10-810: Computational Molecular Biology: a machine learning approach

Physical networks and network motifs
Yeast mating pathway

• A graph depicting physical interactions and functional annotations.
A mechanistic model of gene regulation

- Physical data:
  - Yeast location analysis
  - DIP database
- Functional data:
  - Rosetta compendium data
Inferring the mechanistic model from observed data

Key question: How do we construct the model from known mechanisms and constraints from observed data?

• Decompose data into pairwise items.
• Construct potential functions specifying constraints of each item.
• Combine potential functions by multiplication.
Associations with binding data

• Location data:
• Given a possible protein-DNA interaction $e_i$, the potential function $\phi_{ei}(x_{ei}; y_{ei})$ is related to the direct evidence about this interaction:

\[
\phi_{ei}(x_{ei}; y_{ei}) = \left[ \frac{p(y_{ei} \mid x_{ei} = 1)}{p(y_{ei} \mid x_{ei} = 0)} \right]^{x_{ei}}
\]

• And similarly for protein interaction.
Determining the confidence in the observed data

- In order to determine the probabilistic term in the potential function we use an appropriate error model.
- As a crude approximation, $p(y_{ei} \mid x_{ei})$ can be obtained from the binding p-value.
- First, they set $p(\text{measurement} \mid \text{interaction does not exist}) = p\text{-value}$.
- The other side $p(\text{measurement} \mid \text{interaction exists})$ is set to a fixed value.
- The potential term for the protein interaction case is defined analogously.
Associations with knock-out data

- Relevant variables:
- The interaction effect is associated with the observed data $o$ by:

$$\phi_{i,j}(k_{i,j}, o_{k,i,j}) = \left[ \frac{p(o_{k,i,j} | k_{i,j})}{p(o_{k,i,j} | k_{i,j} = 0)} \right]$$

- $k$ can be explained by cascades of molecular interactions, i.e., paths in the physical model.
Associations with knock-out data

• In order to explain a pair of knock-out interaction $\chi_{(g_1,g_2)}$ attributes its connecting paths must satisfy the following conditions:
  • There is at least one connecting path.
  • Edge directions along the path are consistent with the knock-out effect.
  • The last edge on each path is a protein-DNA edge.
  • The aggregate sign along the path is consistent with the knock-out effect.
  • Intermediate genes along the path either have knock-out effects on $g_2$ or were not tested.
  • The path length is upper bounded.
Knockout (cont.)

- Explanation conditions can be expressed as a logic clause of variables along the paths connecting a knock-out pair:
  - the knock-out interaction effect ($\chi_k$)
  - edge presences ($E_k$),
  - edge directions ($D_k$), and sign ($S_k$),
  - and path selections ($\Sigma_k$).

- The potential term can also incorporates the situations of multiple paths and uncertainties of explanation.
Inference

- Potential functions are combined by multiplication.
- Goal: find the optimal configuration of the variables.
- This is done using a maximum likelihood approach using a variant of belief propagation.
- Using a graph known as a factor graph, the max-product algorithm is applied to obtain a MAP configuration.
- If the network is small, we can apply the max-product recursively to obtain all MAP configurations.
Datasets

- 46 genes including 2 transcription factors (STE12 and MCM1).
- Binding p-value threshold 0.001 result in 34 protein-DNA edges (Lee et al., 2003).
- 30 protein-protein edges (DIP).
- 164 knock-out pairs from 10 experiments (Hughes et al., 2000).
- Maximal path length set to 5.
Results: yeast mating pathway

- 129 knock-out pairs are connected via valid paths.
- 8 MAP configurations.
- 129 knock-out pairs are explained by all MAP models.
- 106 knock-out pairs are explained by non-trivial inference.
- 2 knock-out pairs whose explanatory paths are not constrained by other knock-out pairs
Robustness of the model

- Are prediction outcomes sensitive to parameter settings?
- Robustness tests on location and knock-out p-value cutoffs, potential values and path length
Common features for all MAP models
Variant features
Variant features
Network motifs
Networks in complex systems

- Network is the **backbone** of a complex system
- Answers the question: **who interacts with whom?**
- Recently studied networks:
  - **Biological networks:** interacting biomolecules (metabolic, physical, regulatory); food webs in ecosystems
  - **Technological networks:** Internet and WWW
  - **Social networks:** collaborations between scientists, etc.
Why study networks?

