Lecture Thu 1.12.

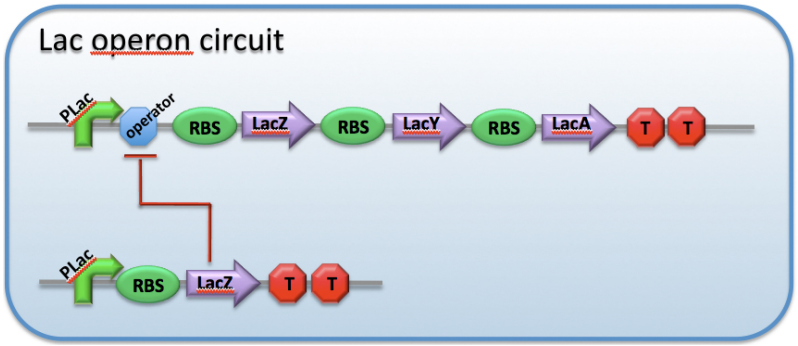
BIOLOGICAL NETWORK INFERENCE

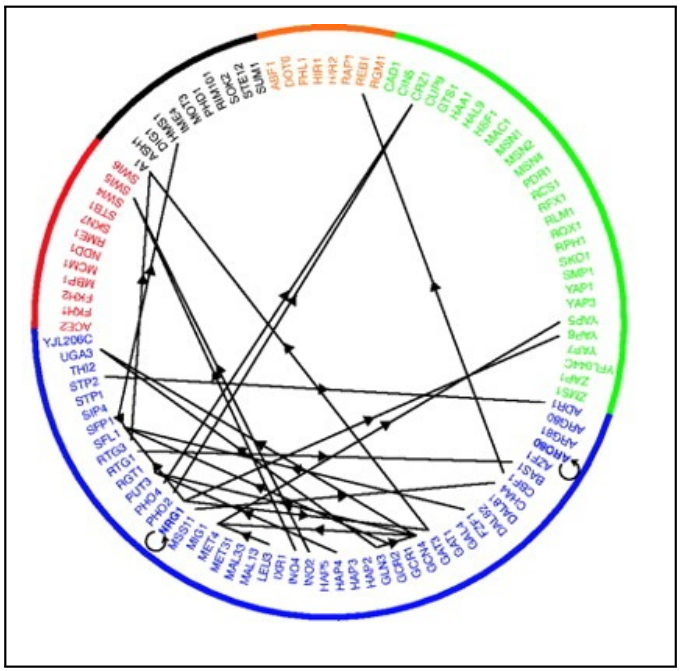
Networks in biology

- Network is a useful formalism for many biological phenomena
- Example networks:
  - Transcription regulation networks
  - Protein-protein interaction networks
  - Metabolic networks
  - Signal transduction networks
- Here we focus on prediction of interactions in the network
- The course 582653 Computational Methods of Systems Biology looks network analysis more in depth

Transcriptional regulation networks

- Describe the relationships between genes encoding regulatory proteins (Transcription factors) and the genes they regulate by binding to promoter region (top figure).
- Network nodes correspond to genes (below figure)
- Edges $A \Longrightarrow B$ correspond to regulatory relations 'product of gene A controls the transcription of gene B'
- Positive (enhancer) or negative (repressor) regulation may be indicated by signs or special arrowheads

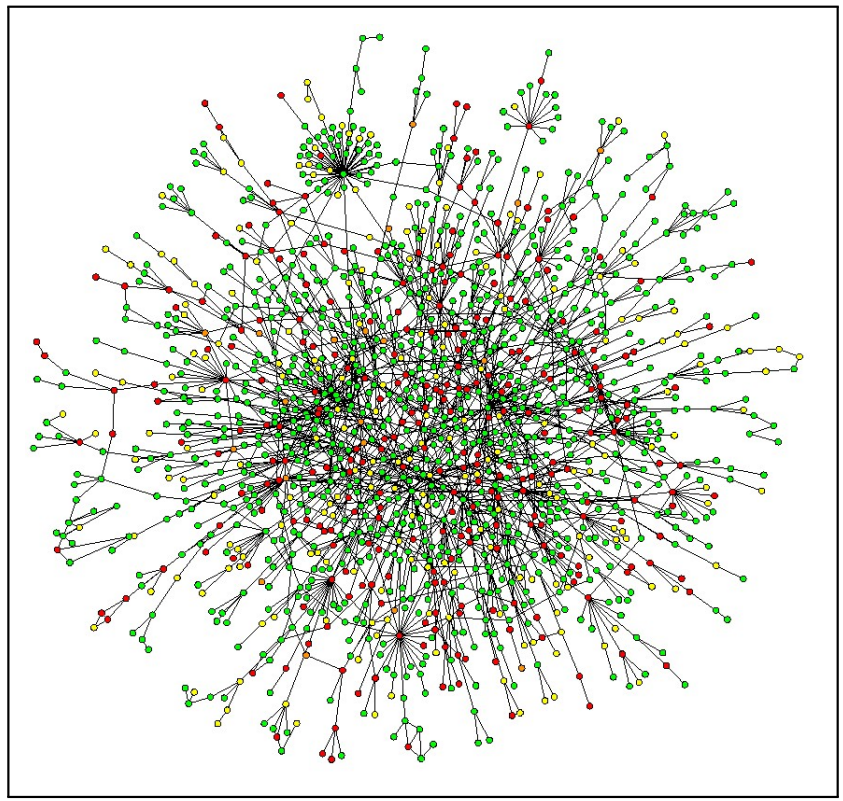




(Lee et al., Science Vol. 298, 2002)

