Protein-protein interaction networks

Q: Why study protein-protein interaction networks, isn't it a rather specialized topic? A: Potentially not, since proteins as the central workhorses contribute to almost everything there happens in a cell. Mutual interactions between proteins and between proteins and other molecules provide the potential mechanisms out of which the suitable ones became chosen. Modeling the potential interactions gives constraints to modeling cellular pathways: regulation, signaling and metabolism. These pathways or their parts are natural candidates for disease markers, and maybe also drug targets.

The measurements are very noisy and hence models are needed to cope with the uncertainty in them. A side benefit of building such models is that they may be applicable to other kinds of relational data as well.

1 Background

Proteins are central workhorses in the cell. They form the physical structure of the cell and move the cells; they regulate gene activity as transcription factors and through DNA methylation (gene silencing); they catalyze bio-chemical reactions of metabolism as enzymes, and convey signals between cells and from the outside world (signal transduction). Moreover, some proteins (kinases) activate (phosphorylate) and others (phosphatases) deactivate (dephosphorylate) other proteins, mark them for further processing (ubiquitination), and cleave them into pieces.

Proteins consist of chains formed of the 20 different amino acids. Typical length is 50-1000 amino acids but they may be longer. This forms a huge number of possible proteins, out of which only a tiny fraction is actually found in nature.

The amino acid chain folds into a characteristic formation, a minimal energy configuration. The resulting 3D physical form, and the properties and locations of active domains capable of binding to other proteins, give the protein its functional properties. It would make a lot of sense to take information about the structure of proteins into account in modeling the interactions, but it is a different (hard) problem and we do not have the time to go into that now.

Single proteins rarely achieve great deeds but working together in *pathways* they may. A pathway is an abstraction describing a function produced by a set of players, including proteins, co-operating to form a succession of reactions. There are three types of pathways: Signaling pathways convey messages from the outside to the regulatory apparatus of the cell. Metabolic pathways are series of chemical reactions that take food and other molecules and transform them to be stored, to be used as building blocks, or to extract energy. The third type, gene regulatory networks, governs gene activity and production of proteins.

The different kinds of pathways naturally interact; gene regulation transforms metabolism, and signaling pathways change gene regulation, for instance. It may be sensible to consider pathways as abstractions of context or condition-specific mechanisms at work. They are formed of the set of (potential) interactions, *interactome*, of the molecules in the cell, and modeling the potential interactions gives border conditions for studying the pathways or discovering new ones.

2 Interaction data

Today we will focus on protein-protein interactions (PPI), which are the mechanisms underlying much of the cellular function. Proteins bind to each other to form complexes; interactions are the mechanisms by which the signaling pathways work; proteins interact to activate or shut down other proteins, or to move other proteins within the cell.

Another good reason for studying interaction data is that there are comprehensive, albeit noisy, PPI data sets available as a result of developments of the measurement techniques. They might be usable as one data source in other modeling tasks, such as modeling of gene regulation or search of disease markers or drug targets. Lastly, if the models are general enough, similar kinds of models could be used to model interactions between other biological molecules, and in fact other kinds of relational data as well, such as social networks, customer relationships, or the hypertexts of the web.

The interactions are due to properties of the 3D protein molecule, where local *domains* may be active in binding reactions. Here we will focus on the level of networks of proteins and not try to model the fine structure or nature of the interaction. This naturally loses a lot of information but makes the modeling task easier. Again, it would make a lot of sense to take protein structure into account as well.

Yeast two-hybrid (Y2H) is a widely used method for measuring PPIs. It uses gene expression measurements and gene regulatory machinery in an ingenious way. It uses the particular property of transcription factors (TFs), activators of gene expression, that they have two separate domains, of which both are necessary for activating the gene. A binding domain is needed for the TF to bind to the binding sites in the promoter region of the gene's DNA, and an activator domain actually activates the gene. For the Y2H measurements the TF is cut in two such that the two domains end up in different subparts. Each subpart of the TF is then fused with one of the two proteins whose interaction is to be measured. The subparts come together if the two proteins interact, and then the transcription factor is functional and activates the gene. Activity of the reporter gene then reveals interaction between the two proteins; and based on the beginning of the course we know well how to measure gene activity.

