Biological network inference: a challenge to structured data-mining

Florence d’Alché-Buc¹,²

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

1. Introduction
   - Gene regulatory networks
   - Protein-protein interaction network

2. Gene regulatory network inference with operator-valued kernels

3. Protein-protein interaction prediction with operator-valued kernels

4. Conclusion
Networks in molecular biology

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

(EGC 2013)
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...
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 medicine, ...

**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

Mathematical model of dynamics (ODE, Markov models)

Estimation algorithm

Objective function including constraints (prior knowledge, qualitative constraints)

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] ≠ 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**
  \[ x_{t+1} = h(x_t) + \epsilon_t, \quad t > 0, \quad x_t: \text{state vector}, \quad h \in \mathcal{H}, \quad \epsilon_t: \text{iid gaussian noise} \]
- **Dynamic Bayesian models**
  \[ x_{t+1}^i = h_i(Pa(i,t)) + \epsilon_t, \quad t > 0, \quad Pa(i,t): \text{parent state variables at time } t \]
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**
  - Gaussian processes for network inference (Aijo and Lahdesmaki 2009)
Outline

1. Introduction
   - Gene regulatory networks
   - Protein-protein interaction network

2. Gene regulatory network inference with operator-valued kernels

3. Protein-protein interaction prediction with operator-valued kernels

4. Conclusion
Protein-protein interaction network

Protein-protein interaction network in yeast

(EGC 2013)
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
- 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

1. Introduction
   - Gene regulatory networks
   - Protein-protein interaction network

2. Gene regulatory network inference with operator-valued kernels

3. Protein-protein interaction prediction with operator-valued kernels

4. Conclusion
### Extend linear autoregressive models

Network inference with linear models

- Estimate $B$ in $x_{t+1} = Bx_t + \epsilon_t$ with a **sparsity** constraint.
- Threshold $B$ to get an estimation of the true adjacency matrix $A$.

---

### Operator-valued kernels based model

- Representer theorem for semi-supervised learning (***) give:
  \[ h(x) = \sum_\ell K(x_t, x_\ell)c_\ell \]
Extend linear autoregressive models

Network inference with operator-valued kernels

\[ x_{t+1} = h(x_t) + \epsilon_t \]
\[ h(x_t) = \sum_{i=1}^{N-1} K(x_t, x_i) c_i \]

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

Learning algorithm

- B is estimated as well as c’s
- Boosting algorithm (build $H(x_t) = \sum_m h_m(x_t)$)
  - Base model: a model h defined on a random subspace
  - Construction through $\ell_2$-boosting
Inference of a synthetic network in yeast [IRMA, Cantone, 2009]

<table>
<thead>
<tr>
<th>Method</th>
<th>Switch-off AUROC</th>
<th>Switch-on AUROC</th>
<th>Switch-off AUPR</th>
<th>Switch-on AUPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>OKVAR-Boost</td>
<td>0.807</td>
<td>1</td>
<td>0.807</td>
<td>1</td>
</tr>
<tr>
<td>LASSO</td>
<td>0.500</td>
<td>0.583</td>
<td>0.253</td>
<td>0.474</td>
</tr>
<tr>
<td>Äijö</td>
<td>0.875</td>
<td>0.838</td>
<td>0.848</td>
<td>0.836</td>
</tr>
</tbody>
</table>
### Results on 10-size networks (DREAM3 challenge)

<table>
<thead>
<tr>
<th>Size-10</th>
<th>Ecoli1</th>
<th>Ecoli2</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>AUROC</td>
<td>AUPR</td>
<td>AUROC</td>
<td>AUPR</td>
</tr>
<tr>
<td>OKVAR + True B</td>
<td>0.932</td>
<td>0.712</td>
<td>0.814</td>
<td>0.754</td>
</tr>
<tr>
<td>OKVAR-Boost (1 TS)</td>
<td>0.665</td>
<td>0.272</td>
<td>0.629</td>
<td>0.466</td>
</tr>
<tr>
<td></td>
<td>± 0.088</td>
<td>± 0.081</td>
<td>± 0.095</td>
<td>± 0.065</td>
</tr>
<tr>
<td>OKVAR-Boost (4 TS)</td>
<td>0.853</td>
<td>0.583</td>
<td>0.749</td>
<td>0.536</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>± 0.039</td>
<td></td>
</tr>
<tr>
<td>LASSO</td>
<td>0.500</td>
<td>0.119</td>
<td>0.547</td>
<td>0.531</td>
</tr>
<tr>
<td>Team 236</td>
<td>0.621</td>
<td>0.197</td>
<td>0.650</td>
<td>0.378</td>
</tr>
<tr>
<td>Team 190</td>
<td>0.573</td>
<td>0.152</td>
<td>0.515</td>
<td>0.181</td>
</tr>
</tbody>
</table>