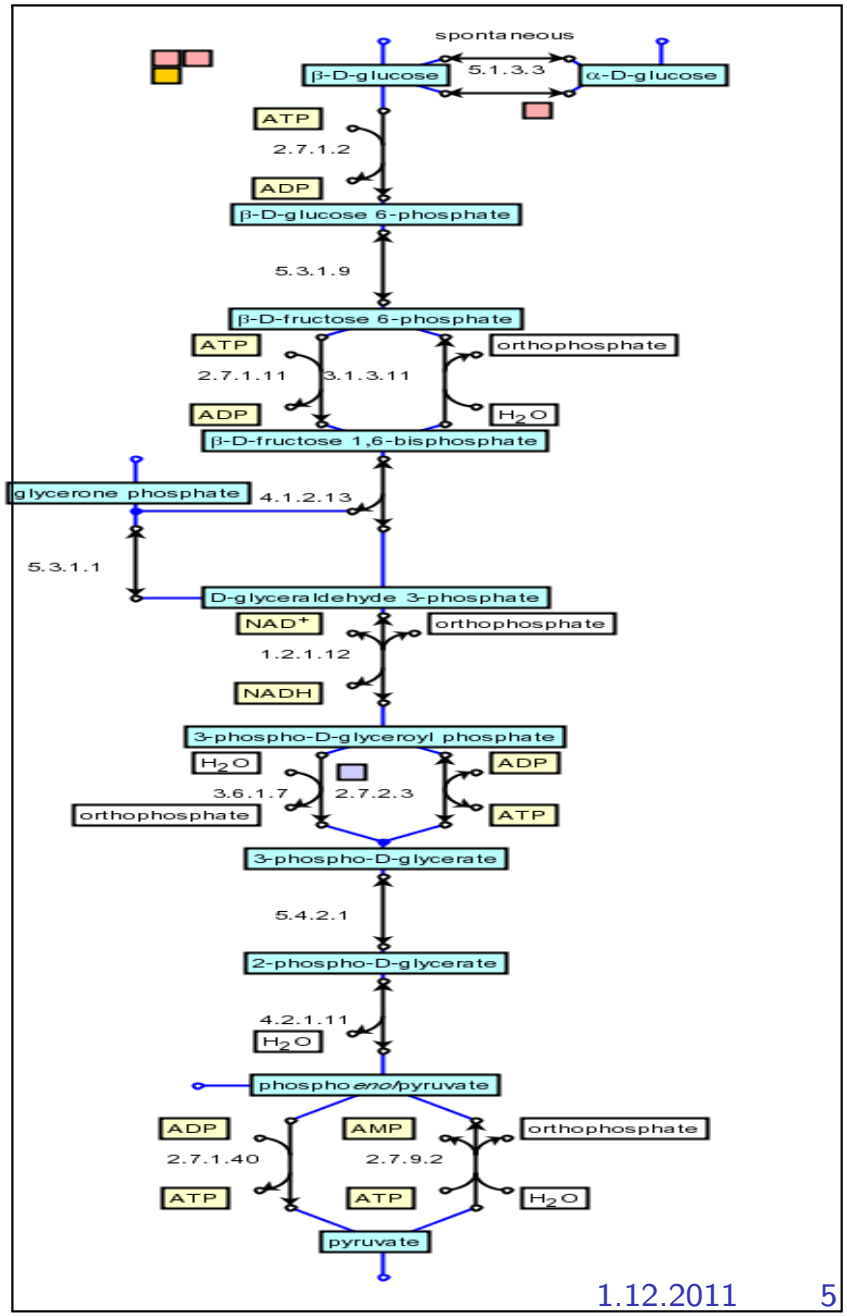
Protein interaction networks (PPI)

- Protein(-protein) interaction (PPI) network models two kinds of interactions:
  - Proteins binding to each other and functioning as a complex
  - Proteins catalyzing biochemical reactions sharing a metabolite (enzyme network)
- Represented as undirected networks with proteins as nodes and the interactions as edges



Metabolic networks

- Metabolism is responsible of providing the cell with energy and building blocks for cell growth
- Metabolic networks are composed of biochemical reactions, catalyzed by enzymes (proteins) and metabolites that participate in the reactions



Representing metabolic networks as graphs

For structural analysis of metabolic networks, the most frequently encountered representations are:

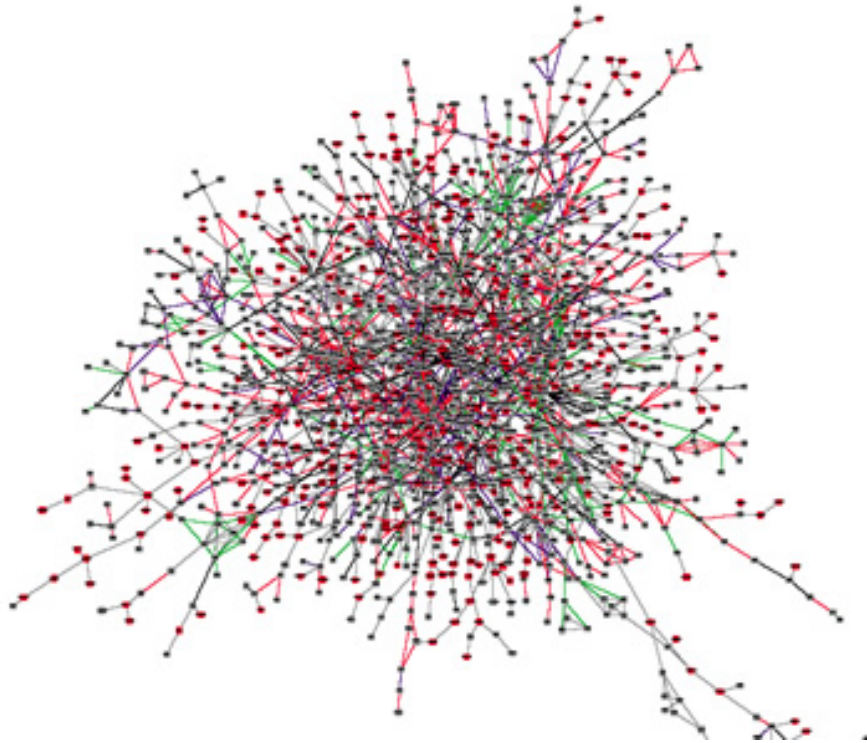
- **Enzyme interaction network**
- Reaction graph
- Substrate graph (also called metabolite graph)

More detailed representations:

- Bi-partite graph: both reactions and metabolites as nodes
- Atom-level representations
- Boolean circuits (AND-OR graphs)

Enzyme interaction networks

- Enzymes as nodes
- Link between two enzymes if they catalyze reactions that have common metabolites
- A special kind of protein-protein interaction network



Interactions of the first kind: Physical interactions

Physical interactions (typically: binding of molecules, forming a complex) between molecules:

- Protein and DNA: transcription factor proteins, epigenetic silencers, histones ...
- Protein and RNA: ribosomes, transcriptase proteins, ...
- Protein and Protein: protein complexes, (some) metabolic pathways, ...
- Protein and Small molecule: enzymes, metabolic regulation, signaling, ...

