Lecture 8: Protein-protein interaction networks/
Supervised inference of network structure

17.2.2011
Protein-protein interactions (PPI)

Several types of PPIs exist:

- Two or more proteins binding to a complex to carry out a biological function
- One protein temporarily interacting with another to modify it
  - e.g. protein kinases add phosphate groups to proteins
  - modified protein typically has different interaction patterns
- Two proteins interact to catalyze adjacent reaction steps in metabolic pathways (c.f. enzyme network)
Experimental methods for detecting PPIs

- Several methods have been developed to experimentally detect PPIs
- We review two frequently used methods:
  - Yeast two hybrid system
  - Affinity capture of protein complexes
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
High-throughput screening with Y2H

- Prepare a set of clone strains with BD fused to different proteins and another one with AD fused to different proteins.
- Via robotics, mate the strains together to map all possible interacting pairs.
- Check reporter gene activity to find interacting pairs.
Affinity capture is another technique for extracting interacting proteins:

(a) Protein in fused with a ”affinity tag” and the resulting ”bait” protein expressed in a host cell

(b) The bait is captured in a purification column along with proteins bound to it

(c) The purified material is analyzed via gel electrophoresis

(d) The proteins in bands of the electrophoresis gel are identified by Mass spectrometry
Other measurement techniques

- Co-immunoprecipitation: suitable when suspected interaction targets for a protein are available
- Protein microarrays: based on immobilizing protein on a solid surface
- Phage display: based on expressing proteins on a coat of the bacteriophage
- Synthetic lethality: If two genes when knocked out at the same time cause lethality it is possible (but not certain) that they also physically interact
- ...
Quality of PPI datasets: False positives

- PPI datasets currently available suffer from high number of false positive interactions
- e.g. Y2H datasets are estimated to have 50-80% false positive
- In the picture the overlap between discovered PPIs in Human, worm and fruit fly proteomes is depicted
- If PPIs are conserved in evolution, the shared portion should be much larger
- Probably many FPs among the interactions found on just one organism

Quality of PPI datasets: coverage

- The estimated size of human interactome is 260,000 interactions between 20,000-25,000 human proteins
- Human protein reference database contains ca. 40,000 interactions
- Coverage is thus low
- May affect analyses of the PPI network structure
Early studies on PPI networks found them to have degree distribution consistent with a power law (picture).

However, the preferential attachment model fails to explain the high clustering coefficient.

Hierarchical organization has been offered as an explanation, but it is not evident in PPI networks.

Alternative model: Geometric random graph

Evolution of the network via duplication and divergence:

- When protein (gene) is duplicated it initially inherits all the interactions of its parent
- As the new protein evolves, some of the interactions will disappear, but new ones may arise
- It is assumed that the interactions happen between proteins that are close to each other in a high-dimensional "biochemical space"
- Modelled by geometric random graph: vertices are drawn uniformly randomly in a metric space, edges between "close enough" vertices
Alternative model: Geometric random graph

- Here comparison is to a geometric random graph with the same number of vertices as the PPI network but 6 times more edges.
- Unlike preferential attachment models (left), the geometric model can explain the high clustering coefficient and its behaviour as a function of vertex degree (right).

Comparing PPIs in terms of network motifs

- Real PPIs have been compared to random networks via their graphlet distribution
- Graphlets are small induced subgraphs
- All 29 graphlets of 3-5 vertices are shown in the picture

Alternative model: Geometric random graph

- The geometric model with 6 times more edges can explain the observed distribution of graphlets, that is, small induced subgraphs to astounding accuracy (picture right).
- Preferential attachment models fail to match the graphlet distribution.

Robustness of PPI networks

- In early studies PPI networks were shown to have the "robust, yet fragile" property
  - Random deletions of vertices do not change the overall topology easily
  - Deletions targeted to hubs break the network easily
- Taken as evidence that the hubs are functionally "essential"
- This view has been later challenged: perhaps hubs look more essential because they have many connections which are broken in a single deletion
  - More like to remove an essential connection just by chance
Supervised inference of biological networks

- We will review a machine learning method for inferring missing edges in biological networks.

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 (genes, proteins, metabolites, reactions)
- Each vertex has an associated feature vector \( \phi(v) \) describing the vertex, e.g. expression levels of a gene, sequence motifs present in the gene
- We wish to reconstruct a set of edges \( E \subset V \times V \) that define the biological network
Data sources for PPI prediction

- The feature representation of vertices is built from data describing the "genomic context" the protein
- Examples of data sources for interaction prediction:
  - Gene co-expression: if the genes of two proteins are expressed together, they may belong to a common protein complex
  - Phylogenetic profiling: if two proteins occur in close proximity of each other across multiple genomes, it increases the likelihood that they interact
  - Sub-cellular localization: if the proteins occur in the same compartments, they may physically interact
  - Sequence information: if some interaction partners are known, their sequences may be used predict other interaction targets
de novo inference vs. graph completion

- *de novo* inference would entail predicting the set of edges \( E \) from the feature vectors of the vertices 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 network 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 absence or presence of any edge in the network, single model is needed
- A local model predicts the edges adjacent to a seed vertex, need one model per vertex
- 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)
  - One of the most widely used classification methods in bioinformatics
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 vertices 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
Support vector machine (SVM)

- SVM estimates a linear score function

\[ h(x) = w^T \phi(x) = \sum_j w_j \phi_j(x) [+b] \]

- \( w \) is the weight vector to be learned
- \( \phi(x) \) is the feature vector of object \( x \)
- \( b \) is an optional bias term

- SVM predicts the class by taking the sign of the score:

\[ f(x) = \text{sign}(h(x)) \in \{-1, +1\} \]
Support vector machine (SVM)

- Learning of $w$ aims to enforce large absolute values (margin) for $h(x_i)$ on training points $x_i$ while controlling the norm of $w$.
- Geometrically corresponds to finding a separating hyperplane between the classes that has a large margin.
- Data points closest to the margin are called support vectors, they are sufficient to define the separating hyperplane.

