Algorithms for Molecular Biology **Fall Semester, 2001**

Lecture 11: January 17, 2002

Lecturer: Roded Sharan and Amos Tanay *Scribe: Amos Tanay and Eyal Zach*[1]

The following lecture briefly overviews selected topics from the general domain of "functional genomics" (with one deviation to SBH - sequencing by hybridization). Functional Genomics is at the front line of todays computational biology, dealing with the understanding of biological systems when the DNA sequence is already known.

Computational biology (or bioinformatics) emerged as a critical component in the scientific machinery assembled to reconstruct large genomes DNA sequences. In the post-genomic era, when the complete DNA sequence of many organisms is available, we begin using computational techniques to answer the most fundamental questions of biology : what is the function of each and any gene and gene product in the cell, how does these factors react to different conditions and how do they interact to process information and create a **biological system.**

In what follows we shall overview some techniques and describe some of the important problems we study today. We begin by an introduction to DNA chips which are part of the technological revolution enabling post-genomic high throughput biology. We then turn to one of the basic problems in analyzing DNA chips experiments, clustering gene expression data, provide some terminology and notion and describe CLICK, a clustering algorithm developed for gene expression analysis. Following are some examples on applications of clustering to study concrete biological questions.

In the last part of this lecture we introduce gene networks and overview the difficulties and possibilities in the computational study of biological systems as a whole. We primarily outline the problems and present the more promising directions in contemporary research.

11.1 Introduction

11.1.1 Functional Genomics

Having (almost) reached the end of Human Genome Project, the question that needs to be asked is: "What's next?". The complete sequencing of the Human Genome is an immense task, which is now declared as being completed (and is actually nearing completion). While much work remains to be done even there, the existence of the full DNA sequence of an organism represents only the beginning of a long journey toward the understanding of living organisms. "functional genomics" is the study of the functionality of specific genes, their

[1] Based on a scribe by Ronny Morad and Tal Moran, Fall 2000.

relations to diseases, their associated proteins and their participation in biological processes. Our current knowledge on gene function is extremely limited, with most of the genes completely uncharacterized and only very few with exact known function. Most of the knowledge gained so far in this area is the result of painstaking research of specific genes and proteins, based on complex biological experiments and homologies to known genes in other species. This "Reductionist" approach to functional genomics is hypothesis driven - we proceed by suggesting a hypothesis and designing an experiment to check its correctness. However, the complexity of living organisms make the challenge of fully understand complex biology unachievable using these methods, and a new paradigm, holistic and high throughput is emerging instead.

High throughput biology is based on our ability to collect large amount of information on the cell such that we can use the information to generate hypotheses and not only to test them. The technological revolution making this possible is combining robotics, computing and material sciences. We can, today, use **DNA chips** to measure the mRNA levels of an entire genome at a single experiment, or apply protein chips to do similar studies with proteins. We can perform assays to capture thousands of interactions among proteins recreating, in a single experiment, the work for which a whole laboratory would work for only five years ago. Having the ability to collect much information still does not imply we can actually use it, and the methodology of biological data analysis, hypothesis generation and testing is at the core of the computational functional genomics domain.

11.1.2 Gene Expression

The first successful high-throughput biological experimentation method is enabling the measurement of gene expression. Using one of several methods we can measure the mRNA level of each gene present in a cell in a given condition. This information is very valuable since we know that one of the most important regulatory mechanisms in the cell is transcription control, which modifies the expression (transcription) rate of each gene to perform complex coordinated tasks and adapt protein concentrations to a changing environment.

Biologically speaking, we can identify a large group of genes involved in a specific process (e.g., heat shock) by performing a high throughput experiment in which a cell line (or cell colony) is transformed into this condition (e.g., changed temperature) and we measure the mRNA levels of all genes in the following few hours. Having measured transcription in the entire genome, we should now have the complete list of genes whose transcription level is being regulated by our specific condition.

Being the first successful high throughput method (apart from sequencing) we should note that gene expression is biologically important but by no means tells the whole story, since many other mechanisms regulate biological activity and are not visible in the transcription level (**post translational regulation** is an important class of such mechanisms). However, using transcription profiling, biologists have already expanded greatly our understanding of important biological processes (including cell cycle [19], cancer [3], metabolism [10] and more).

11.2 DNA Chips/Microarrays

11.2.1 What is Hybridization?

As we have already seen, under normal conditions, the DNA molecule is composed of two strands. These two strands are connected by hydrogen bonds, and together form the well-known double helix structure.

When a solution containing DNA is heated, these hydrogen bonds disappear, and the two strands drift apart. This single-stranded DNA is called *denatured DNA* (or *single-stranded DNA*). When the solution is cooled, hydrogen bonds form between matching bases in the strands. These bonds are formed in places where a match (or at least a partial match) exists. If these bonds begin to form in corresponding parts of two strands, they will quickly completely join and the double-helix will reappear. However, this is not guaranteed to happen. Bonds can form even between strands of different DNA molecules or strands of different length.

Consider a heated solution of some *target* DNA molecule. Let us take short single-stranded chains of nucleotides, called *oligonucleotides* (or *oligos* for short), that we have synthesized and add them to the solution. Each oligo is a known nucleotide sequence between 10 and 12 bases long. Now, when the solution is cooled, the oligos will stick to parts of the target that contain a DNA sequence complementary to that of the oligo. The resulting composition is called *hybrid DNA*. Each oligo thus probes for the presence of its complementary sequence, and indeed oligos are called *probes*

Many biological techniques are based on hybridization. For instance, consider the following situation: Given human DNA and oligos obtained from coding regions of mouse DNA (each one several hundred bases long). We can now create a hybrid of the two. Since homology between these DNAs exists mostly in the coding regions, we can use the hybridization to infer where human coding regions are. If the oligos are tagged, either with fluorescent dye or radioactive label, one can detect where the oligos hybridized with the DNA, and thus infer what the coding regions are (or at least, which regions deserve further study).

11.2.2 DNA Chips

Hybridization experiments were traditionally performed using page-sized filters, with each filter having about 10 bands displaying whether hybridization occurred or not. With the development of miniaturization, small chips, containing an array of 100 micron dots are able to perform simultaneously thousands of hybridization experiments. This can be done using

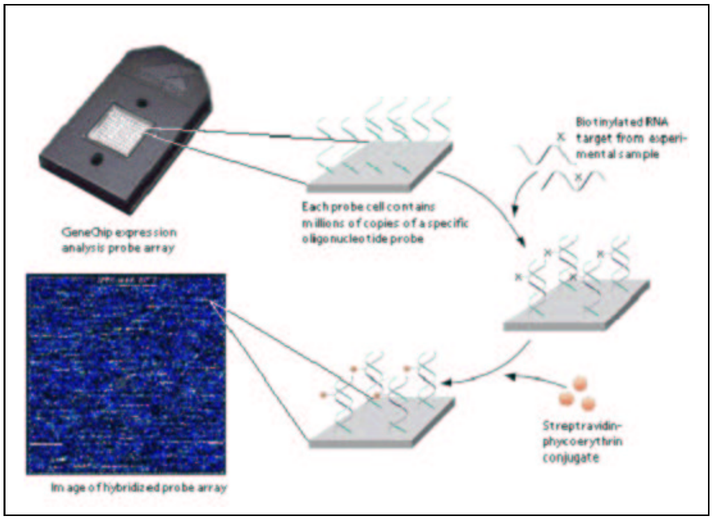

GeneChip expression analysis probe array
Each probe cell contains millions of copies of a specific oligonucleotide probe
Biotinylated RNA target from experimental sample
Streptravidin-phycoerythrin conjugate
Image of hybridized probe array

Figure 11.1: Source: [1]. A typical experiment with an oligonucleotide chip. Labeled RNA molecules are applied to the probes on the chip, creating a fluorescent spot where hybridization has occurred.

several methods.