Interactions of the second kind: Abstract interactions

We will often look at abstract or logical interactions between the components, rather than mapping physical interactions:

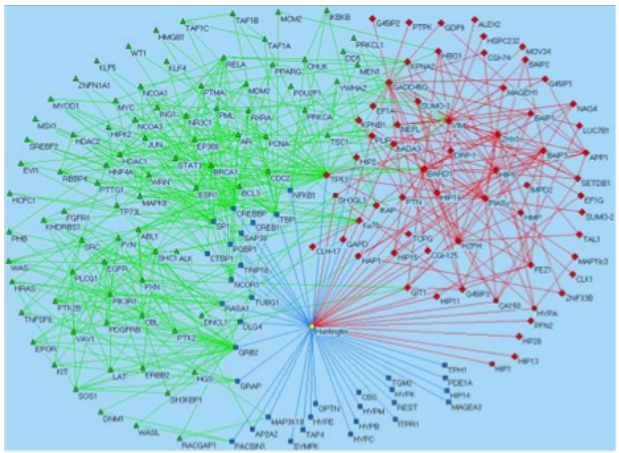
- Gene regulatory network: 'gene A' negatively regulates 'gene B'
  - Biologically: transcription factor protein produced by A, binds to the promotor region of B, thus repressing the transcription of B
- Enzyme interaction network: enzyme E_1 interacts with enzyme E_2
  - Biologically: both enzymes catalyze biochemical reactions that involve metabolite molecule M (e.g. puryvate)
- Correlated behavior: gene A has similar/dependent behaviour to gene B in a set of experiments—do not necessarily need to have direct regulatory relationship, although often they have

Supervised inference of biological networks

- We will review a machine learning method for inferring missing edges in biological networks.
- Source: Jean-Philippe Vert: Reconstruction of biological networks by supervised machine learning approaches. In Huma M. Lodhi, Stephen H. Muggleton: Elements of Computational Systems Biology, Wiley, 2010, pp. 165-186

Graph reconstruction as a pattern recognition problem

- Assume a set of nodes $V = \{v_1, \ldots, v_n\}$ corresponding to the biological entity of interest (here: genes or proteins)
- Each node has an associated feature vector $\phi(v)$ describing the node, composed of different data sources available for the node
- We wish to reconstruct a set of edges $E \subset V \times V$ that define the biological network



Data sources for interaction prediction

Indirect data for learning interations:

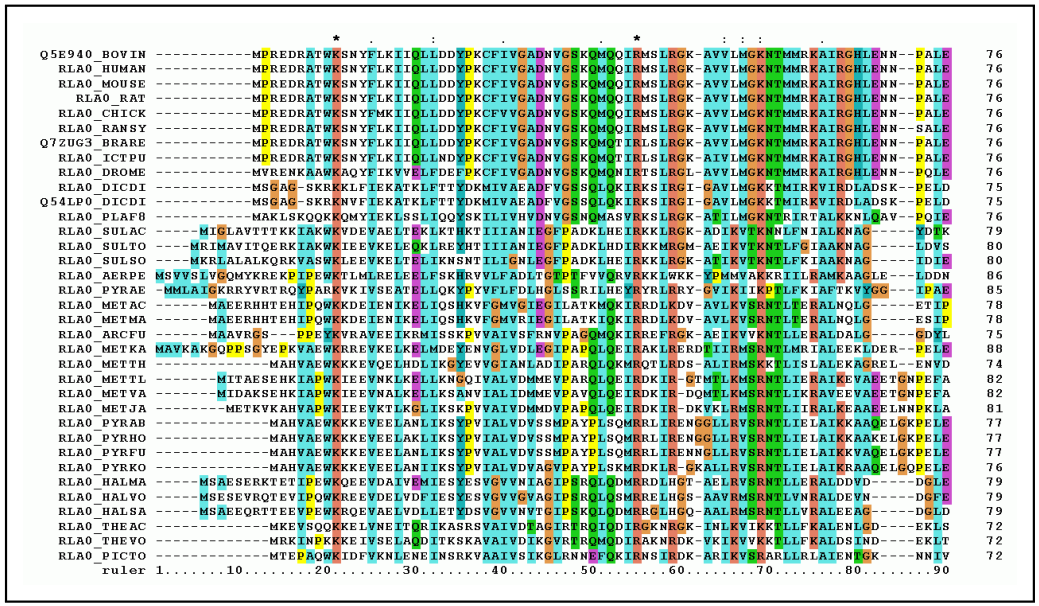
- Sequence information
- Gene co-expression
- Phylogenetic profiling
- Sub-cellular localization

Direct interaction data

- Yeast-two-hydrid - direct measurement of PPIs
- ChIP-seq/ChIP-chip - direct measurement of Protein-DNA binding

