MICROARRAY DATA INTEGRATION AND TOP-DOWN MODELING OF GENE REGULATION

Vladimir Filkov
Department of Computer Science, University of California
One Shields Avenue, Davis CA, 95616, USA
filkov@cs.ucdavis.edu

Abstract: *Microarray technologies allow monitoring of gene activity (or expression) of thousands of genes in parallel at the same time. Although very powerful, their complexity makes the observed data very difficult to integrate across experiments. In the first part of this paper I will introduce the microarray technology and several of our bioinformatics approaches for integrating multiple gene expression data sets using combinatorial and visualization methods. In the second part of the paper I will present some thoughts towards formalizing biological models of gene regulation towards a language of transcription.*

Keywords: data integration, consensus clustering, microarrays, modeling gene regulation

1. Microarray Data Integration

1.1 Introduction

Genes and gene products regulate all of the processes in the cell, as they react to developmental or environmental events. Active (expressed) genes regulate the activity (expression) of other genes by coding for proteins that physically interact with their DNA. In that sense, all the genes in an organism are parts of a gene regulatory network that describes the activities and relationships of all the genes of that organism. The therapeutic benefits of having a blueprint of a gene network are enormous because with it the organism's responses can then be understood and even modified.

DNA microarrays are large-scale technologies that measure gene activity in organisms. Microarrays use a key property of DNA molecules, hybridization, to differentiate between and identify target DNA. Physically, they are rectangular matrices on which DNA molecules, called probes, are pre-affixed in row and column intersections. The probes are characteristic to the organism that is studied. The experimenter prepares an RNA target mixture from the cells of interest, wherein the RNA is converted to the complementary molecules, called cDNA's, which are also color-tagged for later identification. The microarray is exposed to the cDNA, and the corresponding molecules hybridize, i.e. the molecules on the array find complementary molecules in the mixture to join with and form double stranded DNA. In general, the higher the concentration is of cDNA in the target, the stronger the hybridization will be on the microarray.

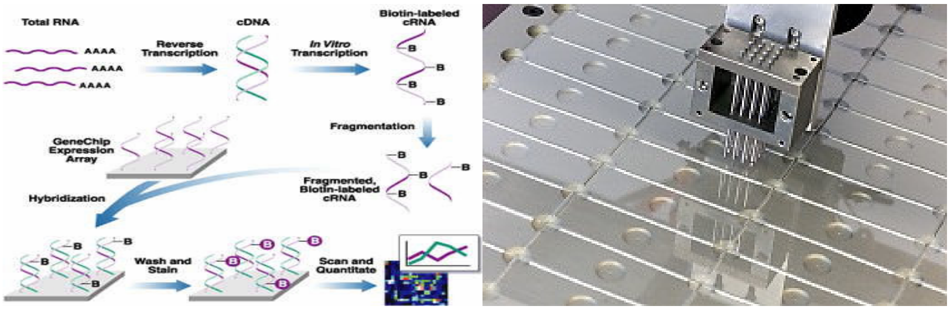
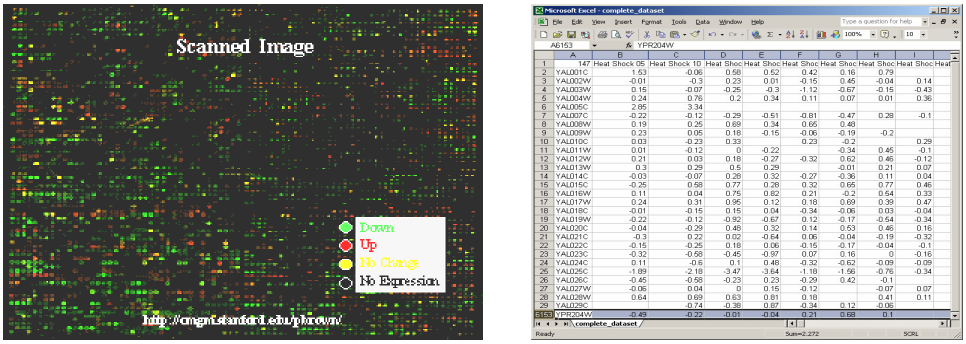


Figure 1: On the left is a diagram of the microarray process from RNA collection to its quantification (picture © SUNYSB Microarray facility); on the right is a robotic printing head which is attaching parts of gene molecules to a plate (picture © Lawrence Berkeley Lab)

