Supervised inference of biological networks
and
Classification of gene expression data with gene networks

Jean-Philippe Vert
Jean-Philippe.Vert@ensmp.fr

Centre for Computational Biology
Ecole des Mines de Paris, ParisTech

Institute for Infocomm Research, Singapore, February 14th, 2007
Outline

1. Supervised inference of biological networks from heterogeneous genomic data

2. Using gene networks for gene expression data classification
Outline

1. Supervised inference of biological networks from heterogeneous genomic data

2. Using gene networks for gene expression data classification
Outline

1. Supervised inference of biological networks from heterogeneous genomic data

2. Using gene networks for gene expression data classification
Motivation

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

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

J.-P. Vert (Ecole des Mines)
Strategies

Unsupervised approaches

The graph is completely unknown

- **model-based** approaches: Bayes nets, dynamical systems,
- **similarity-based**: connect similar nodes

Supervised approaches

Part of the graph is known in advance

- Prior knowledge in model-based approaches
- **Statistical / Machine learning** approaches: learn from the known subnetwork a rule that can predict edges from genomic data
Strategies

Unsupervised approaches
The graph is completely unknown
- model-based approaches: Bayes nets, dynamical systems,..
- similarity-based: connect similar nodes

Supervised approaches
Part of the graph is known in advance
- Prior knowledge in model-based approaches
- Statistical / Machine learning approaches: learn from the known subnetwork a rule that can predict edges from genomic data
Data representation a distances

- We assume that each type of data (expression, sequences...) defines a *negative definite* distance between genes.
- Many such distances exist (cf kernel methods).
- Data integration is easily obtained by summing the distance to obtain an “integrated” distance.
Method 1: Direct similarity-based prediction

- **Motivation:** “connect similar genes”
- Connect $a$ and $b$ if $d(a, b)$ is below a threshold.
- This is an **unsupervised approach** (no use of the known subnetwork).
Motivation: “connect similar genes”
Connect \(a\) and \(b\) if \(d(a, b)\) is below a threshold.
This is an unsupervised approach (no use of the known subnetwork).
Method 2: metric learning

Metric learning

- Motivation: use the known subnetwork to refine the distance measure, before applying the similarity-based method
- Based on kernel CCA (Yamanishi et al., 2004) or kernel metric learning (V. and Yamanishi, 2005).
Metric learning

- Motivation: use the known subnetwork to refine the distance measure, before applying the similarity-based method.
- Based on kernel CCA (Yamanishi et al., 2004) or kernel metric learning (V. and Yamanishi, 2005).
Method 2: metric learning

Metric learning

- Motivation: use the known subnetwork to refine the distance measure, before applying the similarity-based method.
- Based on kernel CCA (Yamanishi et al., 2004) or kernel metric learning (V. and Yamanishi, 2005).
Method 2: metric learning

**Metric learning**

- Motivation: use the **known subnetwork** to refine the distance measure, before applying the similarity-based method.
- Based on **kernel CCA** (Yamanishi et al., 2004) or **kernel metric learning** (V. and Yamanishi, 2005).
Metric learning example

Kernel metric learning (V. and Yamanishi, 2005)

- **Criterion**: connected points should be near each other after mapping to a new $d$-dimensional Euclidean space.
- Add **regularization** to deal with high dimensions.
- Mapping $f(x) = (f_1(x), \ldots, f_d(x))$ with:
  
  $$f_i = \arg\min_{f \perp \{f_1, \ldots, f_{i-1}\}, \text{var}(f)=1} \left\{ \sum_{i \sim j} (f(x_i) - f(x_j))^2 + \lambda \|f\|_k^2 \right\}.$$  

- Interpolates between (kernel) PCA ($\lambda = \infty$) and graph embedding ($\lambda = 0$).
- Equivalent to a generalized eigenvalue problem.
Metric learning example

Kernel CCA (Yamanishi et al., 2004)

- **Criterion**: Find a subspace where the graph distance and the genomic data distance match
- Formulated as a search for correlated directions (kernel trick).

![Graph and data visualization]
Metric learning example

Kernel CCA (Yamanishi et al., 2004)

- **Criterion**: Find a subspace where the graph distance and the genomic data distance match
- Formulated as a search for correlated directions (kernel trick).

![Diagram showing a network and kernel representation](image-url)
Metric learning example

Kernel CCA (Yamanishi et al., 2004)

- **Criterion**: Find a subspace where the graph distance and the genomic data distance match
- Formulated as a search for correlated directions (kernel trick).

Diffusion kernel

Kernel
Metric learning example

**Kernel CCA (Yamanishi et al., 2004)**

- **Criterion**: Find a subspace where the graph distance and the genomic data distance match
- Formulated as a search for correlated directions (kernel trick).