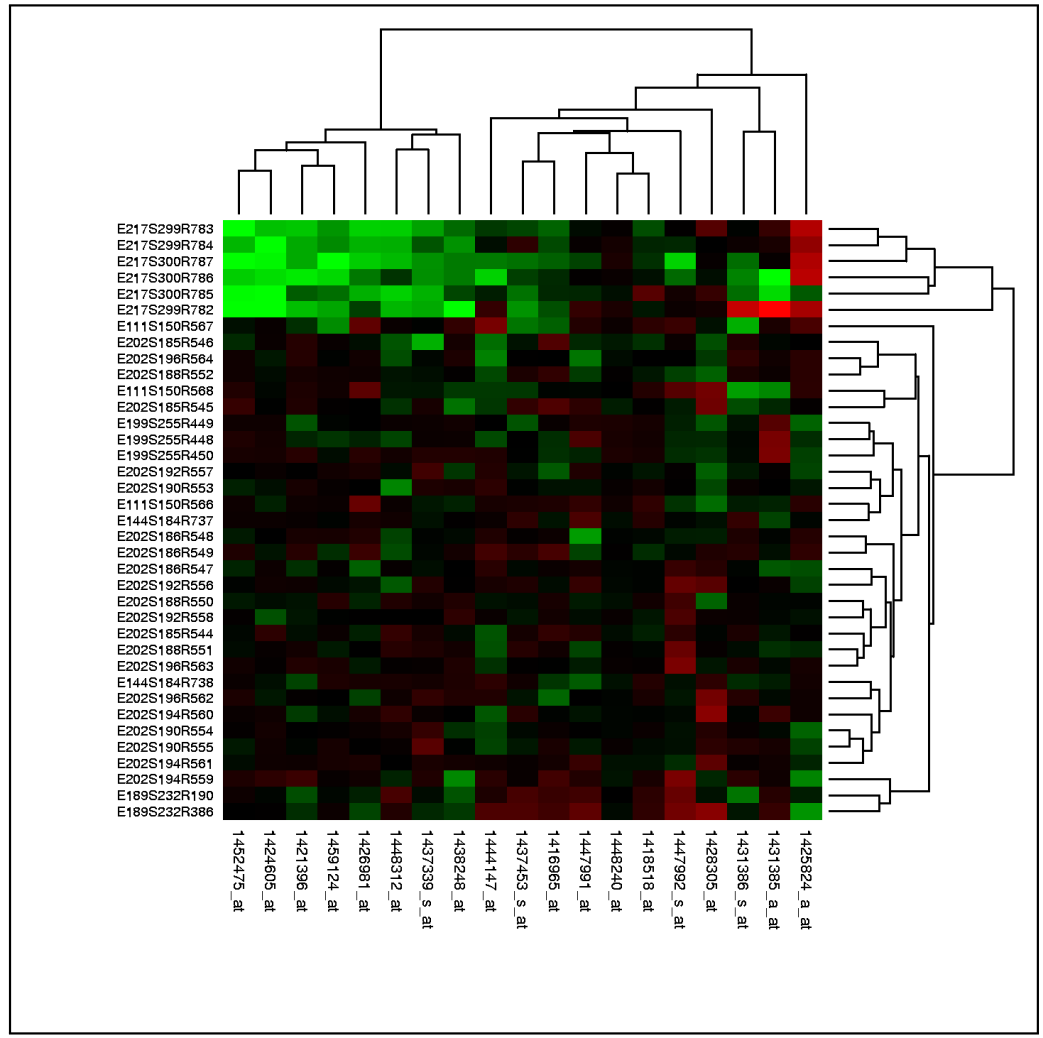
Sequence information

- Most abundant type of data around
- Rationale: interaction partners of a protein might have similar sequences, e.g. sharing a binding domain



Gene co-expression

- Abundant data in online databases such as Gene Expression Omnibus (`http://www.ncbi.nlm.nih.gov/geo/`)
- High-throughput measurements of the whole transcriptome (Microarray data, RNA-sec data)
- Rationale: Genes that are expressed in similar conditions are more likely to interact than others



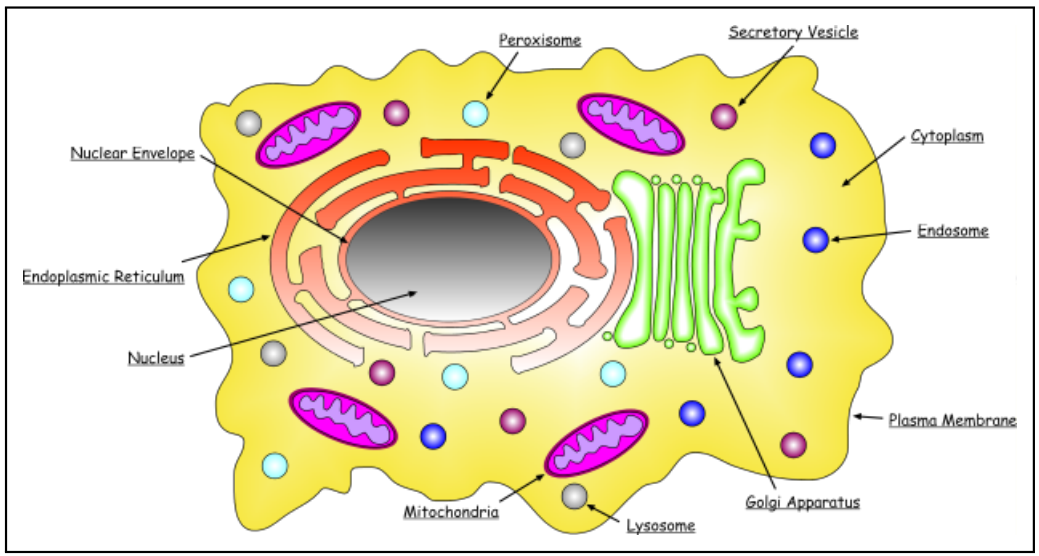
Phylogenetic profiling

- Phylogenetic profile denotes the occurrence of a given protein in a set of species
- Proteins with similar profiles more likely to interact than others

Gene	Organism A	Organism B	Organism C	Organism D	Organism E	Organism F
Gene 1	Yes	Yes	No	Yes	No	Yes
Gene 2	Yes	Yes	No	No	Yes	No
Gene 3	No	No	Yes	No	Yes	Yes
Gene 4	Yes	No	Yes	Yes	No	No
Gene 5	No	Yes	Yes	No	Yes	Yes
Gene 6	No	No	Yes	No	Yes	Yes

Subcellular localization

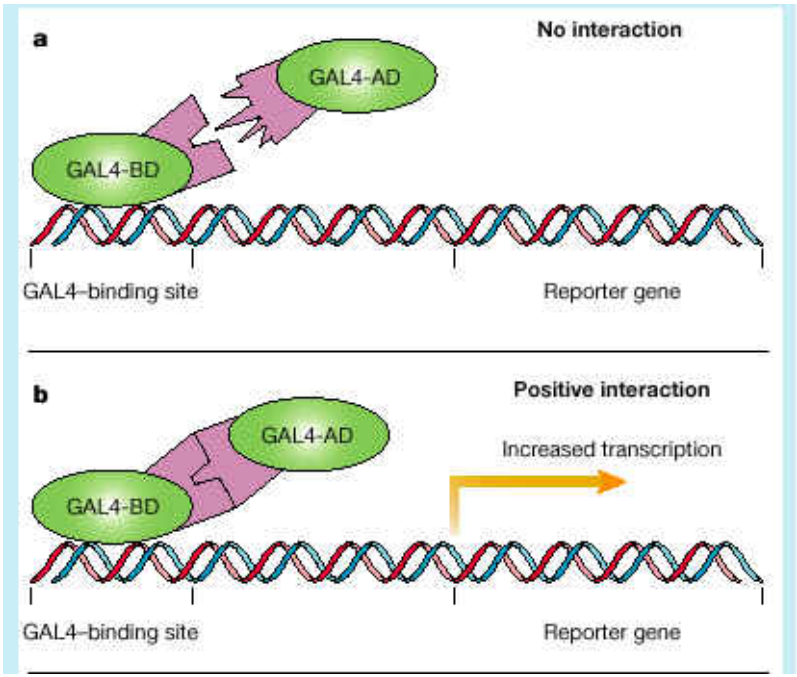
- Subcellular localization denotes where in the cell certain protein is encountered
- Proteins in same subcellular location are more likely to interact than others
- LOCATE database lists protein locations with respect over 30 subcellular locations