The microarray is washed to remove excess molecules, and is color-scanned. The scan of a microarray is a rectangular grid of (usually) red / green (blue) / black dots. The intensity of the colored dots is proportional to the level of hybridization, which in turn indicates levels of genetic expression. The process is pictorially described in Fig 1. In particular, black indicates the level of expression (concentration or activity) of unexpressed control molecules, red may indicate expression higher than control, and green may indicate expression lower than control. As a result, with microarrays specifically tuned to a particular organism, it is possible to measure the level of concentration of any given gene, at any time, relative to a control set of DNA molecules. The technology responsible for this is a robotic printer, the head of which is shown on the right in Fig. 1, which can attach thousands of *spots* on few square inches of medium such that two spots next to each other can be very different molecules.

With microarrays one can examine the reaction of particular genes (via their expression) to environmental conditions, and establish correlation between genes and their function on a cellular level. The power of DNA microarrays, though, lies in their parallelism, since measures of gene expressions are obtained for thousands of genes at the same time. This experimental breadth in principle makes it possible to both identify differentially expressed genes across experiments, important when it is needed to find the markers for diseased vs. healthy cells, and identify genes of similar functionality when the mechanism of action of one but not other gene is known.

As you may imagine, because of their promise, everyone wants to, and does, get their hands on this technology. And once they do they start performing lots of experiments with it and generating tones of data. And since there are infinite numbers of ways to setup experiments it is unlikely that this trend will abate in the near future. Just to illustrate the wealth of data available, the worldwide repository for microarray data, the Stanford Microarray Database, SMD (http://genome-www5.stanford.edu) as of the last week of year 2004 counted more than 50000 catalogued microarray experiments over 35 organisms, and almost 1400 users over nearly 280 labs in the world!



A6153 *fx* YPR204W

	A	B	C	D	E	F	G	H	I	
1	147	Heat Shock 05	Heat Shock 10	Heat Shoc	Heat Shoc	Heat Shoc	Heat Shoc	Heat Shoc	Heat Shoc	Heat
2	YAL001C	1.53	-0.06	0.58	0.52	0.42	0.16	0.79		
3	YAL002W	-0.01	-0.3	0.23	0.01	-0.15	0.45	-0.04	0.14	
4	YAL003W	0.15	-0.07	-0.25	-0.3	-1.12	-0.67	-0.15	-0.43	
5	YAL004W	0.24	0.76	0.2	0.34	0.11	0.07	0.01	0.36	
6	YAL005C	2.85	3.34							
7	YAL007C	-0.22	-0.12	-0.29	-0.51	-0.81	-0.47	0.28	-0.1	
8	YAL008W	0.19	0.25	0.69	0.34	0.65	0.48			
9	YAL009W	0.23	0.05	0.18	-0.15	-0.06	-0.19	-0.2		
10	YAL010C	0.03	-0.23	0.33		0.23	-0.2		0.29	
11	YAL011W	0.01	-0.12	0	-0.22		-0.34	0.45	-0.1	
12	YAL012W	0.21	0.03	0.18	-0.27	-0.32	0.62	0.46	-0.12	
13	YAL013W	0.3	0.29	0.5	0.29		-0.01	0.21	0.07	
14	YAL014C	-0.03	-0.07	0.28	0.32	-0.27	-0.36	0.11	0.04	
15	YAL015C	-0.25	0.58	0.77	0.28	0.32	0.65	0.77	0.46	
16	YAL016W	0.11	0.04	0.75	0.82	0.21	-0.2	0.54	0.33	
17	YAL017W	0.24	0.31	0.95	0.12	0.18	0.69	0.39	0.47	
18	YAL018C	-0.01	-0.15	0.15	0.04	-0.34	-0.06	0.03	-0.04	
19	YAL019W	-0.22	-0.12	-0.92	-0.67	0.12	-0.17	-0.54	-0.34	
20	YAL020C	-0.04	-0.29	0.48	0.32	0.14	0.53	0.46	0.16	
21	YAL021C	-0.3	0.22	0.02	-0.64	0.06	-0.04	-0.19	-0.32	
22	YAL022C	-0.15	-0.25	0.18	0.06	-0.15	-0.17	-0.04	-0.1	
23	YAL023C	-0.32	-0.58	-0.45	-0.97	0.07	0.16	0	-0.16	
24	YAL024C	0.11	-0.6	0.1	0.48	-0.32	-0.62	-0.09	-0.09	
25	YAL025C	-1.89	-2.18	-3.47	-3.64	-1.18	-1.56	-0.76	-0.34	
26	YAL026C	-0.45	-0.58	-0.23	0.23	-0.29	0.42	-0.1		
27	YAL027W	-0.06	0.04	0	0.15	-0.12		-0.07	0.07	
28	YAL028W	0.64	0.69	0.63	0.81	0.18		0.41	0.11	
29	YAL029C		-0.74	-0.38	0.87	-0.34	0.12	-0.06		
6153	YPR204W	-0.49	-0.22	-0.01	-0.04	0.21	0.68	0.1		