Oligonucleotide Arrays

The basic idea in these chips, developed (and patented) by a company named Affymetrix, is to generate probes that would capture each coding region as specifically as possible. The length of the oligos used depends on the application, but they are usually no longer than 25 bases. Since the oligos are short, the density of these chips is very high, for instance, a chip that of 1cm by 1cm can easily contain 100,000 oligos.

The chip is designed as a matrix of hybridization sites, each composed of a selection of coding oligos and control oligos. Coding oligos corresponds to perfect matches of known targets, controls probes are almost perfect matches, with one perturbed base. When reading the chip, hybridization levels at controls is subtracted from the level of match probes providing means to overcome false positives. Actual chip designs uses 10 matches and 10 mismatches probes for each target (gene). This method is very successful, and Affymetrix manufactures today chips for the entire human or yeast genomes.

Figure 11.1 shows a schematic description of an hybridization experiment using oligo chips.

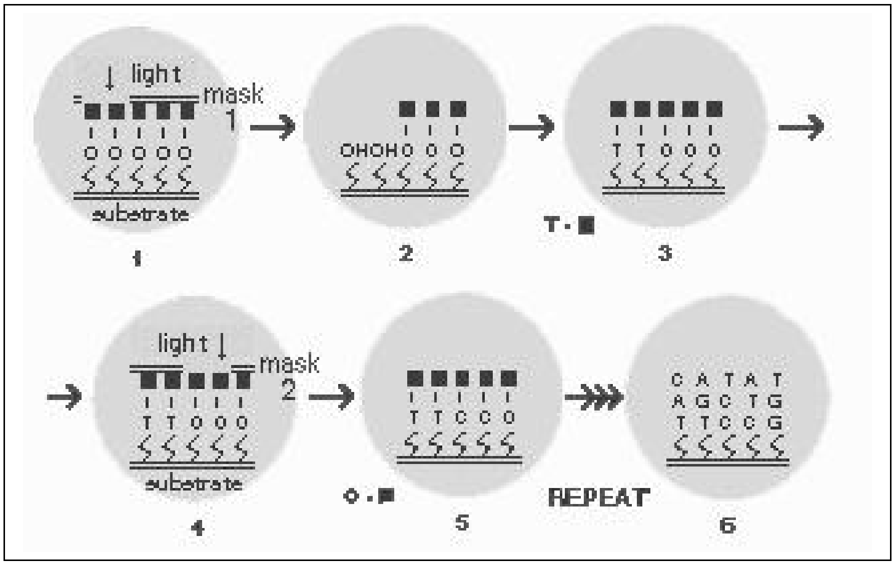


Figure 11.2: Source: [1]. Creating DNA chips. 1 and 2: The light removes the terminator from the chains not covered by the mask, creating hydrogen bonds instead. 3: Bonds are formed with a nucleotide base. 4 through 6: The process is repeated with a different base.

Manufacturing Oligonucleotide Arrays

Oligonucleotide arrays are produced in a way that is similar to the way computer chips are. We start with a matrix created over a glass substrate. Each cell in the matrix contains a "chain" with appropriate chemical properties, and ending with a *terminator*, a chemical gadget that prevents chain extension.

We cover this substrate with a mask, covering some of the cells, but not others. We can then illuminate the substrate. Covered cells are unaffected. In cells that are hit by the light, the bond with the terminator is severed. If we now expose the substrate to a solution containing a nucleotide base, it will form bonds with the non-terminated chains. Thus, some of the cells will now contain this nucleotide.

The process can then be repeated with different masks, and for different nucleotides. This way we can insert a specific nucleotide to each cell of the matrix. Figure 11.2 demonstrates the production process.

cDNA Microarrays

In this approach, developed at Brown lab in Stanford, we use, instead of short oligos, a cDNA clone representing some gene. Since cDNA clones are much longer than oligos (can be thousands of nucleotides long), a successful hybridization with a clone is an almost certain match for the gene. However, due to the different structure of each clone and the fact that unknown amount of cDNA is printed at each probe, we cannot associate directly the hybridization level with transcription level and so cDNA chips experiments are limited to comparisons of a reference pool and a target pool. To perform a cDNA array experiment, we label green the reference pool, representing the standard level of expression in our model system, and label red the target culture of cells which were transformed to some condition of interest. We hybridize the mixture of reference and target pools and read a green signal in case our condition reduced expression level and red signal in case our condition increased expression level.

Oligo-Fingerprinting

This type of chips was the first to be used, and is, in a sense, the opposite to Affymetrix approach. The chips consist of a matrix, with each cell of the matrix containing target DNA. The chip is exposed to a solution containing many **identical oligos**, and hybridization occurs between matching DNA and oligos. Again, if the oligos are tagged, either with fluorescent dye or radioactive label, we can then see at each point of the matrix whether the hybridization occurred (i.e., which of the DNAs hybridized to the oligo we tested).

The chip can then be heated, separating the oligos from the DNA, and the experiment can be repeated with a different type of oligo.

Finally, we get a matrix M, with each row representing a specific target DNA from the matrix, and each column representing an oligo.

11.3 Clustering Gene Expression Data

The outcome of high throughput gene expression experiments is a matrix associating for each gene (row) and condition (column) the expression level. Expression levels can be absolute (as in Affymetrix oligo arrays) or relative (as in Brown's cDNA array). We wish identify biological meaningful phenomena from the expression matrix, which is often very large (thousands of genes and hundreds of conditions). The most popular and natural first step in this analysis is clustering of the genes or experiments. Clustering techniques are used to identify subsets of genes that behave similarly under the set of tested conditions. By clustering the data, the biologist is viewing the data in a concise way and can try to interpret it more easily. Using additional sources of information (known genes annotations or conditions details), one can try and associate each cluster with some biological semantics.

In what follow we shall describe CLICK, a clustering algorithm developed for the analysis of large gene expression data sets. We shall then discuss the problem of analyzing clustering performance and review CLICK's performance vs. other clustering approaches.

11.3.1 The CLICK Algorithm

CLICK (CLuster Identification via Connectivity Kernels) is clustering algorithm developed by Sharan and Shamir [16]. The input for CLICK is the gene expression matrix. Each row of this matrix is an "expression fingerprint" for a single gene. The columns are specific conditions under which gene expression is measured. A more formal definition is as follows:

Let $N = \{e_1, \ldots, e_n\}$ be a set of elements. Let M be an input real-valued matrix of order $n \times p$, where M_{ij} is the j-th attribute of e_i. The i-th row-vector in M is the fingerprint of e_j. For a set of elements $K \subseteq N$, we define the *fingerprint* of K to be the mean vector of the fingerprints of the members of K. One seeks to partition N into clusters (subsets), assuming some real partition exists. In such a partition, elements in the same cluster are called *mates*.