`http://locate.imb.uq.edu.au/`

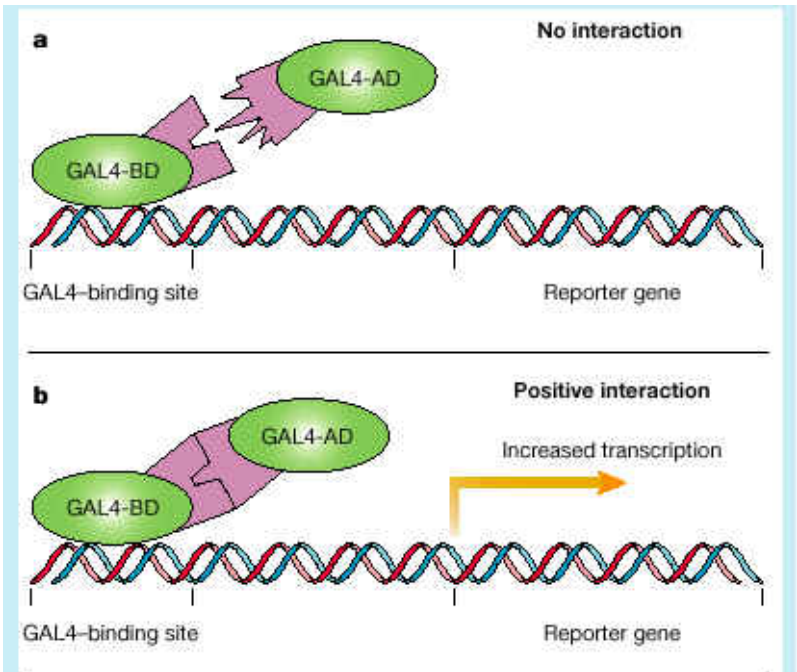
Yeast two-hybrid system

- Takes advantage of the modular structure of eukaryotic transcription factors
  - DNA-binding domain (BD) responsible of attaching the TF to the binding site
  - Activation domain (AD) that is responsible of activating the transcription
- The two domains still function as a TF if they are close proximity to each other, but do not function if they are expressed as individual polypeptides
  - Do not need to be physically part of the same molecule



Yeast two-hybrid system

- BD is fused with one of the potentially interacting protein X to make a "bait" protein
- AD is fused with the other potentially interacting protein Y to make a "prey" protein
- If X binds with Y, BD and AD are brought close each other, and the whole complex starts to work as a TF, activating the reporter gene
- The increased expression of the reporter is taken as a signal of the interaction



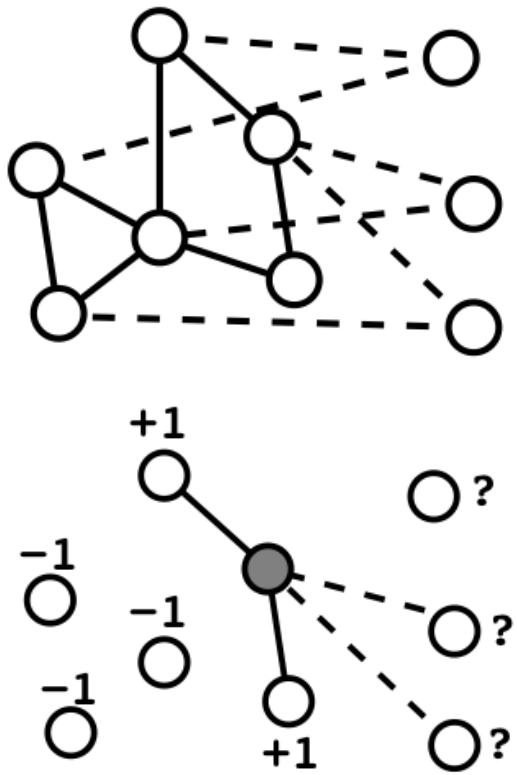
de novo inference vs. graph completion

- *de novo* inference would entail predicting the set of edges E from the feature vectors of the nodes alone
  - This is very hard statistically
  - In biology, part of the network is typically "known" already but this information is not used!
- Instead we will assume that part of the network is already known, and our task is to complete the netowrk by filling in the missing edges
  - Potentially an easier task
  - Conforms better to the way biologist work

Global and local models

The graph completion problem can be solved by global or local models

- A global model is trained to predict the absense or presence of any edge in the network, single model is needed
- A local model predicts the edges adjacent to a *seed* node, need one model per node
- In both cases the known edges are used to construct a training set from which a predictive model is learned



Graph completion as binary classification

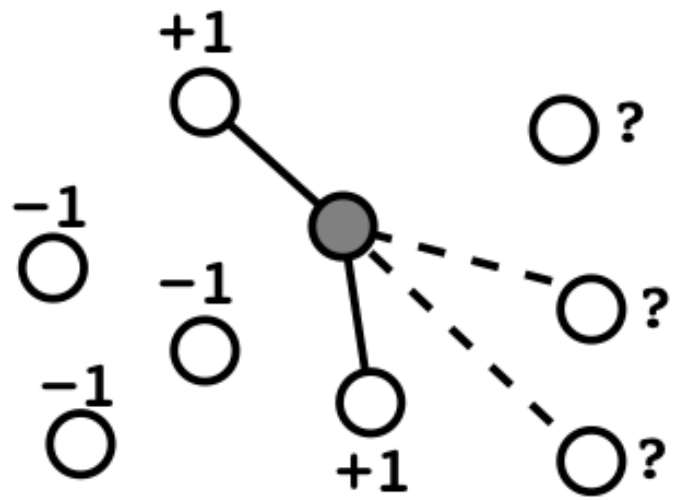
- We will formulate graph completion problem a binary classification problem (-1 = absence of an edge, 1= presence of an edge)
- Well studied branch of machine learning with many algorithms: decision trees, k nearest neighbor, Naive bayes.
- Here the method of choice is the support vector machine (SVM)

Obtaining negative examples

- For binary classification we need knowledge about edges that are *known* to be absent
- This is challenging as most of biological data available is positive data, interactions known to be present
- We need to generate pseudo-negative examples: take random pairs of nodes that are not connected and declare them absent
  - Chance of introducing errors to the network
  - Use background knowledge to choose negative examples in order to decrease this chance