complete_dataset

Ready Sum=2.272 SCRL

Figure 2: Scanned microarray image © P. Brown's lab at Stanford (left), and a spreadsheet snapshot of a typical microarray data set of hundreds of yeast microarrays (genes are rows, experiments columns)

With the exploding volume of microarray experiments comes increasing interest in mining repositories of such data. But meaningfully combining results from varied experiments on an equal basis is a challenging task. This makes the case for data integration. Especially when dealing with large-scale data, integration becomes not just useful but very necessary.

There have been several previous attempts toward general integration of biological data sets in the computational biology community. Marcotte et al.[1], for example, give a combined algorithm for protein function prediction based on microarray and phylogeny data, by classifying the genes of the two different data sets separately, and then combining the gene pairwise information into a single data set. Pavlidis et al.[2] use a Support Vector Machine algorithm on similar data sets to predict gene functional classification. Both methods need hand tuning with any particular type of data both prior and during the integration for best results. Ad hoc approaches to data integration are difficult to formulate and justify, and do not readily scale to large numbers of diverse sources, since different experimental technologies have different types and levels of systematic error. In such an environment, it is not always clear that the integrated product will be more informative than any independent source.

1.2 Data Integration by Consensus Clustering

Clustering has shown to be an extremely useful technique for microarray data analysis, especially if very little is known about that data a priori ("fishing expeditions"). That is because genes grouped by their expression profiles often are functionally related. (see Fig. 3) There are many different clustering techniques: hierarchical, k-means, topology based, fuzzy methods based, etc. For more on clustering microarray data one can consult for example [4] or my lectures at "http://www.cs.ucdavis.edu/~filkov/classes/289a-W03/l6.pdf".

This is where we came in. Our idea was simply that if clustering is so popular then people use it all the time and they must have clustered the same genes of their favorite organism (say yeast, or human) many times over. Intuitively, different

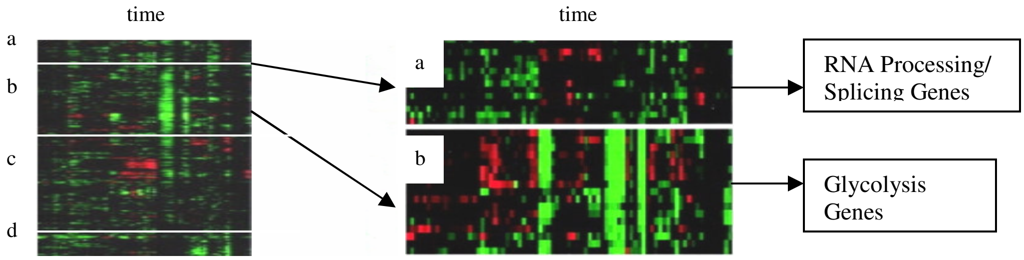


Figure 3: Shown on the left are clustered expression profiles of a subset of genes of yeast (shown are hundreds out of the ~6000 genes), observed on microarrays at different points in time. The visual clusters actually correspond to functional clusters, as illustrated by the first two groups [3].

clusterings of the same genes could be used to tell us more about the groups that the genes co-belong to than the individual clusterings themselves. So, our goal became to develop methods based on the various, source-specific, clusterings of the data (or the *meta-data*) to both (a) provide an integrated view of the data and (b) eliminate misclassifications due to errors in the individual data sets. Both of these goals are naturally addressed by the theoretical problem of *consensus clustering*, which goal is to find a representative or *consensus* clustering that describes well a set of given clusterings.