The measurements can be done *in vivo*, in the living thing, and can detect even transient bindings and, perhaps most importantly, can be done on a massive scale. The main problem is noise; spurious interactions and blockings between the players may tweak the results. And it must be kept in mind that Y2H only detects *potential* interactions, not any specific set of interactions active in specific conditions or even specific cells, and only pairwise interactions.

Another widely used technique is *co-immunoprecipitation* (coIP). There one protein, "target" or "bait" is equipped with a tag, with which all targets can be pulled out (precipitated) at a desired time. All other proteins and protein complexes that have been bound to the target become pulled out at the same time. The molecules can then be identified using 2D gels and mass spectrometry, for instance.

Advantages of this process are that it is more accurate, capable of measuring multiway interactions (with complexes), and all interactions occur in natural conditions (instead of in the nucleus as in the yeast two-hybrid). A disadvantage is that when large protein complexes become bound with the target, it is not known which protein(s) in the complex are primarily responsible for the binding. Moreover, MS-based detection may miss low-abundance proteins.

It is slightly worrisome that overlap between the sets of PPIs found in different studies has been very low. But that only highlights the need for tools to denoise the data. Much of the noise probably stems from differences in the biological settings in which the measurements were made, and differences in sample preparation. Some comes from the measurement process, and statistical error models could be developed for it, analogously to the error models in other high-throughput data. We will have to skip those issues here; I am not aware of conclusive works on these topics. What has been done is to integrate several data sources to computationally increase the confidence in the links. We will return to that in the next section.

3 Denoising and untangling

PPI measurement data is very noisy and so far there are no completely satisfactory noise models for them available. There are, however, several different kinds of measurement methods, all revealing partly different aspects of interactions. Integrating them gives a better picture than using only one of them. Another source of evidence is the network nature: instead of considering each interaction separately we could model the whole network.

3.1 Supervised link prediction

Let's start by integrating measurement sources in a *supervised* setting: Assume that for a subset of the edges, $(i, j) \in E_L$, it is known whether an interaction exists, $e_{ij} = 1$, or not, $e_{ij} = 0$. This knowledge can come from curated databases or from some accurate measurement technique, and our task is to generalize this knowledge to the other links, in the set $E \setminus E_L$. Assume that for each edge we have a set of observations $\mathbf{x} = [x_1, x_2, \ldots]$, which are PPI measurements made with different techniques, or interactions inferred from co-expression or some other properties of the genes. Using the Bayes rule, the posterior probability of an edge e existing, taking into account the observations $\mathbf{x}$, is

$$P(e = 1|\mathbf{x}) = \frac{P(\mathbf{x}|e = 1)P(e = 1)}{P(\mathbf{x}|e = 1)P(e = 1) + P(\mathbf{x}|e = 0)P(e = 0)}$$
$$= \frac{1}{1 + \exp(-\log \frac{P(\mathbf{x}|e=1)}{P(\mathbf{x}|e=0)} \frac{P(e=1)}{P(e=0)})} .$$

This is a sigmoid function ($\sigma(u) = 1/(1 + exp(-u))$ of the (log of the) likelihood ratio $P(\mathbf{x}|e = 1)/P(\mathbf{x}|e = 0)$ multiplied with the prior odds $P(e = 1)P(e = 0)$. The sigmoid function transforms the log odds, which may lie anywhere on the real axis, to probabilities that are in $[0, 1]$.

For computational simplicity we may want to make the simplifying "naive Bayes" assumption that the different data sources are independent given e. Then the likelihoods factorize, and hence the likelihood ratio factorizes as well to

$$\frac{P(\mathbf{x}|e = 1)}{P(\mathbf{x}|e = 0)} = \prod_k \frac{P(x_k|e = 1)}{P(x_k|e = 0)} . \tag{1}$$

Given discrete-value data, these factors can be estimated from tabulated values in the learning data where e is known. If the data is not discrete-valued it needs to be discretized, or then parametric assumptions about $P(x_k, e|\theta)$ can be made, and maximum likelihood estimates of the parameters θ used.

This approach is useful because of its simplicity, but it has the obvious weakness that the different measurements x_k are obviously not independent. The model can be made richer by inferring a richer dependency structure for $\mathbf{x}$ using more flexible Bayes networks. The second weakness is that unless the measurement data inherently is discrete-valued the discretization will lose information, which could be avoided by using the continuous-valued data directly. Then it would be sensible to use *discriminative models*; not to first do maximum likelihood estimation for $P(\mathbf{x}, e|\theta)$, but instead directly maximize the conditional likelihood $P(e|\mathbf{x}, \theta)$. With certain simplifying assumptions this results in *logistic regression* where $P(e = 1|\mathbf{x}) = 1/(1 + \exp(-\theta^T\mathbf{x} - b))$, and $p(e = 0|\mathbf{x}) = 1 - p(e = 1|\mathbf{x})$. Alternatively, any other classification algorithm can of course be used.