Graph inference with local models

1. Take a single node v as the center for which we predict the neighbours (nodes connected with the center)
2. Create a local training set $\mathcal{S}_v = \{(u_1, y_1), \ldots, (u_{N_v}, y_{N_v}))\}$, where (v, u_i) belong to the known part of the network (known to present($y_i = 1$) or absent ($y_i = -1$))
3. Construct a kernel for the local training set $K_V(u, u') = \langle \phi(u), \phi(u') \rangle$ using the data available for the nodes

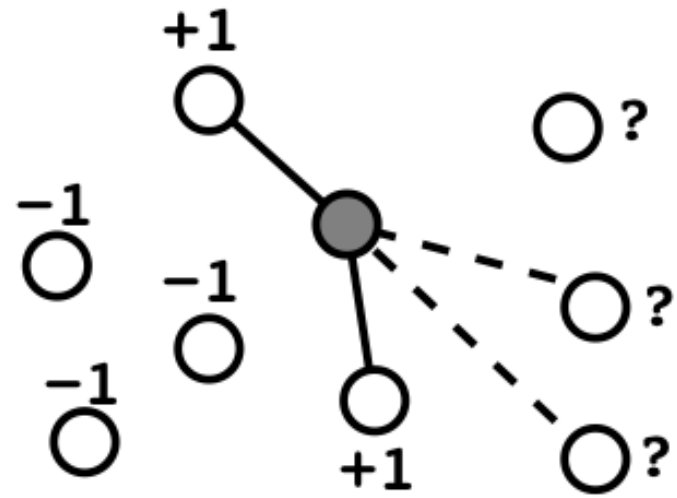


Graph inference with local models

4. Train an SVM model h_v for the node v using the local training set
5. Predict the label for each pair (v, v') that is outside the known part of the network:

$$h_v(v') = \textbf{sign}\left(\sum_{i=1}^{N_v} \alpha_i K_V(v_i, v') y_i\right)$$

6. Repeat the procedure for all nodes in the graph and complete the graph by adding all positively predicted edges



Rationale behind the method

The approach relies on the node feature vectors $\phi(v')$ to provide information on which nodes the seed node is likely to interact with

- e.g. nodes are genes and features are gene expression profiles and the goal is to predict regulatory interactions
- Then we assume that the expression profiles of genes regulated by the gene share features distinct from the features of other genes
- The classifier learns which of the features are predictive of the interaction

Use for undirected graphs

- The approach is directly applicable for directed graphs.
- For undirected graphs, each undirected training pair $\{v, v'\}$ should be considered twice, once in each direction.
- To extract the prediction for an undirected edge, the two directed predictions should be combined e.g. by averaging the scores:

$$f(\{u, v\}) = (f_v(u) + f_u(v))/2,$$

  where we denote by $f_v(u) = \langle \mathbf{w}, \phi(u) \rangle + b$ the SVM score for edge (v, u) of local classifier at node v.
- If average score is positive predict an edge $\{u, v\}$.

Pros and cons of the local method

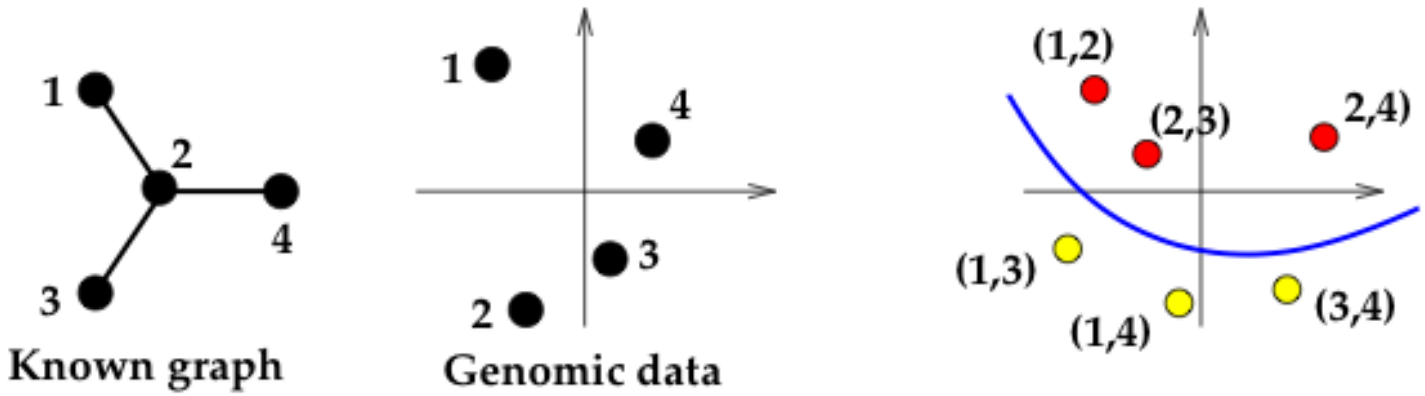
- Splitting a large network problem into a set of local problems can be beneficial in terms of computation time
  - Time to train and predict in each node gets smaller
  - Parallel architectures can be easily used as the local problems are treated independent
- Data fragmentation is a potential pitfall: if there are not enough examples for some seed nodes, accuracy of the model can suffer

Graph inference with global models

- Local approach splits the data into independent units
- Information sharing between the local problems is not possible
- e.g. if (u, v) interact, u is similar to u' and v is similar to v', the pair (u', v') is likely to interact a well
- The local approach only uses pairs with a single node as the center, so this information is not used
- To make use of the above kind of information, the model needs to be defined on edges (or pairs of nodes), not single nodes

Graph inference with global models

- We wish to represent each pair of nodes by a feature vector $\psi(u, v)$ which should contain features predictive of the interaction of that pair
- Using this representation the classifier then learns to separate interacting pairs from non-interaction
- However, our feature vectors $\phi(v)$ are defined on nodes



Features for pairs of nodes

- Consider building a feature representation $\psi(u, v)$ for pairs of nodes from feature representations of the nodes
- We want to enable learning from correlations of node features: $\phi_k(u)$ and $\phi_l(v)$ co-occur in the data
- Without assuming that exactly the same features are present
  - e.g sequence motif k in protein u co-occurs with sequence motif l in protein v

Features for pairs of nodes

- To build all feature pairs we take the tensor product (also called outer product or direct product):

$$\psi(u, v) = \phi(u) \otimes \phi(v) = \left(\phi_k(u) \cdot \phi_l(v)\right)_{k,l=1}^{d}$$

- The feature vector now maps all feature pairs between the two nodes
- The above feature representation $\psi(u, v)$ is not symmetrical: the positions os u and v matter
- For undirected graphs we average the directed features

$$\psi_{TPPK}(\{u, v\}) = \left(\psi(u, v) + \psi(v, u)\right)/2$$

Tensor product pairwise kernel (TPPK)