Mathematically, clusterings are just set partitions, and formally, the Consensus Clustering problem (CC) can be written as:

> ***CC:*** *Given k set-partitions $P_1, P_2, \ldots, P_k$ and a distance measure d(.) on them, find a consensus partition C that minimizes* $S = \sum d(P_i, C)$

As our distance measure we chose the *symmetric difference distance,* which counts pairs of simultaneously co-clustered (or simultaneously not co-clustered) elements in both clusterings (i.e. partitions). It is possible to show that with this distance measure the consensus clustering problem above becomes NP-complete [5], so exact solutions are intractable, especially since our data sets are very big. Instead, we developed several different heuristics for solving CC, and chose the "best" out of them for our system (by best we mean one satisfying some theoretically provable bounds as well as outperforming the other heuristics in our tests). In that heuristic the space of solutions is traversed by moving between set partitions by simple single element swaps between clusters, while deciding whether a move is beneficial based on a simulated annealing optimizer [5]. The heuristic could handle hundreds of genes and hundreds of clusterings (i.e. set partitions) in real time (seconds) and thousands of genes in minutes.

We implemented a software system, CONPAS (CONsensus Partitioning System), around this heuristic, and tested it around many different data sets of genes and clusterings on yeast. In addition to providing an integrated view of clusterings of data, CONPAS also is very successful at eliminating misclassifications due to errors in the individual clusterings, as can be seen in Fig 4, thus addressing both

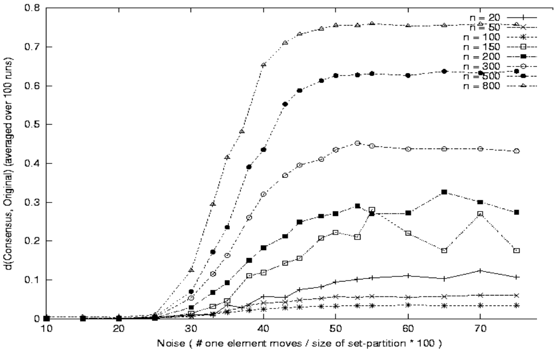


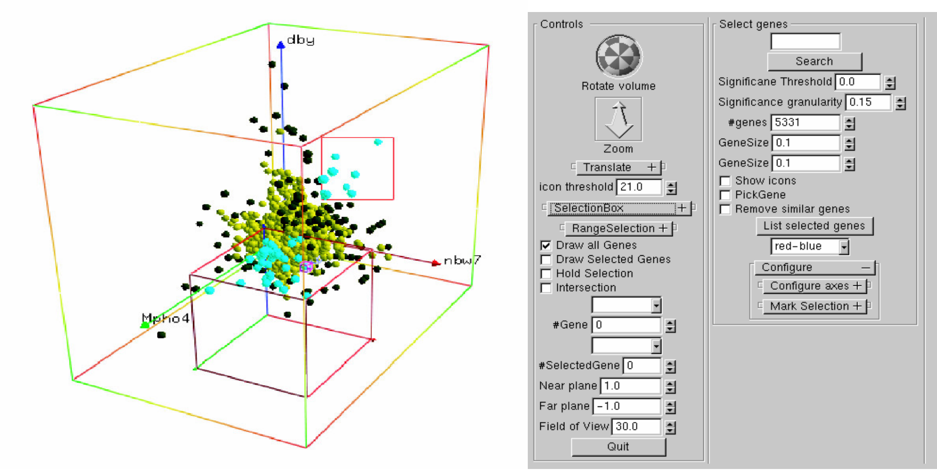
Figure 4: This plot shows the benefit of having multiple clustering from different sources. Even if the noise is as high as 20-25% in the individual clusterings (x-axis) CONPAS can resolve if genes are actually co-clustered with very low error (y-axis). The experiment was performed by deriving noisy clusterings from an original clustering that later CONPAS was trying to match from the noisy ones. The error of the match was d(.)

the original goals. As an additional benefit, the average sum of distances to the consensus clustering, when properly corrected statistically, happens to be a good indicator of the *goodness of the data set integration,* thus providing a confidence coefficient, and is of independent interest.

We have further refined and extended the original consensus clustering concept into many different directions, e.g. imputing missing data, bi-clustering, weighting experiments of different importance, etc., and have proven it to be a very rich and extensible data integration platform [6].