The CLICK algorithm attempts to find a partition of N into clusters, so that two criteria are satisfied: *Homogeneity* - elements inside a cluster are highly similar to each other; and *separation* - elements not inside the same cluster have low similarity to each other.

Probabilistic Assumptions

The CLICK algorithm makes the following assumptions:

1. Similarity values between mates are normally distributed with mean μ_T and variance σ_T^2.

2. Similarity values between non-mates are normally distributed with mean μ_F and variance σ_F^2.

3. $\mu_T > \mu_F$

These assumptions are justified both empirically and theoretically in some cases by the Central Limit Theorem.

The Basic CLICK Algorithm

The CLICK algorithm represents the input data as a weighted *similarity graph* $G = (V, E)$. In this graph vertices correspond to elements and edge weights are derived from the similarity values. The weight w_{ij} of an edge (i, j) reflects the probability that i and j are mates, and is set to be

$$w_{ij} = \log \frac{p_{mates} f(S_{ij} | i,j \text{ are mates})}{(1 - p_{mates}) f(S_{ij} | i,j \text{ are non-mates})}$$

where $f(S_{ij}|i,j \text{ are mates}) = f(S_{ij}|\mu_T, \sigma_T)$ is the value of the probability density function for mates at S_{ij}:

$$f(S_{ij}|i,j \text{ are mates}) = \frac{1}{\sqrt{2\pi}\sigma_T} e^{-\frac{(S_{ij}-\mu_T)^2}{2\sigma_T^2}}$$

```
Basic-CLICK($G(V, E)$)
    if ($V(G) = \{v\}$) then
        move $v$ to the singleton set $R$
    elseif ($G$ is a kernel) then
        Output $V(G)$
    else
        ($H$,$\bar{H}$, $cut$) $\leftarrow$ MinWeightCut($G$)
        Basic-CLICK($H$)
        Basic-CLICK($\bar{H}$)
    end if
end
```

Figure 11.3: The Basic-CLICK algorithm

Similarly, $f(S_{ij}|i,j$ are non-mates) is the value of the probability density function for non-mates.

The basic CLICK algorithm is defined in figure 11.3.

The idea behind the algorithm the following: given a connected graph G, we would like to decide whether $V(G)$ is a subset of some true cluster, or $V(G)$ contains elements from at least two true clusters. In the first case we say that G is *pure*. In order to make this decision we test for each cut C in G the following two hypotheses:

- H_0^C: C contains only edges between non-mates.
- H_1^C: C contains only edges between mates.

G is declared a *kernel* if H_1 is more probable for all cuts. Using the following lemma (11.1), we can simply calculate the minimum weighted cut to determine whether G is a kernel.

Lemma 11.1 *G is a kernel iff the Minimum Weight Cut of G is positive.*

Proof: Using Bayes Theorem, it can be shown that

$$W(C) = \log \frac{Pr(H_1^C|C)}{Pr(H_0^C|C)}$$

Obviously, $W(C) > 0$ iff $Pr(H_1^C|C) > Pr(H_0^C|C)$. If the minimum cut is positive, then obviously so are all the cuts. Conversely, if the minimum cut is non-positive, then for that cut $Pr(H_1^C|C) \leq Pr(H_0^C|C)$, therefore G is not a kernel. ■

Refinements

The Basic-CLICK algorithm will divide the graph into kernels and singletons. To use the algorithm for solving clustering problems, we must introduce a number of refinements:

- **Removing Negative Edges**: The MINCUT problem for a weighted graph with both positive and negative edges is NP-Complete. In order to use the efficient MINCUT algorithms we must remove the negative edges. Applying CLICK to the modified graph approximate the original problem.

- **Adoption Step**: In practice, "true" clusters are usually larger than just the kernel. To accommodate this, in the refined algorithm, kernels "adopt" singletons to create larger clusters. This is done by searching for a singleton v and a kernel K, whose pairwise fingerprint similarity is maximum among all pairs of singletons and kernels. The refined algorithm iteratively applies the adoption step and then the Basic-CLICK algorithm on the remaining singletons, stopping when there are no more changes.

- **Merge Step**: In this step we merge clusters whose fingerprints are similar (the justification for this is that, in practice, clusters can contain multiple kernels). The merging is done iteratively, each time merging two clusters whose fingerprint similarity is the highest (provided that the similarity exceeds a predefined threshold).

11.3.2 Assessing Clustering Quality

A measure for the quality of a solution given the true clustering is an important tool for evaluating clustering algorithms performance. Let T be the "true" solution and S the solution we wish to measure. Denote by n_{11} the number of pairs of elements that are in the same cluster in both S and T. Denote by n_{01} the number of pairs that are in the same cluster only in S, and by n_{10} the number of pairs that are in the same cluster only in T. We define the *Minkowski Score* to be:

$$D_M(T,S) = \sqrt{\frac{n_{01}+n_{10}}{n_{11}+n_{10}}}$$

In this case the optimum score is 0, with lower scores being "better".

An alternative is the *Jaccard Score*:

$$D_J(T,S) = \frac{n_{11}}{n_{11}+n_{10}+n_{01}}$$

Here the optimum score is 1, with greater scores being "better".

We shall use these scores to compare CLICK's performance with other clustering algorithms.

Program	#Clusters	Homogeneity		Separation	
		H_{Ave}	H_{Min}	S_{Ave}	S_{Max}
CLICK	30	0.8	-0.19	-0.07	0.65
GENECLUSTER	30	0.74	-0.88	-0.02	0.97

Table 11.1: A comparison between CLICK and GENECLUSTER [20] on the yeast cell-cycle dataset [5]. Expression levels of 6,218 S. cerevisiae genes, measured at 17 time points over two cell cycles.

Program	#Clusters	#Singletons	Minkowski	Jaccard	Time(min)
CLICK	31	46	0.57	0.7	0.8
HCS	16	206	0.71	0.55	43

Table 11.2: Source: [16]. A comparison between CLICK and HCS on the blood monocytes cDNA dataset [8]. 2,329 cDNAs purified from peripheral blood monocytes, fingerprinted with 139 oligos. Correct clustering known from back hybridization with long oligos.

11.3.3 Algorithm Performance Comparisons

This section contains examples of comparisons between CLICK and other clustering algorithms. Analysis of the comparison summary (table 11.6) shows that CLICK outperforms all the compared algorithms in terms of quality. In addition, CLICK is very fast, allowing clustering of thousands of elements in minutes, and over 100,000 elements in a couple of hours on a regular workstation. Figure 11.7 shows the result of a comparison in which the authors of each algorithm were allowed to run the test on their own. The graph shows a tradeoff between the homogeneity and separation scores; The further the algorithm is from the origin the "better" its overall performance.

