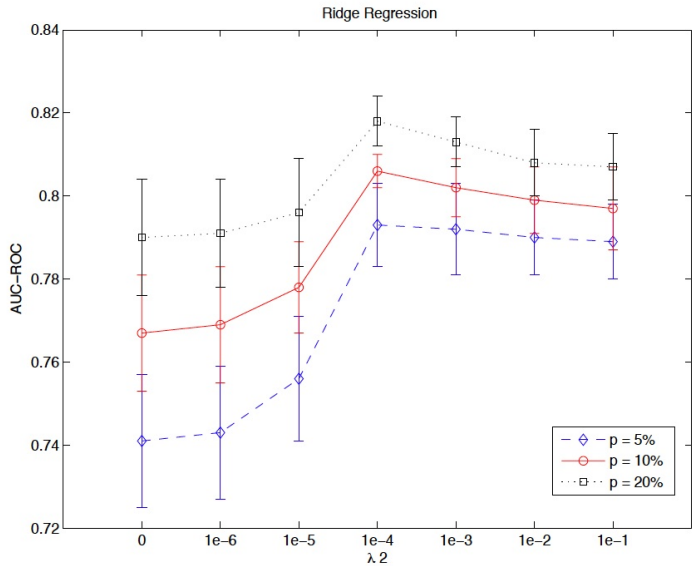
Choice of K_x

- Let us define an operator valued kernel $K_x : \mathcal{V} \rightarrow \mathcal{L}(\mathcal{F}_y)$
- $\mathcal{L}(\mathcal{F}_y)$ is the set of bounded operators on $\mathcal{F}_y$
- A simple but efficient choice of K_x is $K_x(v, v') = k_x(v, v').Id$
- With $k_x(v, v') =< x(v), x(v') >_{\mathcal{F}_x}$, $x : \mathcal{V} \rightarrow \mathcal{F}_x$

Operator-valued kernel-based model 2

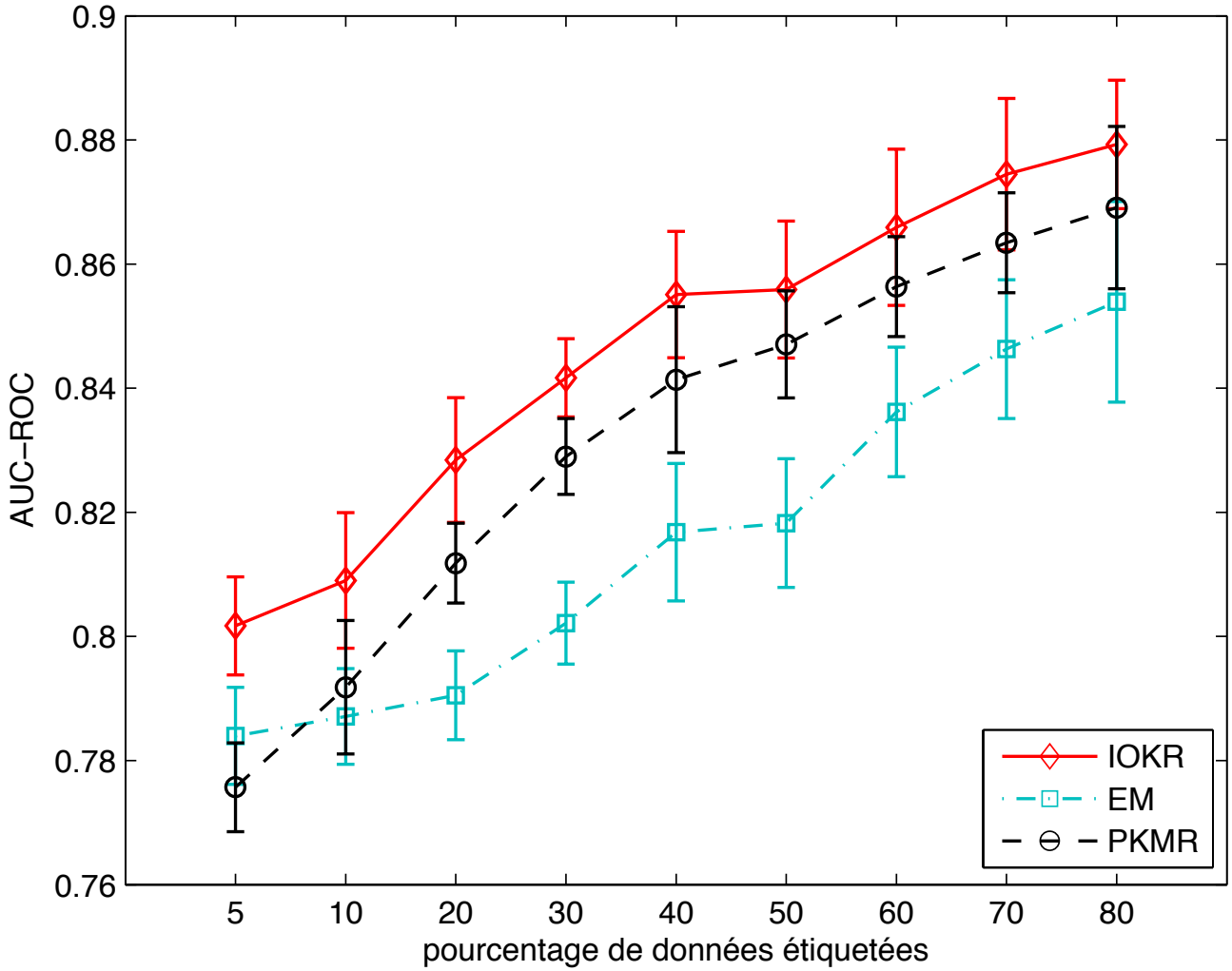
- $J(h) = \sum_{i=1}^{\ell} \|h(v_i) - y_i\|^2_{\mathcal{F}_y} + \lambda_1 \|h\|^2_{\mathcal{H}} + \lambda_2 \sum_{i,j=1}^{\ell+u} k_{x,ij} \|h(v_i) - h(v_j)\|^2_{\mathcal{F}_y}$
- Minimizing J(h) gives a closed-form solution for $\hat{h}$
- as for $\hat{k}_y$