3.2 Untangling with a factor graph

A lot based on [2].

So far we have not considered dependencies between the links at all, as a source of information for inferring interactions from noisy measurement data. If we want to study networks instead of all links separately, we implicitly assume that there are some underlying constraints on the link structure of the network, and it would be good to take this implicit assumption into account in the models too.

So let's take seriously the fact that the interactions form a graph. Assume a graph with a fixed set of N vertices. The variables e_{ij} indicate edges as earlier, and assume observations (measurements) for the edges, indexed in the same way as the edges, x_{ij}. The observations may be multivariate (made with serveral different methods), and some observations may be missing. For simplicity, let's start with scalar and binary-valued observations (edge is there or is not according to Y2H), and full measurement matrix. For simplicity, the edges are assumed undirected, so that $e_{ij} = e_{ji}$ and $x_{ij} = x_{ji}$.

In order to do Bayesian inference about the structure of the graphs, we need to have a prior over graphs. When treating each edge separately they were in effect assumed independent (given parameters of the noise model) which, translated into a prior, would imply a prior that factorizes into terms that only involve one edge each:

$$p(E, X|\theta) \propto p(X|E, \theta)p(E) = \prod p(x_{ij}|e_{ij}, \theta)p(E) = \prod p(x_{ij}|e_{ij}, \theta)p(e_{ij}) \ .$$

The priors of the edges were assumed to be same for each edge, and furthermore uniform, effectively resulting in maximum likelihood estimation for the parameters θ of the edge noise.

Now we would like to introduce constraints on the networks. Inspired by the recent studies on properties of degree distributions in random and social networks, it might make sense to pose priors on the degree distribution and assume that all graphs having the same degree distribution are equally likely.

Denote the degree of vertex i by d_i. The degree is a constraint on the edges, such that $\sum_j e_{ij} = d_i$ must hold. Now

$$P(E, X, D|\theta) = \prod P(x_{ij}|e_{ij}, \theta)P(E, D) = \prod P(x_{ij}|e_{ij}, \theta)P(E, D) \ .$$

Determining the prior distribution over d_i by a non-negative potential $f_i(d_i)$, and assuming graphs that have the same vertex degrees to be equiprobable, gives

$$P(E, D) \propto \prod_i f_i(d_i) I(d_i, \sum_j e_{ij}) \ ,$$

Here the "$\propto$"takes care of proper normalization of the potentials, and I makes sure that the prior differs from zero only if the constraint that $\sum_j e_{ij} = d_i$ holds. $I(a, b) = 1$ if $a = b$ and 0 otherwise.

When the joint distribution factorizes into a product it can be conveniently represented as a *factor graph*; many inference algorithms, in particular the *sum-product algorithm* also called *loopy belief propagation* are more easily derived for factor graphs than for alternative formulations, Bayesian networks or Markov random fields. What is also nice about factor graphs is that both directed (Bayesian networks) and undirected (Markov random fields) graphical models can be easily converted to factor graphs.

In the factor graph there are two kinds of nodes: variables (denoted by circles or simply their names, and factors denoted by filled squares. Each factor is connected to all the variables on which it depends.

This factor graph is useful for inferring which edges underlying the noise in the measurements are real, assuming the degree priors and the conditional probabilities $p(x|e)$ have sufficiently useful forms. Simple forms will be tried in the exercise session. The remaining problem then is that the constraint $\sum_j e_{ij} = d_i$ quickly becomes impossibly hard to compute with for large graphs; there is a nice solution to this in the original paper.