1.3 Data Integration by Visualization with GeneBox

Early microarray experiments were overall very simple, focusing only on gross differential expression under test conditions, many even lacking repeats. Consequently, many early microarray data analysis tools were geared towards finding genes that are simply differentially or similarly expressed under the test (versus control) conditions. The use of any such data analysis tool requires the researcher to appropriately tune these parameters for their data set, in order to avoid arbitrary results in terms of the number of data clusters, size, density, etc. Today though, microarray experiments are routinely performed under multiple experimental conditions, on multiple test samples, and for multiple controls. Such versatile data allows more interesting questions to be asked, like which genes are expressed similarly under some but not all experiments. Since it is actually the differential expression under some, but conceivably not all, experimental conditions that sets the genes apart functionally, the ability to consider such questions is ultimately very important. For example, in an experiment with two genotypes and two time points, a scientist might be interested in finding genes that are similarly expressed at the first time point in both genotypes but expressed differently at the other point in the genotypes. Exploring such data then, can benefit from versatile and interactive visualization tools that bring the problem of data mining and analysis closer to the individual researcher in the field, by allowing real-time visual data manipulation.



icon threshold 21.0

☑ Draw all Genes

□ Draw Selected Genes

□ Hold Selection

□ Intersection

#Gene 0

#SelectedGene 0

Near plane 1.0

Far plane -1.0

Field of View 30.0

Significane Threshold 0.0

Significance granularity 0.15

#genes 5331

GeneSize 0.1

GeneSize 0.1

□ Show icons

□ PickGene

□ Remove similar genes

red-blue

Figure 5: The GeneBox graphical interface is shown on the left, with the genes rendered as points in 3D, and selection tools shown as cubes and squares. The box is fully interactive with some of the parameters manually changeable in the right panel

To that end we developed GeneBox [7], a general-purpose 3-D visualization tool for multi-variable microarray gene expression data. GeneBox was designed to help scientists answer complex queries through interactive visual exploration of microarray data sets. It works with microarray data coming from multiple chips, e.g. genotypes under multiple experimental conditions. GeneBox is built around a few core methods for microarray data analysis: (1) data normalization methods, as data comes from multiple microarray chips; (2) statistical differential expression inference; and (3) statistical significance of differentially expressed genes in three dimensions. Its real strength, however, comes from the multidimensional visual interface, necessary to represent the multi-variable microarray data, coupled with interactive functionalities and variety of user controls to customize the output.

The basic setup of GeneBox is a unit cube in space, rendered in perspective, in which the gene shapes are visualized. Each gene is rendered as a 3-D icon in GeneBox (Fig 5). Its location and color is determined by its differential expression. GeneBox maps differential expression of a gene under three different experimental conditions to coordinates along three axes in 3-D space. The differential expression is calculated using state of the art statistical methods. Color is used to indicate the significance of the genes differential expression.

The interactive setup of GeneBox is the key ingredient to its success with the life scientists. Compared to the CONPAS system, GeneBox is much more hands-on and although it requires much more training it certainly requires much less explanation of the results. Our experience is that the user niches for the two systems have overall been different, with GeneBox becoming a life scientist favorite.

2. Modeling Gene Regulation

2.1 Introduction

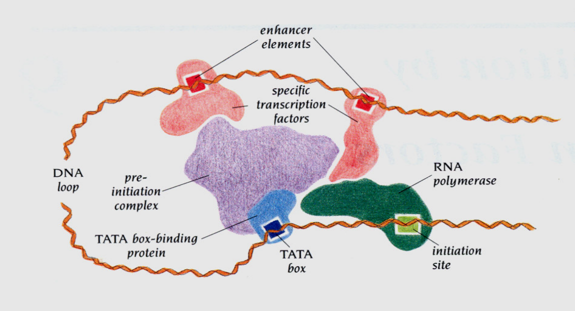
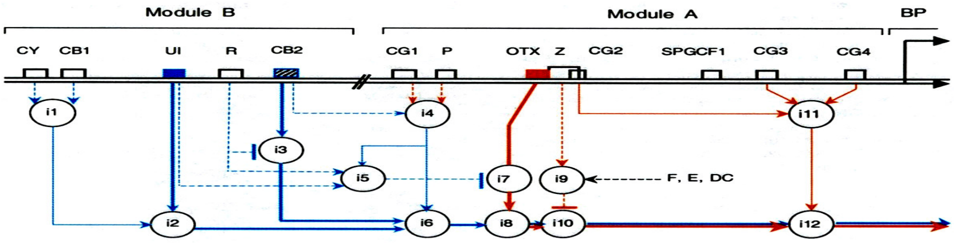