- The kernel, called Tensor product pairwise kernel (TPPK)

$$K_{TPPK}(\{u, v\}, \{u', v'\}) = \langle \psi_{TPPK}(u, v), \psi_{TPPK}(u', v') \rangle$$

  represents similarity of two pairs of nodes
- The classifier can now learn which co-occurring features are predictive of the interaction
- Because of the properties of tensor product, computation of a kernel from the feature representation is easy:

$$K_{TPPK}\left(\{u, v\}, \{u', v'\}\right) = \\ \left(K_V(u, u') \cdot K_V(v, v') + K_V(u, v') \cdot K_V(v, u')\right) \quad (1)$$

  where $K_V(u, u') = \langle \phi(u), \phi(u') \rangle$ is the kernel similarity of nodes.
- The kernel can be built from similar data sources as the local models

Putting it together

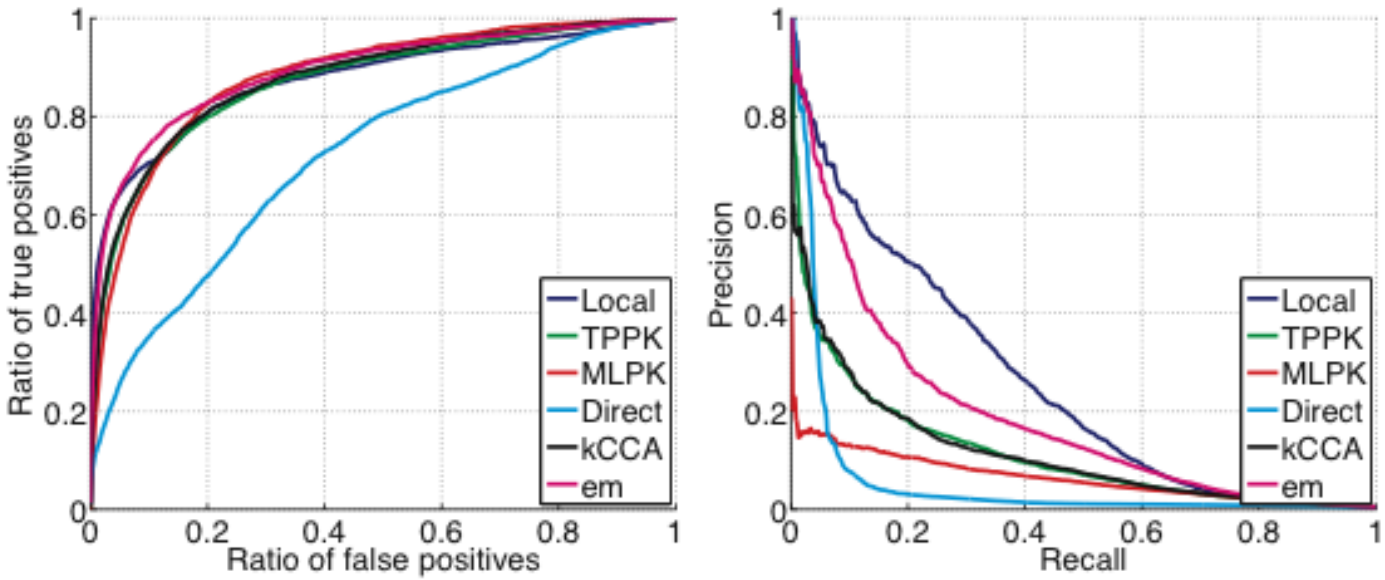
- With the global model, the training and prediction setup is straight-forward
- We take the training set of pairs $\mathcal{S} = \{(e_1, y_1), \ldots, (e_N, y_N)\}$
- Train a single SVM model
- For each pair not in the training set, predict with SVM using the TPPK kernel

$$f(\{u, v\}) = \textbf{sign}\left(\sum_{i=1}^{N} \alpha_i K_{TPPK}(e_i, \{u, v\}) y_i\right)$$

Experiments in PPI inference

Compared methods (not explained here):

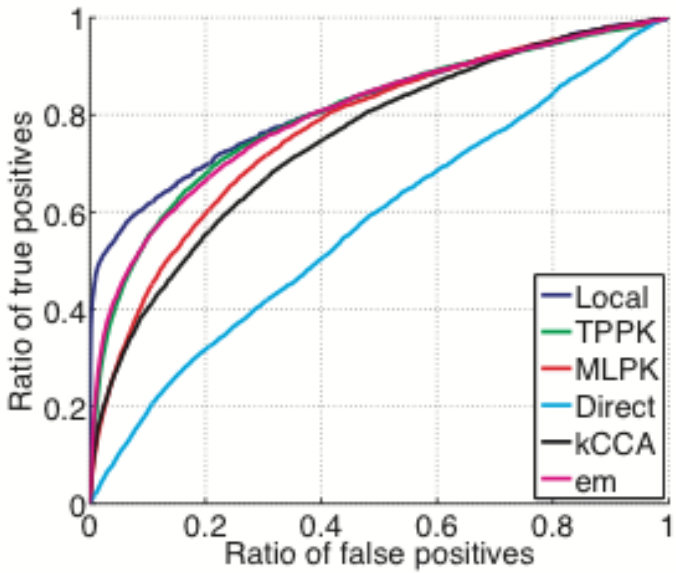
- MLPK–global model with a different kernel
- kCCA–kernel canonical correlation analysis
- em–expectation maximization based method
- Direct–de novo inference predicting edges between similar edges

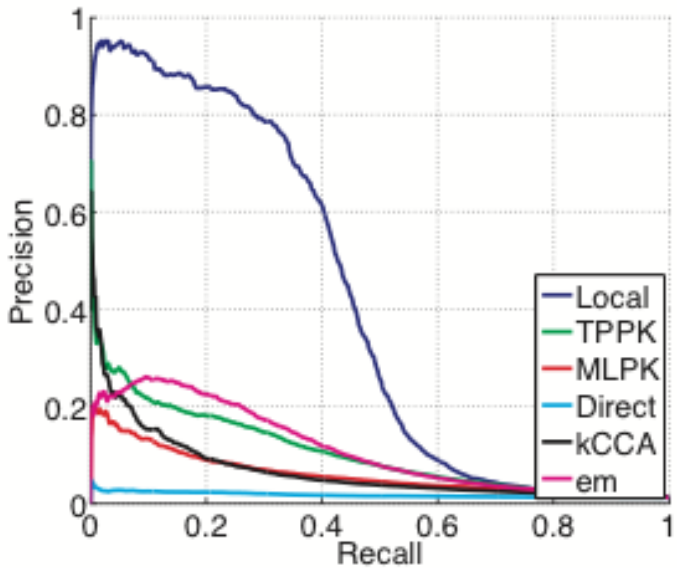


Experiments in enzyme interaction network inference

Compared methods (not explained here):