$H_2$ is the solution returned by SVM, $H_1$ separated the classes correctly but has a small margin, $H_3$ does not separate the classes correctly.
Use of kernels

A distinct feature of SVMs is that they can be trained and used when there is no access to original data, just the similarities of training points given by a kernel function $k(x, x') = \phi(x)^T \phi(x')$

The prediction is given by a linear combination of kernel function values of support vectors (they have $\alpha_i > 0$)

$$f(x) = \text{sign}(w^T \phi(x)) = \text{sign}(\sum_i \alpha_i y_i k(x_i, x))$$
Assume $\phi_1(x)$, $\phi_2(x)$ and $\phi_3(x)$ are three different feature sets for gene $x$

- e.g. expression profile, sequence motifs, phylogenetic profile
- Learning with concatenated feature vector

$$\phi(x) = \left( \phi_1(x)^T, \phi_2(x)^T, \phi_3(x)^T \right)^T$$

is equivalent of learning with the sum of kernels

$$K(x, x') = K_1(x, x') + K_2(x, x') + K_3(x, x')$$
Data integration with kernels

- Learning with all combined features of the type

$$\phi_{1,i}(x) \cdot \phi_{2,j}(x) \cdot \phi_{3,k}(x)$$

is equivalent of learning with elementwise product of kernels is equivalent of learning with the sum of kernels

$$K(x, x') = K_1(x, x') \cdot K_2(x, x') \cdot K_3(x, x')$$

- Useful, e.g. for picking up co-occurrences of features in different datasets
Graph inference with local models

1. Take a single vertex $v$ as the center for which we predict the neighbours (vertices connected with the center)

2. Create a local training set $S_v = \{(u_1, y_1), \ldots, (u_{N_v}, y_{N_v})\}$, where $(v, u_i)$ belong to the known part of the network (known to present ($y_i = 1$) or absent ($y_i = -1$)).
Graph inference with local models

3. Train SVM with the local training set

4. Predict the label for each pair \((v, v')\) that is outside the known part of the network:

\[
f_v(v') = \text{sign} \left( \sum_{i=1}^{N_v} \alpha_i K_V(v_i, v') y_i \right)
\]

5. Repeat the procedure for all vertices in the graph and complete the graph by adding all positively predicted edges
Rationale behind the method

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

- e.g. vertices 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:

\[
h(\{u, v\}) = \frac{h_v(u) + h_u(v)}{2},
\]

where we denote by \( h_v(u) \) the score for edge \((v, u)\) of local classifier at vertex \(v\).
- 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 vertices, 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 as well
- The local approach only uses pairs with a single vertex 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 vertices), not single vertices
Graph inference with global models

- We wish to represent each pair of vertices 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 vertices.
Features for pairs of vertices

- Consider building a feature representation $\psi(u, v)$ for pairs from feature representations of the vertices.
- We generally want to enable learning from correlations of vertex features (e.g. $\phi(u)_k$ and $\phi(v)_l$ are 'high' at the same time) without assuming that exactly the same features are present (e.g. sequence motifs of interacting genes may be different different).
- To build all feature pairs we take the tensor product (also called outer product or direct product)

$$
\psi(u, v) = \phi(u) \otimes \phi(v) = (\phi(u)_k \cdot \phi(v)_l)^d_{k,l=1}
$$

- The classifier can now learn which co-occurring features are predictive of the interaction.
Undirected case: TPPK kernel

- The tensor product feature representation $\psi(u, v)$ is directed.
- For undirected graphs one can average the directed features:
  \[
  \psi(\{u, v\}) = \frac{(\psi(u, v) + \psi(v, u))}{2}
  \]
- Computation of a kernel from the feature representation is easy:
  \[
  K_{TPPK}(\{u, v\}, \{u', v'\}) = (K_V(u, u') \cdot K_V(v, v') + K_V(u, v') \cdot K_V(v, u'))
  \]
  where $K_V(u, u') = \phi(u)^T\phi(u')$ is the kernel similarity of vertices.
- TPPK stands for "tensor product pairwise kernel"
Putting it together

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

\[
f(\{u, v\}) = \text{sign} \left( \sum_{i=1}^{N} \alpha_i K_{TPPK}(e_i, \{u, v\})y_i \right)
\]
Experiments in PPI inference

- Local and global (TPPK) methods were used in reconstruction of S. cerevisiae PPI network
- Performance curves are based on thresholding the score at different thresholds (e.g. for the global model $h(\{u, v\}) < \tau \implies -1$, with different $\tau$):
  - ROC curve: True positive (TP) rate against False positive (FP) rate
  - Precision-Recall curve: $Precision = TP / (TP + FP)$, $Recall = TP / (TP + FN)$ ($FN =$ False negative)
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