Figure 6: Trans-factors binding to cis regions of a hypothetical gene. The DNA is the red strand and the globs are the proteins. The binding sites are marked with rectangles. The gene starts at the initiation site at continues to the right. Some of the TFs have a DNA looping role, i.e. they bring segments of the DNA in proximity for the other TFs to interact with © McMIllan pubs

The reason we have to use methods to integrate genomics data is because we simply do not know how our cells work on the genome level. Although we do know that at the most basic biological level the genetic information is passed from the DNA (through the process of transcription) to the mRNA and out into the cytoplasm to the proteins through translation, we still have only very vague understanding of how genes and proteins are connected and interact into what are called *gene regulatory networks* to sustain life. Although there clearly exist needs for good formal and verifiable models of genetic networks, there is very little understanding out there about what a good model should offer, and what a good data set on which it should be measured should be. In this section I would like to offer some introduction to modeling transcriptional regulation and thoughts on properties that formal models for transcriptional regulation should have, based on some recent work by E. Davidson and colleagues on developmental regulation in sea urchin.

2.2 Genes and the Logic of Transcription

Genes are heritable pieces of DNA representing only 2-3% of the DNA in the cell nucleus. Like the rest of the DNA they are pretty dormant, until the process of transcription starts. Then, proteins called Transcription Factors (TFs), bind to DNA regions near the genes (called cis-regions) which attract a big molecular machinery called RNA polymerase, see Fig 6, which in a lock-in-key fashion recognizes the TFs signatures and starts copying letter for letter the gene into a complementary molecule called mRNA, which is the active version of the gene. As long as the TFs are attached, RNA polymerases will copy the gene, and the gene is considered active. The TFs don’t just bind anywhere; they are very picky and specific: given their chosen site they bind to it, while others will rarely do. The binding sites are not always available as the DNA is not always untangled. So, the activity of a gene correlates with the availability of the binding sites and the concentration of the TFs.



if CY & CB1	i1 = 1	if i5 = 0	i7 = OTX(t)
else	i1 = 0.5	else	i7 = 0
	i2 = i1 • UI(t)		i8 = i6 + i7
if R	i3 = CB2(t)	if (F or E or DC) & Z	i9 = 1
else	i3 = k • CB2(t) (1<k<2)	else	i9 = 0
if P & CG1 & CB2	i4 = 2	if i9=1	i10 = 0
else	i4 = 0	else	i10 = i8
if UI (t)>threshold & R & i4≠0	i5 = 1	if (CG2 & CG3 & CG4)	i11 = 2
else	i5 = 0	else	i11 = 1
	i6 = i4 • (i2+i3)		i12 = i11•i10

Figure 7: The endo16 cis-region information processing logis © E. Davidson and Science magazine. The final transcription is a linear combination of three input signals, the colored boxes, with the constants depending on the occupancy of the other sites [9]

The question then, is, given sufficient concentrations of all possible TFs how will different combinations of binding sites react to them? In other words, if the cis-regions were treated as information processing units then what types of signals can they process? Eric Davidson at Caltech and his colleagues asked similar questions to these but in a biological setting. In the past 30 years they have made very descent attempts at answering some of them [8]. An excellent example of their work is the elucidation of the processing logic of the cis-region of the endo16 gene in sea urchin [9] (see Fig. 7). There they have tried to explain the effects of elimination of a part of the system on the system behavior as a whole. It is interesting to follow their scheme computationally, and play out different input/output scenarios.