![Diagram of Kernel CCA](image)
Method 3: Matrix completion

- Motivation: Fill missing entries in the adjacency matrix directly, by making it similar to (a variant of) the data matrix
- Method: EM algorithm based on information geometry of positive semidefinite matrices (Kato et al., 2005)
A pair can be connected (1) or not connected (-1)

Use known network as a training set for a SVM that will predict if new pair is connected or not

Example: SVM with tensor product pairwise kernel (Ben-Hur and Noble, 2006):

\[ K_{TTPK} ((x_1, x_2), (x_3, x_4)) = K(x_1, x_3)K(x_2, x_4) + K(x_1, x_4)K(x_2, x_3). \]
Method 5: Local predictions

- 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.
Method 5: Local predictions

- 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.
## Local predictions: pros and cons

### Pros
- Allow **very different models** for nearby nodes on the graph
- Faster to train $n$ models with $n$ examples than 1 model with $n^2$ examples

### Cons
- Few positive examples available for some nodes
Local predictions: pros and cons

Pros
- Allow very different models for nearby nodes on the graph
- Faster to train $n$ models with $n$ examples than 1 model with $n^2$ examples

Cons
- Few positive examples available for some nodes
## Experiments

### Network
- Metabolic network (668 vertices, 2782 edges)
- Protein-protein interaction network (984 vertices, 2438 edges)

### Data (yeast)
- Gene expression (157 experiments)
- Phylogenetic profile (145 organisms)
- Cellular localization (23 intracellular locations)
- Yeast two-hybrid data (2438 interactions among 984 proteins)

### Method
- 5-fold cross-validation
- Predict edges between test set and training set
Results: protein-protein interaction
Results: metabolic gene network

- **Graphs**:
  - Ratio of true positives vs. False discovery rate
  - Ratio of false positives vs. Ratio of true positives

- **Lines**:
  - Direct
  - kML
  - kCCA
  - em
  - local
  - Pkernel

- **Legend**:
  - Blue: Direct
  - Green: kML
  - Red: kCCA
  - Cyan: em
  - Black: local
  - Magenta: Pkernel
Local SVM, protein-protein interaction network.
Results: effect of data integration

Local SVM, metabolic gene network.
A variety of methods have been investigated recently. Some reach interesting performance on the benchmarks: Local SVM retrieve 45% of all true edges of the metabolic gene network at a FDR below 50%. Valid for any network, but non-mechanistic model. Future work: experimental validation, improved data integration, semi-local approaches...
1. Supervised inference of biological networks from heterogeneous genomic data

2. Using gene networks for gene expression data classification
Tumor classification from microarray data

Data available
- Gene expression measures for more than 10\(k\) genes
- Measured on less than 100 samples of two (or more) different classes (e.g., different tumors)

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

Data available

- Gene expression measures for more than 10k genes
- Measured on less than 100 samples of two (or more) different classes (e.g., different tumors)

Goal

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

The approach

- Each sample is represented by a vector \( x = (x_1, \ldots, x_p) \) where \( p > 10^5 \) is the number of probes.
- **Classification**: given the set of labeled sample, learn a linear decision function:

\[
f(x) = \sum_{i=1}^{p} \beta_i x_i + \beta_0,
\]

that is positive for one class, negative for the other.
- **Interpretation**: the weight \( \beta_i \) quantifies the influence of gene \( i \) for the classification.
Linear classifiers

Pitfalls

- No robust estimation procedure exist for 100 samples in $10^5$ dimensions!
- It is necessary to reduce the complexity of the problem with prior knowledge.
The approach

A common method in statistics to learn with few samples in high dimension is to constrain the norm of $\beta$, e.g.:

- Euclidean norm (support vector machines, ridge regression):
  \[ \| \beta \|_2 = \sum_{i=1}^{p} \beta_i^2 \]
- $L_1$-norm (lasso regression): \[ \| \beta \|_1 = \sum_{i=1}^{p} |\beta_i| \]

Pros

- Good performance in classification

Cons

- Limited interpretation (small weights)
- No prior biological knowledge
### Example 2: Feature Selection

#### The approach
Constrain most weights to be 0, i.e., select a few genes ($< 20$) whose expression are enough for classification. Interpretation is then about the selected genes.

#### Pros
- Good performance in classification
- Useful for **biomarker** selection
- Apparently easy interpretation

#### Cons
- The gene selection process is usually **not robust**
- Wrong interpretation is the rule (too much correlation between genes)
Basic biological functions are usually expressed in terms of pathways and not of single genes (metabolic, signaling, regulatory).

Many pathways are already known.

How to use this prior knowledge to constrain the weights to have an interpretation at the level of pathways?
Pathway-derived norm constraint

One solution (Rapaport et al., 2007)

- Let the set of pathways be represented by an undirected graph.
- Consider the pathway-derived norm:
  \[ \Omega(\beta) = \sum_{i \sim j} (\beta_i - \beta_j)^2 . \]

- Constrain \( \Omega(\beta) \) instead of \( \| \beta \|_2^2 \)
- Remark: this is equivalent to a SVM with a particular kernel.
Pathway interpretation

Bad example

- The graph is the complete known metabolic network of the budding yeast (from KEGG database)
- We project the classifier weight learned by a SVM
- Good classification accuracy, but no possible interpretation!
Pathway interpretation

Good example

- The graph is the complete known metabolic network of the budding yeast (from KEGG database)
- We project the classifier weight learned by a spectral SVM
- Good classification accuracy, and good interpretation!

J.-P. Vert (École des Mines)
Use the gene graph to encode prior knowledge about the classifier.

Prior knowledge is always needed to classify few examples in large dimensions (sometimes implicitly)

Future work: validation of the method on more data, other formulations, directed graphs...
## Acknowledgements

### Supervised graph inference
- Yoshihiro Yamanishi, Minoru Kanehisa (Univ. Kyoto): kCCA, kML
- Kevin Bleakley, Gerard Biau (Univ. Montpellier): local SVM

### Classification of microarray data
- Franck Rapaport, Emmanuel Barillot, Andrei Zynoviev, Marie Dutreix (Curie Institute)