11.3.4 Application of Clustering - Tissue classification

An important application of gene expression analysis is the identification of clinical markers in the expression levels, enabling the identification of new clinical sub-categories with prognostic

Program	#Clusters	#Singletons	Minkowski	Jaccard	Time(min)
CLICK	2,952	1,295	0.59	0.69	32.5
K-Means	3,486	2,473	0.79	0.4	–

Table 11.3: Source: [16]. A comparison between CLICK and K-Means [9] on the sea urchin cDNA dataset. 20,275 cDNAs purified from sea urchin eggs, and fingerprinted with 217 oligos. Correct clustering of 1,811 cDNAs known from back hybridizations.

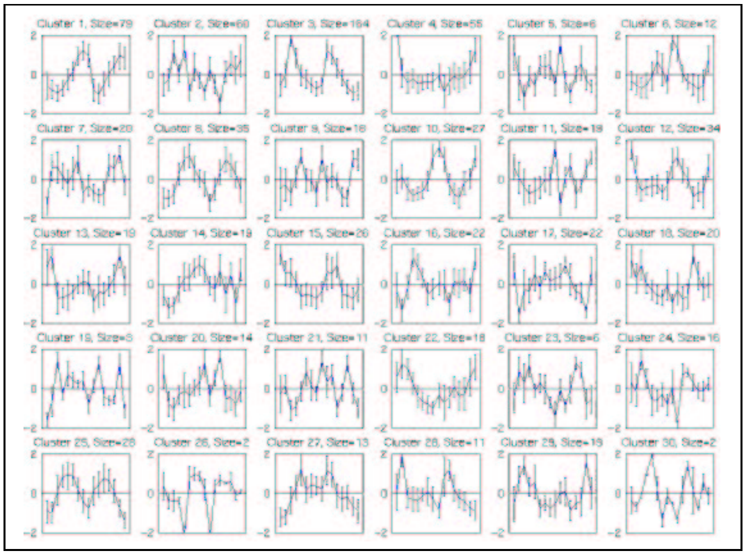


Figure 11.4: Source: [16]. CLICK's clustering of the yeast cell-cycle data [5]. x-axis: time points 0-80, 100-160 at 10-minute intervals. y-axis: normalized expression levels. The solid line in each sub-figure plots the average pattern for that cluster. Error bars display the measured standard deviation. The cluster size is printed above each plot.

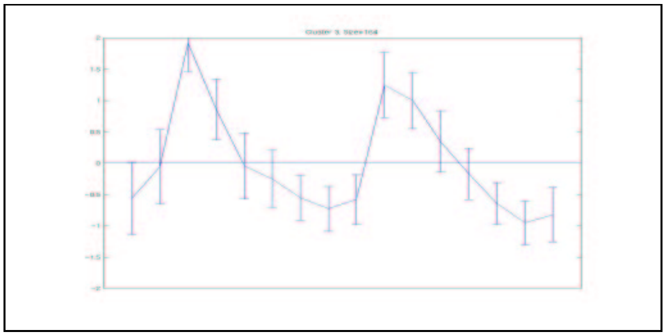


Figure 11.5: Source: [16]. Yeast Cell Cycle: late G1 Cluster (cluster 3 from figure 11.4). The cluster found by CLICK contains 91% of the late G1-peaking genes. In contrast, in GeneCluster 87% are contained in 3 clusters.

Program	#Clusters	Homogeneity		Separation	
		H_{Ave}	H_{Min}	S_{Ave}	S_{Max}
CLICK	10	0.88	0.13	-0.34	0.65
Hierarchical	10	0.87	-0.75	-0.13	0.9

Table 11.4: Source: [16]. A comparison between CLICK and Hierarchical [6] clustering on the dataset of response of human fibroblasts to serum [11]. Human fibroblast cells starved for 48 hours, then stimulated by serum. Expression levels of 8,613 genes measured at 13 time points.

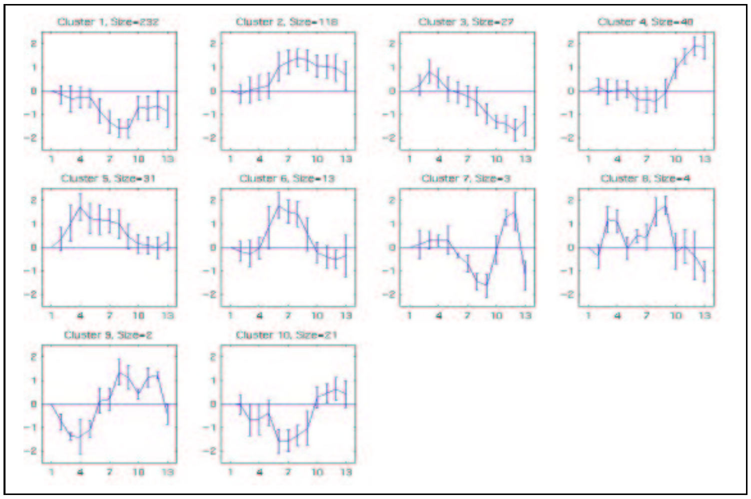


Figure 11.6: Source: [16]. CLICK's clustering of the fibroblasts serum response data [11]. x-axis: 1-12: synchronized time-points. 13: unsynchronized point. y-axis: normalized expression levels. The solid line in each sub-figure plots the average pattern for that cluster. Error bars display the measured standard deviation. The cluster size is printed above each plot.

Program	#Clusters	#Singletons	Homogeneity	Separation	Time(min)
CLICK	9,429	17,119	0.24	0.03	126.3
SYSTERS	10,891	28,300	0.14	0.03	–

Table 11.5: Source: [16]. A comparison between CLICK and SYSTERS on a dataset of 117,835 proteins [13]. Measures based on similarity when no correct solution is known: For a fixed threshold t, homogeneity is the fraction of mates with similarity above t, and separation is the fraction of non-mates with similarity above t.

Elements	Problem	Compared to	Improvement	Time(min)
517	Gene Expression Fibroblasts	Cluster [6]	Yes	0.5
826	Gene Expression Yeast cell cycle	GeneCluster [20]	Yes	0.2
2,329	cDNA OFP Blood Monocytes	HCS [8]	Yes	0.8
20,275	cDNA OFP Sea urchin eggs	K-Means [9]	Yes	32.5
72,623	Protein similarity	ProtoMap [22]	Minor	53
117,835	Protein similarity	SYSTERS [13]	Yes	126.3

Table 11.6: Source: [16]. A Summary of the time performance of CLICK on the above mentioned datasets. CLICK was executed on an SGI ORIGIN200 machine utilizing one IP27 processor. The time does not include preprocessing time. The "Improvement" column describes whether the solution of the CLICK algorithm was better than the compared algorithm.