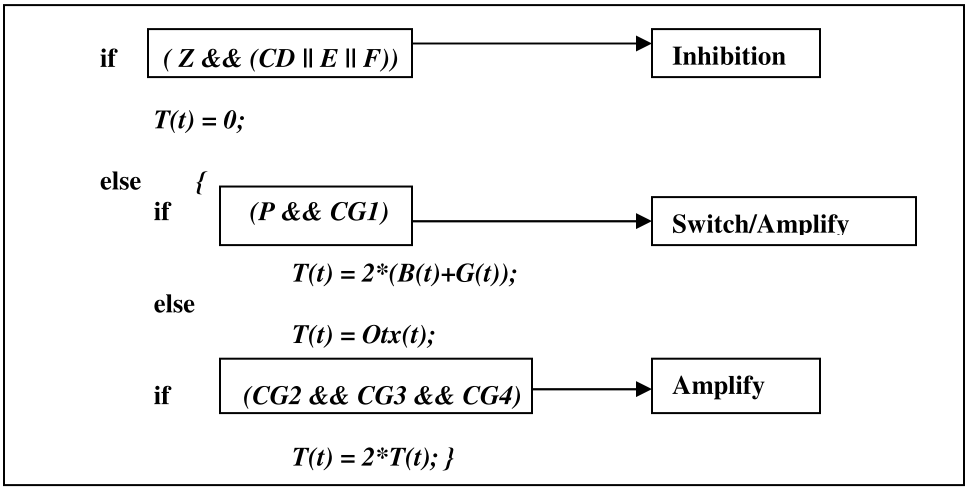
2.3 Reverse Engineering Nature

The results of Davidson's and colleagues are a very good starting point for a realistic model of transcription. Their most important result is an empirical example of reverse engineering of a cis-region's information processing logic. In those lines, if a cis-region is considered a black box that accepts input (TF-DNA binding) and produces corresponding output, than that black box can be reverse engineered.

But why did they succeed, what did we learn from them, and how can we as computational scientists generalize their methods? The answer to all three questions may be the same: because biological systems at functional level are modular [10].

2.4 Towards a Language of Transcription

Mmodularity allows us not just to elucidate the logic of endo16 in fewer experiments, although that is certainly the case. More importantly it helps us to scale the phenomenon of transcriptional regulation so that we can think of it not in terms of biochemistry but in terms of abstract processes and ideas, and in terms of an expressive language. Fig 8 shows an example. It is Fig 7 rewritten in a C-like language, with added modular semantics. The underlying meaning is that there are building blocks (i.e. amplify, inhibit, switch) that transcriptional regulation reuses to make genes active and to make networks connected. If, perhaps there are a finite number of them, then there might be a language of transcription and gene regulation that is very much like the programming languages that we know, written in the genetic codes of animals.



```
if    ( Z && (CD || E || F))
      T(t) = 0;
else    {
      if    (P && CG1)
                T(t) = 2*(B(t)+G(t));
      else
                T(t) = Otx(t);
      if    (CG2 && CG3 && CG4)
                T(t) = 2*T(t); }
```

Figure 8: The endo16 logic rewritten in a C-like code. The logical statements are to be interpreted TRUE iff the corresponding binding site is present and bound. T(t) is the transcription of endo16 at time t. The boxes on the right indicate the modular actions of the logical combinations of those combinations of binding sites, and are the same wherever those binding sites occur together.

Acknowledgments

Various parts of this paper are excerpts from longer time, collaborative work that I have performed over the past few years with my colleagues Steven Skiena at SUNY Stony Brook, Sorin Istrail at Celera Genomics, Eric Davidson at Caltech, Nameeta Shah at UC Davis, and Bernd Hamann at UC Davis. Where noted, the original authors or publishers reserve the copyrights of the pictures I have used in this paper.

References

1. Marcotte, M., et al., “A combined algorithm for genome wide prediction of protein function,” *Nature*, v. 402, 83-86, 1999

2. Pavlidis, P., et al., “Learning gene functional classification from multiple data types,” *Journal of Computational Biology*, v.9, 401-411, 2002

3. Eisen, M., et al., “Cluster analysis and display of genome-wide expression patterns,” *PNAS*, v. 95, 14863-14868, 1998

4. Speed, T., *Statistical Analysis of Microarrays Data*, Chapman & Hall/CRC, 2003

5. Filkov, V., Skiena, S., “Integrating Microarray Data by Consensus Clustering.” *Int. Conference on Tools with Artificial Intelligence* 2003, 418-425, 2003

6. Filkov, V., Steven Skiena, “Heterogeneous Data Integration with the Consensus Clustering Formalism.” *DILS* 2004, 110-123, 2004

7. Shah, N., Filkov, V., Hamann, B., Kenneth I. Joy, “GeneBox: Interactive Visualization of Microarray Data Sets”, *METMBS* 2003, 10-16, 2003

8. Davidson, E. H., *Genomic Regulatory Systems: Development and Evolution*, Academic Press, San Diego, 2001

9. Davidson, E. H. et al., “A genomic regulatory network for development,” *Science*, 295, 1669-1678, 2002

10. Filkov, V., Istrail, S., “Inferring Gene Transcription Networks: The Davidson Model,” *Genome Informatics* 13, 236-239, 2002