When the joint density is described as a factor graph, its margins can be computed with the sum-product algorithm. In our case, the particularly interesting margins are the $p(e_{ij})$ that tell whether an edge exists or not. This requires summing over all variables except e_{ij}. In the current model, for two proteins the summation to get e_{11} would be

$$\begin{aligned}
P(e_{11}|x_{11},x_{12},x_{22}) &= \sum_{e_{12},e_{22},d_1,d_2} P(e_{11},e_{12},e_{22},d_1,d_2|x_{11},x_{12},x_{22}) \propto \\
&\sum_{e_{12},e_{22},d_1,d_2} P(x_{11}|e_{11}) I(d_1,e_{11}+e_{12}) f_1(d_1) P(x_{12}|e_{12}) I(d_2,e_{12}+e_{22}) f_2(d_2) p(x_{22}|e_{22}) \\
&= P(x_{11}|e_{11}) \sum_{d_1,e_{12}} I(d_1,e_{11}+e_{12}) f_1(d_1) P(x_{12}|e_{12}) \sum_{d_2,e_{22}} I(d_2,e_{12}+e_{22}) f_2(d_2) p(x_{22}|e_{22}) \\
&= P(x_{11}|e_{11}) \sum_{d_1,e_{12}} I(d_1,e_{11}+e_{12}) f_1(d_1) P(x_{12}|e_{12}) \mu_{I_2\to e_{12}}(e_{12}) \\
&= P(x_{11}|e_{11}) \sum_{d_1,e_{12}} I(d_1,e_{11}+e_{12}) f_1(d_1) \mu_{e_{12}\to I_1}(e_{12}) \\
&= P(x_{11}|e_{11}) \mu_{I_1\to e_{11}}(e_{11})
\end{aligned}$$

The two factors $I(d_1,\ldots)$ and $I(d_2,\ldots)$ have been denoted by I_1 and I_2, respectively. Subsums and producs have been denoted by the μ on the way. They are called *messages* passed between the neighboring nodes on the factor graph. The end result is a non-normalized distribution which can be easily normalized by computing the sum over the values of e_{11} and normalizing. It can now be checked easily that when the same kinds of messages are progagated in the reverse direction, each node receives enough information to compute its marginal.

In practice, this message passing can be taken care of by an inference engine, to which we supply the factors.

The sum-product algorithm gives exact marginals for tree-structured (non-loopy) graphs, after one pass forward and back, so that messages have passed from each node to each other. For loopy graphs the exactly same algorithm can be used as an approximate technique, by passing messages iteratively and hoping that the messages converge.

So far the factor graph is a generative model for the PPI graph given priors on the degree distributions, and it can be used to infer the margins for the variables

e_{ij}, giving probabilities for the edges existing. What is even more interesting is to assume that each interaction may come from any one of a set of graphs, and to "untangle" the different graphs. The idea is that one of the graphs is the interesting one and the others represent different kinds of observation noise.

Assuming the different tangled networks are independent but that they interact at the level of producing the measurements, the joint distribution of all networks is

$$P(X, E^1, D^1, \ldots, E^H, D^H) = \prod_{ij} P(x_{ij}|e^1_{ij}, \ldots, e^H_{ij}) \prod_h P(E^h, D^h) \ ,$$

where the different networks have been indexed by superscripts.

For practical application with two networks we need to define and optimize the (edge-specific) likelihoods $P_{ij}(x_{ij}|e^1_{ij}, e^2_{ij})$, which are in the original paper parametererized with Bernoulli distributions, which are network and data source-specific but otherwise global (only one parameter per network and data source). The parameters were learned to maximize the likelihood of the noise model (without taking into account the rest of the network). The other "parameters" to be optimized are the potentials of the degree priors for each network. The potentials are set to reflect the empirical degree distributions as closely as possible, by computing them with kernel density estimates over a learning data set.

The remaining difficult question is how should the learning data set be constructed, such that the model would learn to untangle the real interactions in one network and measurement noise in the other. "Ground truth" needs to be known for some of the data, such that in the parameter learning phase E^1 will contain the true edges and E^2 the false positives.

The results could probably be improved a lot by taking into account known biases towards high-abundance proteins etc, but we need to skip those developments here.

4 Latent structure in networks

In the previous section the PPI network was modeled by a very large set of latent variables e_{ij}, one for each interaction. Modeling was made feasible by constraining the solution space though priors on the degree distribution, which in turn constrain the possible configurations of the latent variables e_{ij}. This is a very viable approach if the constraints are realistic and the computation feasible.

We will next discuss a different kind of generative approach, where the goal is to explain the interaction network with a much smaller set of latent variables, slightly resembling the generators in model-based clustering. The generators are embedded into a hierarchical Bayesian framework. It is too early to tell which of the approaches is more useful, or more specifically, which kinds of applications each is good for. A guess is that the untangling methods would be useful at least as preprocessing methods for other algorithms, whereas the clustering-type networks, equipped with well-thought assumptions about the model structure, might be good for generating hypotheses on new pathways and protein complexes.