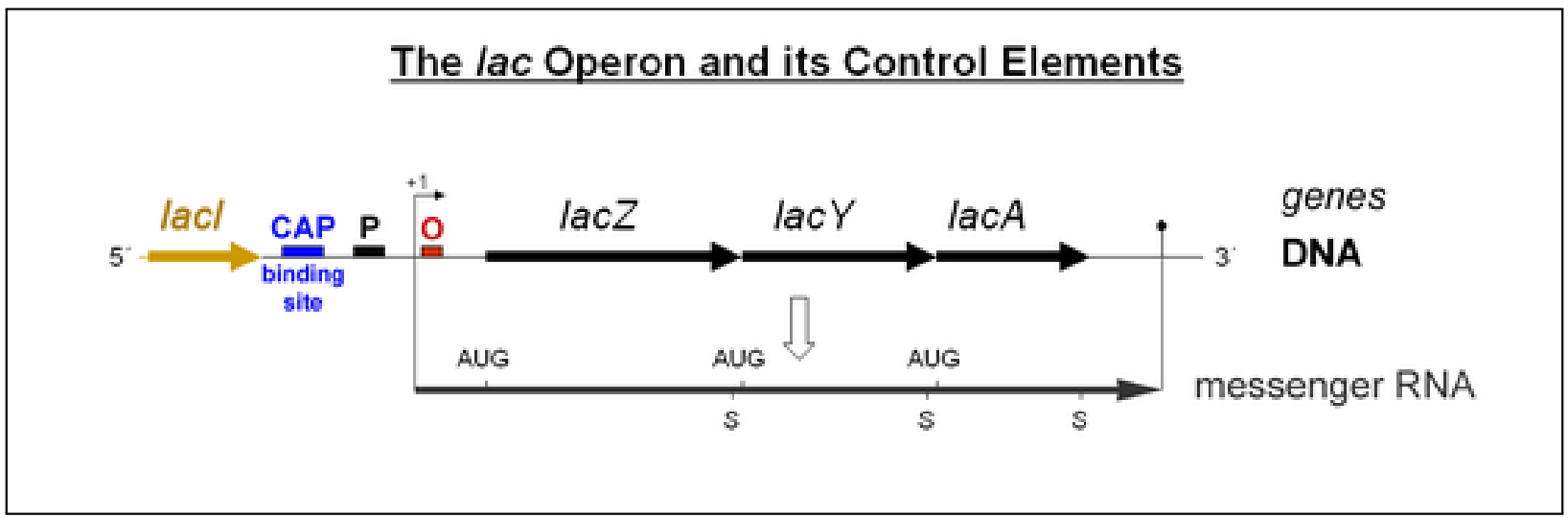
- MLPK–global model with a different kernel
- kCCA–kernel canonical correlation analysis
- em–expectation maximization based method
- Direct–de novo inference predicting edges between similar edges





Experiments for transcription regulation network inference

- Input data: gene expression data of E. coli bacteria
- Stratified cross-validation scheme used:
  - Genes that are part of the same operon typically behave very similarly
  - Given one gene from the operon in the training set, it is very easy to predict the others
  - Considered to artificially boost the predictive accuracy
  - This problem is avoided if all genes of an operon belong to the training set or test set at the same time



Experiments in transcription regulation network inference

Compared methods:

- SIRENE - local supervised model stratified for operon sharing
- SIRENE-bias - local supervised model without stratification
- CLR - context likelihood of relatedness algorithm

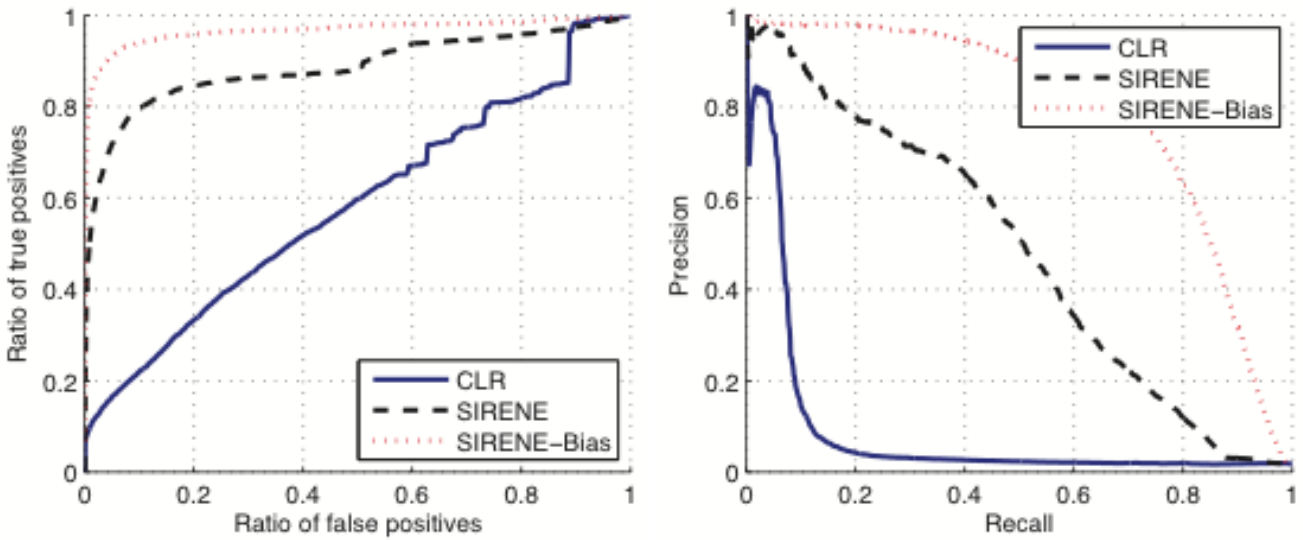


Figure 6: Comparison of the CLR method and the local pattern recognition approach (called *SIRENE)* on the reconstruction of a regulatory network: ROC (left) and precision/recall (right) curves. The curve *SIRENE-Bias* corresponds to the performance of *SIRENE* with a cross-validation