6.047 / 6.878 Computational Biology: Genomes, Networks, Evolution
Fall 2008

6.047/6.878 Fall 2008
Lecture 24: Module Networks

Aviv Regev

1 Introduction

Biological systems are generally incredibly complex, consisting of a huge number of interacting parts. Consider, for example, the task of understanding the structure of the genetic regulatory system. The dependence among expression levels of all the genes in a cell is so complicated that it makes any sort of detailed understanding almost impossible. The diagram for the functional network of yeast genes looks more like a hairball than anything else! Therefore, for a high-level, conceptual understanding of how transcription is controlled, it is useful to talk about higher-level objects rather than individual gene expressions. Often such conceptually simpler networks are drawn by hand by biologists. Is there a systematic way to accomplish this?

To this end, we define the notion of a *module*, which is a set of biological entities that act collectively to perform an identifiable and distinct function. A module should act as an entity with only weak linkage to the rest of the system. Some examples of modules are metabolic pathways, molecular machines and complexes for the function of signal transduction. Our focus here is on *regulatory modules*, modules in the genome that perform a distinct function. Regulatory modules are co-expressed, co-evolved and regulated by the same set of transcription factors. For example, an operon, which consists of genes coded next to each other and constitutes a functional unit of transcription, is a regulatory module. Another paradigmatic example is the ribosomal module.

Using modules as the basic units, one could now construct a *regulatory module network* where each node is a regulatory module and where interdependencies between modules are well-defined since all the genes in a module are co-expressed and co-regulated. The module network captures redundancies and structure not easily represented in a complete regulatory network.

Now, module networks can be described in several ways. They can de described in terms of their functional behavior as a whole; this is the simplest approach but sometimes suffices. Or it can be described as a logical program. This is the approach we will take in the following and describe in detail. Or it can described in terms of the low-level chemistry that determines the dynamics of the network. This is hardest to accomplish but sometimes necessary.

2 Reconstructing Module Networks: Yeasts

Work by Pe'er, Segal, Koller and Friedman shows how module networks might be reconstructed from gene expression data and a set of known candidate regulatory genes.

We first describe the action of regulators on gene expression by a logical program, represented as a decision tree. Each node in the tree consists of a regulatory gene. If the gene is upregulated, one of the two outgoing edges from the node is followed; otherwise, the other edge is followed. Each leaf of the decision tree is a regulation context, that is, a configuration of the regulator genes, determined by the path from the root to that leaf.

A regulation context determines[1] the gene expression probabilistically. For example, given that regulator gene Y_1 is upregulated and regulator gene Y_2 is downregulated, the expression of gene X might be a normal probability distribution centered around some value. These dependencies of gene expression on regulation contexts can be captured by a Bayesian network. Therefore, by applying standard algorithms to learn

[1] We will return to this assumption later. Of course, in nature, the target gene expression is not only a function of the regulatory gene expressions but also activities occurring during translation, transport, etc.

Bayesian networks, we can learn the genetic regulatory netowrk for yeast. But we come back to our starting question. How does one make sense of the structure of this hairball network?

This is where we use the idea of modules. Note that genes in the same module have the same dependency on the regulatory context, because they are, by definition, coexpressed and coregulated. So, we can combine genes in the same module into one node and simplify the network. Therefore, in order to learn the regulatory module network, the learning algorithm has to both partition the genes into modules as well as learn the Bayesian network between the module nodes.

We accomplish this using an EM algorithm. In the M-step, learns the best decision tree for each module, given the partition of genes into modules. In the E-stage, the gene pool is partitioned into modules, given the regulatory programs. Let us discuss this in a bit more detail. In the M-stage, given the module assignments, for a given module, the algorithm builds the regulatory program by choosing the regulator whose split best predicts behavior of module genes, creating a node on the decision tree for this regulator and then recursing on the two branches. In the E-step, the regulatory programs define a probability distribution over the expression levels for each gene. That is, for each particular gene, we can obtain a probability value that a genes expression value was obtained by a particular regulatory program. In this case, we simply assign the gene to the module that best predicts it. Each reassignment therefore is guaranteed to improve the overall predictiveness.