• Studying networks is **simpler** than full dynamics of a complex system!
• **Large networks** may contain information about **design principles** and/or **evolution** of the complex system
• **Lots** of easily available data
Transcription regulatory network in yeast

- Downloaded from the YPD database: 1276 regulations among 682 proteins by 125 transcription factors (10 regulated genes per TF)
- Part of a bigger genetic regulatory network of 1772 regulations among 908 proteins
- Positive to negative ratio 3:1
- Broader distribution of out-degrees (up to 72) and more narrow of in-degrees (up to 21)
Transcription regulatory network in *Homo Sapiens*

- Data courtesy obtained from literature search: 1449 regulations among 689 proteins
- Positive to negative ratio is 3:1 (again!)
- Broader distribution of out-degrees (up to 95) and more narrow of in-degrees (up to 40)
Transcription regulatory network in *E. coli*

- Data was curated from the Regulon database: 606 interactions between 424 operons (by 116 TFs)
- Positive to negative ratio is 3:2 (different from eukaryots)
- Broader distribution of out-degrees (up to 85) and more narrow of in-degrees (only up to 6)
Yeast protein interaction network

- Data from the high-throughput two-hybrid experiment (T. Ito, et al. PNAS (2001))
- The full set containing 4549 interactions among 3278 yeast proteins
- 87% nodes in the largest component
- The highest connected protein interacts with 285 others
- Figure shows only nuclear proteins
Common properties of complex (biological) networks

- Very connected (one giant cluster plus a few very small ones)
- Broad distribution of connectivities: hubs interacting with many neighbors, low-connected nodes with just a few neighbors, and everything in-between.
Random vs designed/evolved features

- Which features of a large complex network are random?
- Which features are there for a reason:
  - design principles (e.g. feed-forward loops)
  - constraints (e.g. all nodes on the Internet must be connected to each other)
  - evolution, growth dynamics (e.g. network growth is mainly due to gene duplication)
How to detect design elements of a large complex network?

- Main idea: Compare the abundance of some topological property in the real and properly randomized networks.
Simplest random network

- **Erdos-Renyi** model: randomly draw $E$ edges between $N$ nodes
- Conserves only the **average** number of neighbors (connectivity) of a node
  
  $\langle k \rangle = \frac{2E}{N}$

- **No hubs!** Narrow distribution of connectivities
How to detect design elements of a large complex network?

• Main idea: Compare the abundance of some topological property in the real and properly randomized networks.
• Must retain the global features of the network when looking for motifs.
• Same distribution of in and out degrees (preserving the hub structure of the network).
How to prepare such a random network?

- **Stub reconnection algorithm** (M. E. Newman, et al, 2001, also known in mathematical literature since 1960s)
- Break every edge in two “edge stubs”
  \[ A \rightarrow B \text{ to } A \rightarrow B \]
- Randomly reconnect stubs
- Problems:
  - Leads to multiple edges
  - Cannot be modified to preserve additional topological properties
Local rewiring algorithm

- Randomly select and **rewire** two edges (Maslov, Sneppen, 2002, also known in mathematical literature since 1960s)
- Repeat **many times**
- Preserves both the number of **upstream** and **downstream** neighbors of each nodes
Identified network motifs

- Compared the abundance of small loops in *E. coli* transcription regulatory network to its randomized counterpart.

- There are 13 types of 3-node connected, directed subgraphs.

- Only Feed-Forward Loops (FFL) were significantly over-represented (40 in real vs 7 +/- 5 in random).
Coherent vs incoherent FFLs

- In E.coli only coherent feed-forward loops were over represented
- Coherent FFL - sign-sensitive filter
- Incoherent FFL – sign-sensitive differentiator
Other motifs identified