Assume, for simplicity, that the interaction measurements are binary, $x_{ij} \in \{0, 1\}$. Assume that there are K underlying groups or classes to which the proteins may belong; to different groups in different situations. The potential for interactions between two proteins is assumed to depend only on the groups they belong to; if i belongs to group z_i and j to group z_j, $p(x_{ij}) = \eta_{z_i,z_j}^{x_{ij}}(1 - \eta_{z_i,z_j})^{x_{ij}}$. The probability of a protein i to belonging to the group z is θ_{iz}. The protein may belong to several groups; the group is sampled independently from the multinomial distribution θ_i for each interaction.

To allow a hierarchical Bayesian treatment, the protein-specific multinomials are equipped with a conjugate prior (Dirichlet). In summary, the model assumes that the protein-protein interactions have been generated as follows [1] (simplifying a bit):

1. For each protein i, sample $\theta_i \sim \mathit{Dirichlet}(\alpha)$
2. For each interaction i, j:
   (a) Sample group of i: $z_{ij} \sim \mathit{Multinomial}(\theta_i, 1)$
   (b) Sample group of j: $z'_{ji} \sim \mathit{Multinomial}(\theta_j, 1)$
   (c) Sample $x_{ij} \sim \mathit{Bernoulli}(\eta_{z_{ij},z'_{ji}})$

The joint density given the hyperparameters here is

$$p(\{x_{ij}\}, \theta, \mathbf{z}|\alpha, \eta) = [\prod_i p(\theta_i|\alpha)] \prod_{ij} p(z_{ij}|\theta_i)p(z'_{ji}|\theta_j)p(x_{ij}|z_{ij}, z'_{ji}, \eta_{ij}) \ .$$

To infer links in this model we would need to estimate the η and the hyperparameters α (or assign priors and integrate over them), and to integrate the joint density over the latent parameters $\mathbf{z}$. This would give estimates for the posterior distribution of the group memberships θ. Exact inference is intractable so variational approximations were applied in the original paper.

An even simpler and hence computationally attractive alternative is to assume that the proteins belong to groups or "communities", which may be tighter or looser but if two proteins belong to the same community they are more likely to interact. Again, a protein may belong to different communities in different situations. This can be formulated as the *interactions* belonging to groups, such that each interaction is generated by choosing a group and then two proteins in the group to interact. The generative process goes as follows [3]:

1. Sample the "prior probabilities" for choosing each group: $\theta \sim \mathit{Dirichlet}(\alpha)$
2. For each group z, sample the membership distribution of proteins: $m_z \sim \mathit{Dirichlet}(\beta)$

3. For each interaction:

   (a) Sample group $z \sim Multinomial(\theta, 1)$

   (b) Sample proteins for the interaction: $i \sim Multinomial(m_z, 1)$; $j \sim Multinomial(m_z, 1)$

   (c) Set x_{ij} to 1.

The nice property of this model is that inference on the z can be made very rapidly using collapsed Gibbs sampling. It means that all parameters of the model are integrated out, leaving only the latent memberships z, hyperparameterized by α and β which govern how sharply the edges and nodes belong to the groups, respectively.

A Dirichlet process prior can be easily incorporated into the sampling, to avoid having to explicitly choose the number of clusters. Let's skip all details on the sampling.

In conclusion, these generative models are expected to be useful for generating hypotheses on pathways and protein complexes, but it is too early to tell how useful. Anyway, they have the attractive property that they explicate the underlying distributional assumptions. Now that the assumptions are explicit, alternatives can be suggested, formulated, and the ability of the different models to explain a given data set compared.

References

[1] Edoardo M. Airoldi, David M. Blei, Stephen E. Fienberg, and Eric P. Xing. Mixed membership stochastic blockmodels. arXiv:0705.4485v1, 2007.

[2] Quaid D. Morris, Brendan J. Frey, and Christopher J. Paige. Denoising and untangling graphs using degree priors. In *NIPS 16*. 2003.

[3] Janne Sinkkonen, Janne Aukia, and Samuel Kaski. Inferring vertex properties from topology in large networks. In *MLG'07, the 5th International Workshop on Mining and Learning with Graphs*. 2007.