This algorithm was carried out for yeast cells. The results were very encouraging. The predicted module assignments resulted in genes in each module being functionally coherent. The match between regulator genes and target genes was there, as well as match between regulators and cis-reg motifs. Finally, one could also verify that the predicted logic in how the regulators affected the target genes corresponded to our previous understanding. The Hap4 motif was a test case for this study.

Composing the map between genes and modules and the map between modules and regulators yields a map between genes and regulators. That is, we can use the predictions from the EM algorithm to predict regulatory intereactions. Looking at the regulatory programs then allow us to make predictions of the following kind: regulator X activates process Y under condition Z. This is the sort of logic that is buried beneath the hairball in usual networks but now can be teased out. Experiments were then conducted to verify some of these predictions. The test conducted was to find out if knockout of the regulator gene X affected the process Y under condition Z. A statistical paired t-test was done to find differently expressed genes under the knockout. To find out if the predicted modules were responding, the distribution of the differently expressed genes among the modules was looked at and then the modules were ranked according to enrichment for differently expressed genes. Finally, to test what process was being affected, the regulator-process annotation provided by the predictions were compared to the pattern of differently expressed genes. There were three test cases, $YPL230W$, $PPT1$ and $Kin82$. Other than the algorithm failing to predict the processes affected by the $Kin82$ correctly, the experimental validations generally supported the predictions made by the EM algorithm.

Finally, before we move on, the question asked in the footnote remains. Why does simple association between regulator expression and target gene expression even work; why can we ignore details of translation, etc.? Many regulators are, after all, controlled post-transcription. The explanation hinges on the presence of positive and negative feedback loops in which a signaling molecule regulates a transcription factor that, in turn, regulates the expression level of the target gene. The effect is that even if a transcription factor's activity is not directly determined by its expression level, it often reveals itself in the expression of indirect regulators (such as signalling proteins that control the transcription factor's activity) during microarray experiments. In terms of our algorithm, we can view the gene encoding a transcription factor as itself being a target gene, and then run the EM algorithm to find out the logical relationships between the signaling molecules and the secondary transcription factors. (In fact, in auto-regulatory relationships, the regulatory and target genes are identical.)

3 Module Networks in Mammals

The fact that we could successfully reconstruct module networks in yeasts is encouraging. A typical response to such a success story, though, is that the algorithms can only be applied to simple systems like yeasts and they are highly unlikely to work in more complex systems like mammals which have a lot of regulatory relationships. However, although it is true that eukaryotic regulatory networks are far more complex, the presence of such structure provides more modularity and in fact might be helpful to a module network reconstruction algorithm. In this section, we talk about work with Levine, Subramanian and Novershtern on constructing a regulatory module network for Hematopoiesis.

The data for the experiment was acquired from at least 5 donors for almost each state. There is a rich source of data for hematopoiesis already present in the form of microarray analysis of levels of RNA, and this provides a validation system (called D-MAP) for our experiments. Experiments have shown that the lineage difference information yielded by the microarray results correspond to changes in relative expression levels of most genes.

Using a list of known regulatory factors, we can run the EM algorithm as described in the previous section. The algorithm is run on 523 regulator genes and 193 target genes (which are also transcription factors). This results in a predicted module network along with regulatory programs. We can now do experiments to verify if the functional units predicted by the module partitionings indeed correspond with reality. One example is the PBX1 module. Here, several known factors were discovered automatically along with the correct regulatory logic. Also previously unknown factors and regulatory logic were predicted and verified experimentally; for instance, the role of GATA1 in the MLLT7 module. Such knowledge might be used to find a code for hematopoiesis in the future.

We also briefly discussed a module network for lymphoma, that is joint work with Monti, Leite, Shipp, Lu and Golub. One can construct an miRNA module network using the previously discussed EM algorithm and then get annotation of regulators by targets. They show the existence of combinatorial regulation, where the ACB and GCB modules behave differently with respect to each other depending upon the context.

4 Conclusion

In sum, we discussed how regulatory module networks, an abstraction of transcriptional circuits, might be reconstructed given gene expression data and a set of regulators. We applied this to yeasts and to mammals, showing that the method makes reasonable predictions and allow us to get a deeper understanding of the functionality of the regulatory network.