Network inference on yeast ppinetwork, improvement brought by unlabeled data



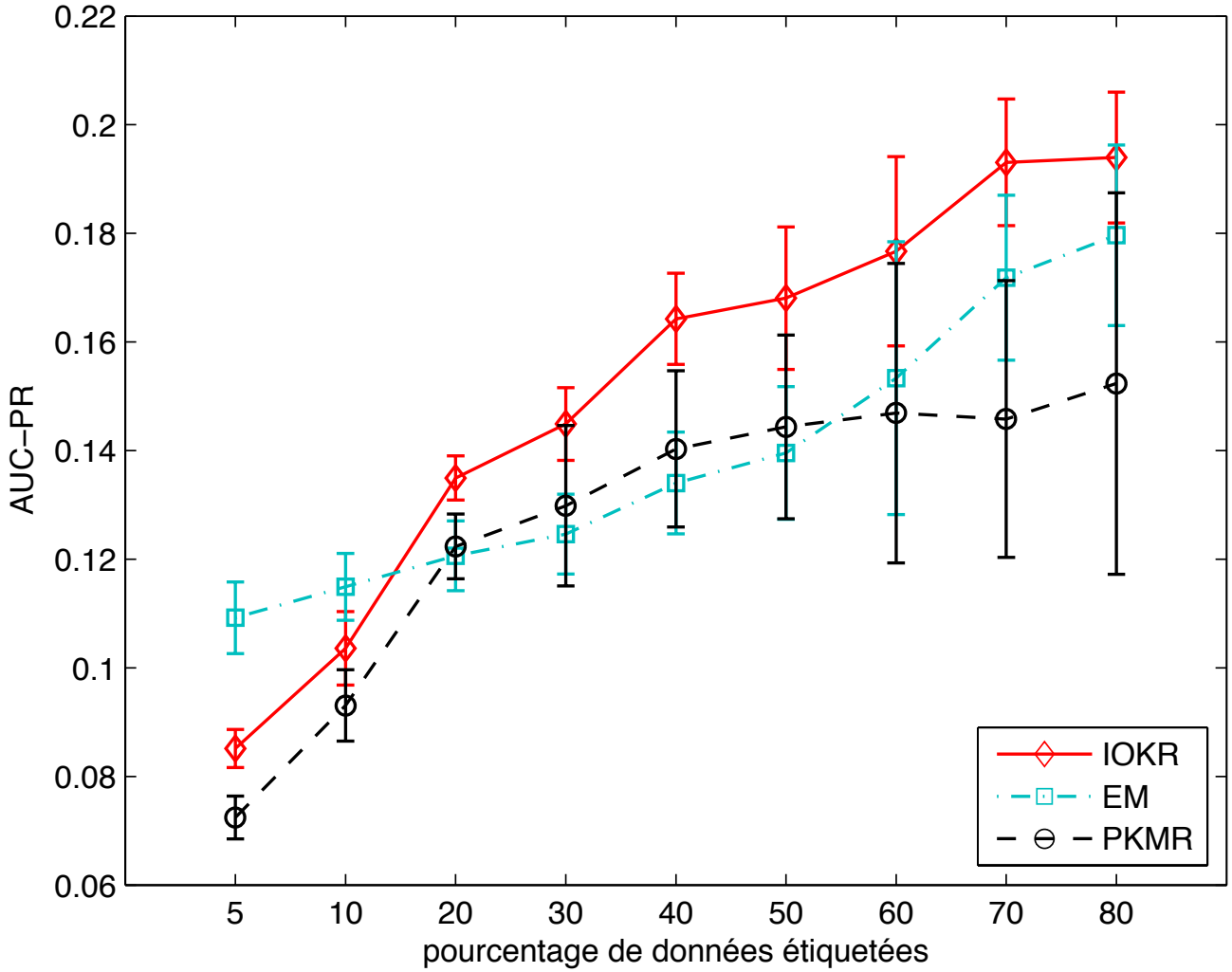
- Results of Céline Brouard
- 10 random training/test sets

Results of network inference: AUROC on yeast ppi net



- Results of Céline Brouard

Results of network inference: AUPR on yeast ppi net



- Results of Céline Brouard

Outline

Input and Output Kernel Regression (IOKR)

Take home message

Take car of the output space (choice of k_y) as well as the input space (choice of K_x)

Issues

- Kernel design, kernel learning
- Multiple kernel learning / data integration
- Model selection, scaling
- Pre-image problem in general

Remaining challenges in network inference

- Integration of various structured prior knowledge
- Scaling to a large number of genes
- Experimental design: active learning

Acknowledgements

- IBISC - CNRS - Université d'Evry, Genopole, France
  - Céline Brouard (PhD student)
  - Marie Szafranski
- INRIA-Saclay - LRI, CNRS,Orsay, France
  - Jérome Azé
  - Christine Froidevaux
- Université de Liège, GIGA, Belgium
  - Pierre Geurts
  - Louis Wehenkel
- Radbound University Nijmegen, The Netherlands
  - Adriana Birlutiu
  - Tom Heskes
- Hopital Necker, INSERM, Paris, France
  - Aleksander Edelman
  - Chiara Guerrera