**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 ± 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 **boldface**. )
Outline

1. Introduction
   - Gene regulatory networks
   - Protein-protein interaction network

2. Gene regulatory network inference with operator-valued kernels

3. Protein-protein interaction prediction with operator-valued kernels

4. Conclusion
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 $\ell$ first nodes ($V_\ell$)
- **Goal**: learning a predictor $f : V \times V \rightarrow \{0, 1\}$ from:
  - descriptions of proteins in $V_\ell$ (localization, sequence ...),
  - the adjacency matrix $A_\ell$ of the training subgraph.

![Diagram showing training and test sets with known edges](image-url)
Semi-supervised link prediction

- Let us use unlabeled data!
- Let \(\mathcal{V}_{\ell+u} = \{v_1, \ldots, v_{\ell+u}\} : \ell\) fully labeled nodes, \(u\) unlabeled nodes
- We assume that the description of \(v_{\ell+1}, \ldots, v_{\ell+u}\) 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 \(A_\ell\), with \(\ell << u\).
Building a classifier $f$ by learning similarity $\kappa_y$

- $\kappa_y$: similarity between two proteins as nodes in the known graph,

**Similarity-based model:**

$$f_\theta(v, v') = \text{sgn}(\hat{\kappa}_y(v, v') - \theta)$$

- Learning a proxy of $\kappa_y$ and choosing $\theta = \text{learning the classifier } f_\theta$
Reminder: scalar-valued kernel

**Definition**

A symmetric function $k$ from $\mathcal{V} \times \mathcal{V}$ to $IR$ 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 $c_1, ..., c_n$,

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

(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 $\phi : \mathcal{V} \rightarrow \mathcal{F}$ such that for all $(v, v')$: $k(v, v') = \langle \phi(v), \phi(v') \rangle_{\mathcal{F}}$
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: $(K_y)_{ij} = k_y(v_i, v_j)$, $k_y : \mathcal{V} \times \mathcal{V} \rightarrow \mathbb{R}$
- For instance, $K_y$ is a diffusion kernel matrix
Output Gram matrix: diffusion kernel

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

Here we use the **diffusion kernel** [Kondor & Lafferty, 2002]:

The Gram matrix $K_{Y,\ell}$ 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 $A_\ell$ the adjacency matrix and $D_\ell$ the diagonal matrix of vertices degrees.
Building the classifier $f$ by learning an output kernel $k_y$

$$\forall (v, v') \in \mathcal{V} \times \mathcal{V}, k_y(v, v') = \langle y(v), y(v') \rangle_{\mathcal{F}_y}$$

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

$\Rightarrow$ instead of learning a pairwise classifier, we learn a single variable function with output values in a Hilbert space
How to learn \( h \)?

A protein (for instance, CFTR)

Level of expression of (the) gene encoding CFTR

Position of CFTR as a node in the interaction graph

\[ x \quad y \]

\[ g \]

\[ h(v) = g \circ x(v) \]
Which family $H$ of models to build?

Output kernel tree (OK3)* and extensions

- $h_{\text{tree}}(v) = \sum_{m=1}^{M} 1_m(x(v)).\bar{y}_m$
- 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
- $\bar{y}_m = \frac{1}{N_m} \sum_{i=1}^{n} 1_m(x(v_i))$

*: 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:
  \[ h(v) = \sum_{i=1}^{\ell+u} K(x, v_i)c_i \]
- **: joint work with Céline Brouard (PhD student) and Marie Szafranski (Brouard et al. 2011)
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 $k_x(v, v') = \langle x(v), x(v') \rangle_{\mathcal{F}_x}$, $x : \mathcal{V} \rightarrow \mathcal{F}_x$
$J(h) = \sum_{i=1}^{\ell} \| h(v_i) - y_i \|_{\mathcal{F}_y}^2 + \lambda_1 \| h \|_{\mathcal{H}}^2 + \lambda_2 \sum_{i,j=1}^{\ell+u} k_{x,ij} \| h(v_i) - h(v_j) \|_{\mathcal{F}_y}^2$

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

(EGC 2013)
Outline

1. Introduction
   - Gene regulatory networks
   - Protein-protein interaction network

2. Gene regulatory network inference with operator-valued kernels

3. Protein-protein interaction prediction with operator-valued kernels

4. Conclusion
**Input and Output Kernel Regression (IOKR)**

---

**Take home message**

Take care 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