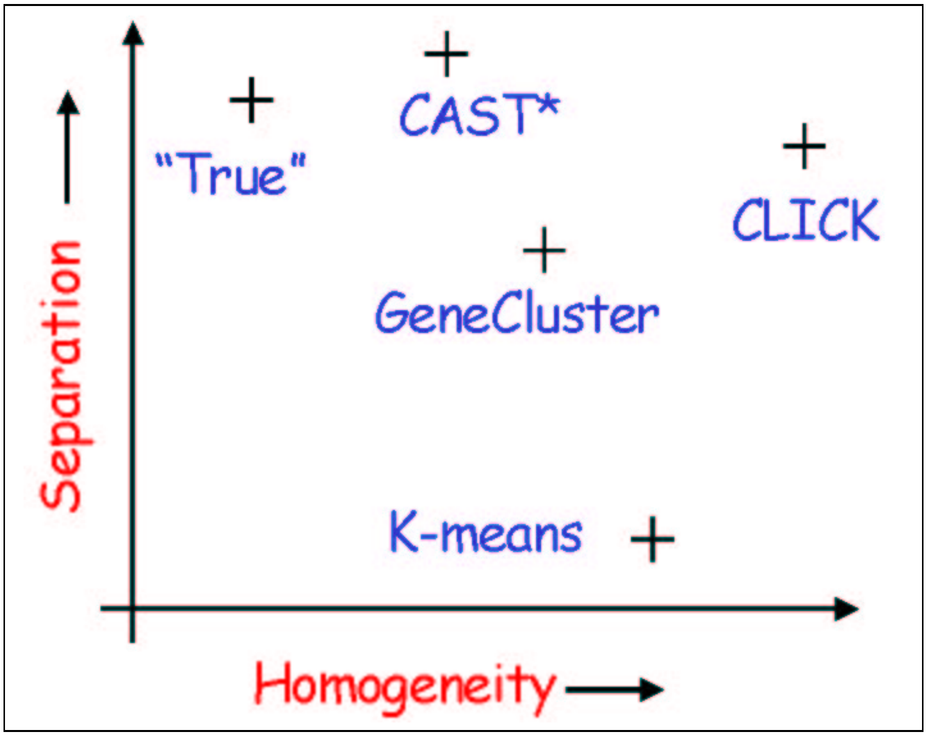


Figure 11.7: Source: [17]. Comparison of clustering algorithms using homogeneity and separation criteria. The data consisted of 698 genes, 71 conditions [19]. Each algorithm was run by its authors in a "blind" test.

implications. Given a human tissue, we would like to devise a computational method allowing us to determine weather this person is having some kind of cancer (for example) base solely on the expression levels in a sample from some relevant tissue. In this approach we cluster **tissues** rather than genes, and analyze the subsets of tissues to find clinically important groups.

We exemplify on the usage of such methods based on [17]. In this study, two data sets were analyzed: Alon et al. [4] provide data from 40 tumors and 22 normal colon tissues, Golub et al [7] provide data from two luekemia types (25 AML, 47 ALL).

Applying CLICK to the data and analyzing the correspondence between clusters and the know classification resulted in very good match. To asses the performance of clustering, a leave one out cross validation (LOOCV) techniques was applied. In this method, we hide one of the tissue classification and try to predict its class by the known class of tissues in the cluster it belongs to. Results are summarized in table 11.7.

To improve classification accuracy, feature selection may be applied. In this approach the genes are sorted by the ratio of their between-sum-of-squares and within-sum-of-squares values. This leads to a set of genes which are **informative** for the classification. Table 11.8 shows that improved performance is achieved by applying clustering after feature selection.

DataSet	Method	Correct	Incorrect	Unclassified
Colon	Click	85.5	9.7	4.8
	CAST	88.7	11.3	0.0
Leukemia	Click	90.3	4.2	5.5
	CAST	87.5	12.5	0.0

Table 11.7: Results of LOOCV analysis for human cancer classification using Click and CAST.

DataSet	Size	Correct	Incorrect	Unclassified
Colon	2000	85.5	9.7	4.8
	50	90.3	9.7	0.0
Leukemia	2000	90.3	4.2	5.5
	50	98.6	1.4	0.0

Table 11.8: The effectiveness of feature selection for tissue classification.

11.4 Sequencing by Hybridization

Standard oligo chips can, at least theoretically, be used for sequencing. Let us prepare an oligo chip that contains all possible sequences of length k. These sequences are called k*-mers.* Practical values of k are 8-10. If we expose this chip to a solution containing some target DNA, the results will show which k-mers occur in the target sequence.

Definition The k*-spectrum* of sequence T is the multi-set of all its substrings of length k.

Note, that the k*-spectrum* is a *multi-set.* We assume that if a k-mer appears more than once, in the target DNA, the hybridization experiment will report the number of its occurrences. To date, this requirement is impractical.

Problem 11.1 Reconstructing a sequence from hybridization data
INPUT: A multi-set S of k-mers
QUESTION: Does exist a sequence T such that S is the k-spectrum T? If yes, find T.

For instance, for $k = 3$:

$$\begin{aligned} T &= ATGCAGGTCCAG \\ S &= \{ATG, AGG, CAG, GCA, GGT, GTC, TCC, TGC, CCA, CAG\} \end{aligned}$$

The naive approach

Let us define a directed graph $G` = (V`, E`)$, where $V` = \{\text{existing } k\text{-mers}\}$ and an edge $E` = (v_1, v_2)$ exists iff the last $k-1$ characters of v_1 match the first $k-1$ characters of v_2 (i.e., it is possible that v_2's index in T, is the successor of v_1's index.

The problem is now to find a Hamiltonian path in the directed graph G. However, it is a common knowledge that the Hamiltonian path problem is NP-Complete. Therefore, this solution cannot be used for large input sets.

The polynomial solution

Luckily, a polynomial solution for this problem exists, due to Pevzner [15]. Define another directed graph $G = (V, E)$. This time the vertices will be $(k-1)$-mers. An edge $e = (v_1, v_2)$ is introduced if the first $k-2$ characters of v_1 match the last $k-2$ characters of v_2, and the concatenation of the first character of v_1, with the $k-2$ common characters and the last character of v_2 form a k-mer that was reported present in the sequence. This graph is called the *de-Bruijn* graph of the sequence. See Figure 11.8 for an example.

It should be noted that for this construction, it is *very important* to know whether a given k-mer occurs more than once in the target sequence. For instance, if ACA occurs two

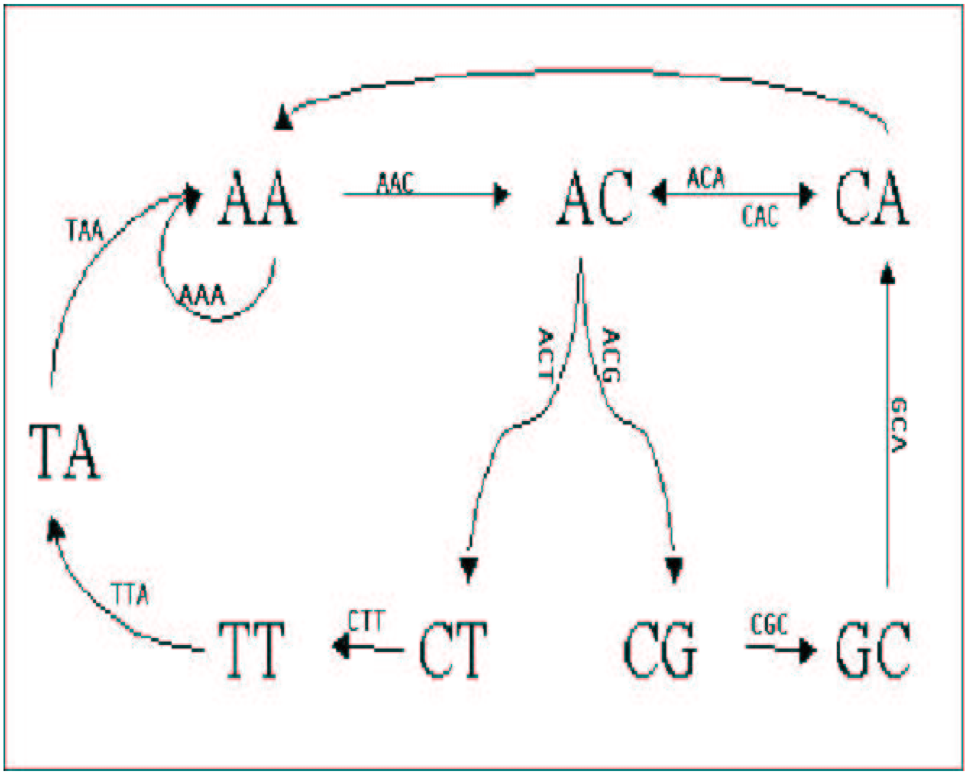


Figure 11.8: The $(k-1)$-mer graph of $S = \{AAA, AAC, ACA, CAC, CAA, ACG, CGC, GCA, ACT, CTT, T$
The mathematical problem here is to find an *Eulerian path*, that is a path that uses each edge once, and only once, as seen in the figure (T=ACAAACGCACTTAA).

times in S, then there should be two edges between AC and CA. Otherwise, our solution will not be correct.

While this solution is mathematically elegant, there are several problems with using it in true biological context:

1. For some graph configurations, there is more than one Eulerian path. In such cases we will not be able to reconstruct the sequence. For an example of such a graph, see figure 11.8. Figure 11.9 shows the probability that a random string S cannot be uniquely reconstructed from its k-spectrum. Some other chip designs achieve a somewhat better result, but these designs are only theoretical. They are usually very difficult or impossible to manufacture, and cannot be used in true biological context.

2. As in all biological experiments, the spectrum we measure contains a large proportion of errors. This solution is not robust enough to handle them.

3. A related problem is that of edge multiplicity. We can consider ourselves lucky to know with certainty whether a certain k-mer occurs in our sequence. In most cases we have no way of knowing exactly how many times it occurs.

Sadly, then, sequencing based on hybridization is not a real alternative to standard sequencing.

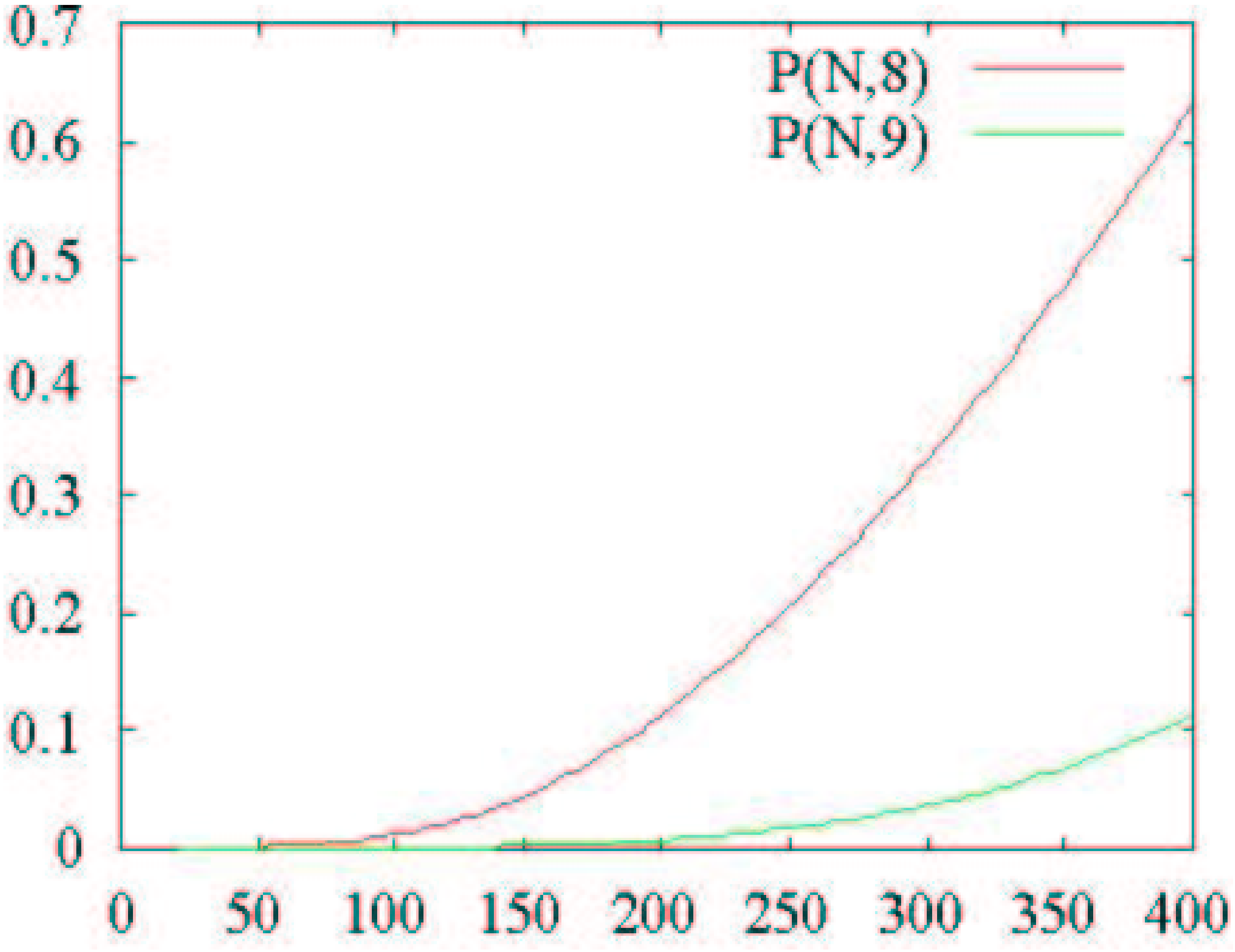


Figure 11.9: $P(N,k)$ is the probability that for a random string S of length N there exists a sequence S_0, whose k-spectrum equals that of S.

11.5 Biological Networks

The computational study of biological systems is characterized by being focused on the relations among biological factors rather than on the factors themselves. The biological function is derived from an extremely complex, redundant and robust **network** of interacting molecules, which together process information, perform sophisticated decision making and survive in a dynamic environment. There are many types of important biological interactions, all of the are combined to create the functioning cell.

The first type of interaction is denoted as transcription control and involve relation between proteins and genes. Each cell contain a full copy of the organism's DNA, but not all genes are always required or needs to be produced at the same rate. To control the rate of transcription, specialized protein called transcription factors can bind the DNA at different proximity to the gene start (50 to 20000 bp) and catalyzes the reaction that initiate transcription. Since these proteins are themselves genes products, and are thus subject to transcription control, we can model this system as a mathematical network of genes affecting the transcription of each other in a dynamic way (mathematically speaking, we have some kind of a dynamical system).

A second archi-type of interaction is the interaction between proteins. Protein can modify other protein post-translationally and completely change their ability to catalyze important reactions. For example, protein kinases can phosphorilate target protein thus Changing their conformation and activating them. Other proteins cooperatively bind partners to disable their activity or create a hetrodimer with specialized features. The reaction among proteins can also be regarded as a network. For example, a typical **Signaling Cascade** involve a membrane protein which change its conformation in response to the binding of an external signaling molecule. The conformation change can activate the membrane protein as a kinase for a second protein which in turn may also be activate to phosphorilate a third protein and so on (this is called a MAPK cascade). In the final step of this example protein network, we would usually have a transcription factor whose activity is regulated post- translationally (say by the final kinase of the cascade). This way information from the signaling cascade can find its way into the genetic network and change transcription.

A third class of interaction is the interactions of metabolites and proteins, since proteins act as enzymes for all of the important biochemical reactions in the cell, we can view the different metabolites and the metabolic pathways constructed by them as part of the larger biological network. The rate of metabolic activity is regulated by the level of enzymes available for each metabolic reaction, enzymes activity is itself subject to regulation by levels of certain metabolites.

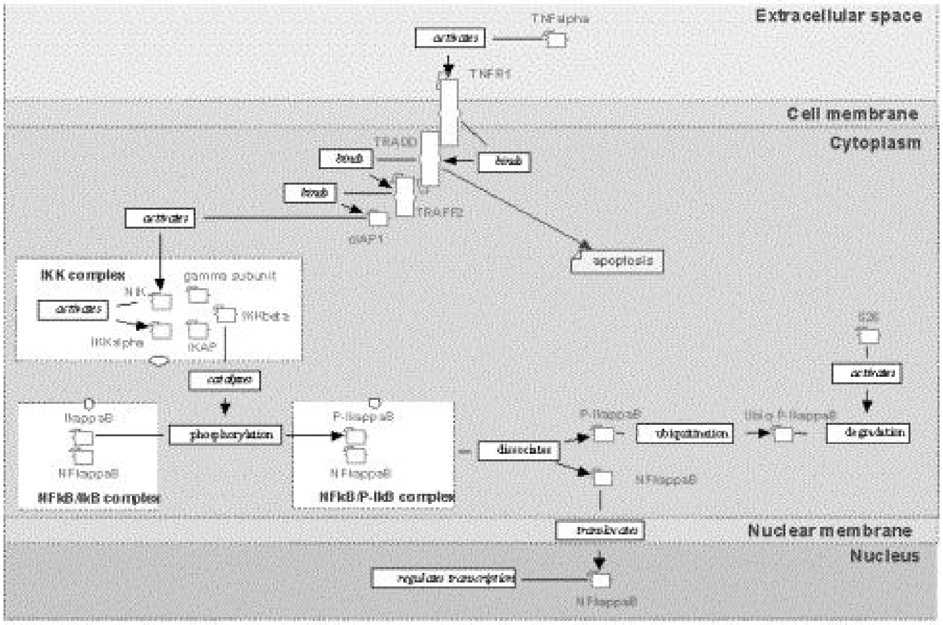


Figure 11.10: Source: MIPS website (mips.gsf.de). A signal transduction cascade is transmitting signals from outside the cell via post translational modifications of a series of proteins (primary reaction) and later activating a transcription factor post translationaly to initiate secondary reactions via the gene network of transcription control. The biological system is a composition of signaling pathways, transcription control networks and metabolic pathways which are functioning together in an amazingly complex set of relations.

11.5.1 The Vision

The ultimate goal of computational biological network analysis is to be able to use high throughput experimental information to construct a predictive model of the activity of the entire set of factors in a cell, much like the physics ideal of a state equation but for an amazingly large collection of variables. Such a goal is very non realistic with today's capabilities, but even modest intermediate steps in our way to it would be priceless in any attempt to understand biology, disease and drugs effect on the cell. Our early steps in this domain are using gene expression data to construct models of transcription control.

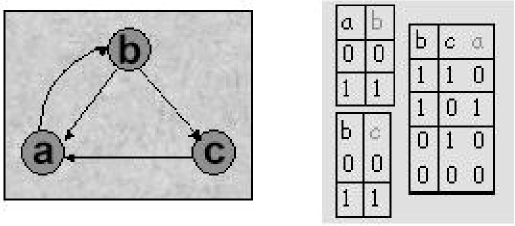
11.5.2 Experimental Complexity

We define **Experimental Complexity** as the number of experiments we must perform to reconstruct a network (in the worst case). Some prior knowledge on possible network topologies and logic change the experimental complexity of the resulted network class (more constraint will reduce experimental complexity). In what follow we shall use a simple mathematical model for gene networks to gain insights on the potential problems arising when attempting reconstruction of a network from expression data. Our model will be Boolean and we will assume each gene is a node assigned with a Boolean function defining its state at time t as a function of other genes at time $t-1$. An example of a Boolean network is given in Figure 11.11. We will model an expression experiment as a vector of Boolean values over the genes and will analyze the relations between the size and topology of networks to the number of examples needed to reconstruct a network in the worst case. To construct our set of examples we shall use **perturbations** of the network, each time fixing the values of a selected set of perturbed genes and measuring the values of all others. The biological motivation to this is our ability to perform directed mutations in model organisms and study the resulted phenotype (this is called a **knockout** experiment, and have a counterpart called **overexpression** in which a plasmid with a target gene is inserted to the cell and activated at a specific timing to produce large amounts of the proteins). The result of a collection of perturbation experiments can be summarized in a table as shown in Figure 11.12.

The theory in this section is taken from [2], the general theory of Boolean genetic network is rooted in the works of Stuart Kauffman (see [12] for much more on the motivation and consequences of such networks).

General topologies

We began by assuming no limitations on the topology of the genetic network. This is a very artificial case since we know from biology that the mechanism of regulation limits the number of genes or proteins that can interact with a single gene. We first show that an exponential number of experiments are required in the worst case.

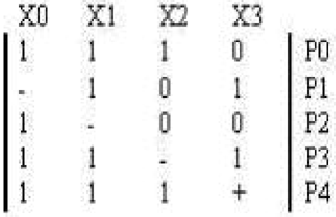


a	b
0	0
1	1

b	c
0	0
1	1

b	c	a
1	1	0
1	0	1
0	1	0
0	0	0

Figure 11.11: Source: [18]. Sample Boolean network



X0	X1	X2	X3	
1	1	1	0	P0
-	1	0	1	P1
1	-	0	0	P2
1	1	-	1	P3
1	1	1	+	P4

Figure 11.12: Source: [18]. An example expression matrix

Proposition 11.2 At least 2^{n-1} experiments must be performed in order to identify a general gene regulatory network in the worst case.

Proof: A counting argument. Consider a Boolean function of $(n-1)$ variables $f(x_1, x_2, .., x_{n-1})$ which is assigned to the node x_n. There are $2^{2^{n-1}}$ possible Boolean functions of $(n-1)$ variables. Hence we can identify this function by examining 2^{n-1} assignments and less examinations without suffice (we get one output bit per experiment). ■

Proposition 11.3 $n2^{n-1}$ experiments always suffice in order to identify a gene regulatory network.

Proof: We can always reconstruct a network by perturbation all $n-1$ sets of nodes to all possible assignments. We read the Boolean functions entries one at a time. ■

Theorem 11.4 *An exponential number of experiments are necessary and sufficient for the identification of a general Boolean network from perturbations.*

Bounded In-degree Case With Bounded Cost

As said above, general topologies are not of any practical interest. The topological structure of the genetic network is constraint in many ways, the most direct of those is the limit on incoming degree. Indegree limitations are not only a known biological fact but also provide the organism with evolutionary robustness as noted by [12]. Kauffman's simulation of random networks showed that general topologies network behave chaotically, but indegree limitation forces order on the network space and enable evolution to proceed and construct beneficial building blocks without destroying the organism after each minor change. In what follow we shall assume the in degree is bounded by a constant D. First, we consider the case $D = 2$.

Proposition 11.5 $\Omega(n^2)$ experiments are necessary for network reconstruction even if the maximum in degree is 2 and all nodes are AND nodes, if we assume that each experiment can perturb a bounded number of experiments.

Proof: Denote by C the maximal number of perturbation per experiment (the experiment cost). First, consider the case of $C = 2$. Assume that $\neg x \wedge \neg y \rightarrow z$ is assigned to z and the other nodes have in degree 0. Among all experiments only $(\neg x, \neg y)$ can activate z. Therefore, we must test $\Omega(n^2)$ pairs of nodes in order to find (x, y).
Next, we consider a case of $C = 3$ with the same function $\neg x \wedge \neg y \rightarrow z$. If we disrupt or overexpress u, v, w such that $x \notin \{u, v, w\}$ or $y \notin \{u, v, w\}$, we can only learn that $(u, v), (u, w), (v, w)$ are different from (x, y). Since there are $\Theta(n^3)$ triplets and only $\Theta(n)$ triplets can include $\{x, y\}$, at least $\Theta(n^2)$ triplets must be examined in the worst case (each experiment removes at most a constant number of pairs out of the $\Theta(n^2)$ possible ones).
For $C > 3$, similar arguments work.

■ If C is not bounded, the above proposition does not hold. It is possible to identify the above pair (x, y) by $O(\log(n))$ experiments of maximum cost n, using a strategy based on binary search. Although this strategy might be generalized for other cases, it is not practical since large number of simultaneous perturbation is not realistic. (The cells simply die if they are heavily mutated.)

Next, we consider the upper bound.

Proposition 11.6 $O(n^4)$ experiments with maximum cost 4 are sufficient for reconstruction if the maximum in degree is 2.

Proof: We assume (w.l.o.g) that all nodes are of in degree 2 since identification of nodes of in degree of 1 or 0 is easier. Let c be any node of V. We examine all assignments to all quadruplets $\{a, b, x, y\}$ with $c \notin \{a, b, x, y\}$. The Boolean function $g(a, b)$ is assigned to c (i.e., $f_c \equiv g$) if and only if there exists a Boolean function $g(a, b)$ such that $c \equiv g(a, b)$ for any assignment to $\{a, b, x, y\}$, where $c \equiv g(a, b)$ means that the *state* of c equals to $g(a, b)$. The 'only if' part is trivial, let us prove the 'if' part. Suppose that $g(a, b)$ is not assigned to c i.e. $f_c = h(a, b)$ and $h(a, b) \neq g(a, b)$. Clearly, $c \equiv g(a, b)$ does not hold. Next, consider a case where $h(p, q)$ is assigned to c where h may be equal to g and $\{p, q\} \cap \{a, b\} = \emptyset$. In this case, c takes both 1 and 0 by changing assignments to $\{p, q\}$ even if assignment to $\{a, b\}$ is fixed. Therefore, $c \equiv g(a, b)$ does not hold. In case of $\{p, q\} \cap \{a, b\} \neq \emptyset$, suppose $f_c \equiv h(p, b)$ and $a \neq p$. Then there is a value of b so that $h(0, b) \neq h(1, b)$, but then $f_c(a, b, p = 0, y) \neq f_c(a, b, p = 1, y)$ and $c \equiv g(a, b)$ does not hold again. Since all assignments to all quadruplets are examined, in total $0(n^4)$ experiments are sufficient. ■

This is easily generalized to general D, and can actually be improved to $O(n^{D+1})$ (left as exercises).

Theorem 11.7 *$O(n^{2D})$ experiments with maximal cost $2D$ are sufficient for the identification of a gene regulatory network of bounded in degree D. On the other hand, $\Omega(n^D)$ experiments are necessarily in the worst case if cost of each experiment is bounded by a constant.*

Other types of topology and logic restriction may further improve experimental complexity of the resulting reconstruction problem, this is summarized in Table 11.9.

Constraints	**Lower bounds**	**Upper bounds**
None	$\Omega(2^{n-1})$	$O(2^{n-1})$
In-degree $\leq D$	$\Omega(n^D)$	$O(n^{2D})$
In-degree $\leq D$ All genes are AND-nodes (OR-nodes)	$\Omega(n^D)$	$O(n^{D+1})$
In-degree $\leq D$ Acyclic	$\Omega(n^D)$	$O(n^D)$
In-degree ≤ 2 All genes are AND-nodes (OR-nodes). No inactivation edges.	$\Omega(n^2)$	$O(n^2)$

Table 11.9: Bounds on number of experiments needed for reconstruction (n - number of genes, D - maximum in degree). As seen from the table, forcing more constraints on the possible network topologies can improve experimental complexity significantly. The cases of acyclic topologies and restricted monotone logic (AND/OR gates only) are simpler mathematically but unfortunately have no good biological support.

11.5.3 Practical Approaches

Top down

Using a Bayesian network approach Pe'er et al. [14] constructed a framework for identifying significant subnetworks from expression data. The framework first learn a large number of Bayesian network topologies over the genes using each time a slightly randomized expression matrix (this is called bootstrapping). The collection of networks is analyzed to find edges of high significance (those which appear in many of the learned networks). Finally, dense subgraphs are being searched in the graph of significance edges and the resulted subgraphs are used as the output. The approach was proved to generate biologically meaningful results and is an example of hybrid techniques that analyze the entire network and search for significant patterns (a data mining approach).

Bottom Up

A complementary approach to the Top-Down method described above is given in [21]. In this framework, we start by modeling a known submodel and search for a best fit **expansion** of it using expression or other types of information. Using a known submodel as a starting point, we are facing reduced experimental complexity and enable the biologist to ask directed questions about a subsystem of interest (the biologist uses any knowledge to guide the reconstruction process and can use the system to test and generate new hypotheses). The method's first implementation proved to generate biologically meaningful results and is guaranteed to generate better and better results as the number of conditions in public databases increase.

Bibliography

[1] The chipping forecast. Special supplement to Nature Genetics Vol 21, 1999.

[2] T. Akutsu, S. Kuhara, O. Maruyama, and S. Miyano. Identification of gene regulatory networks by strategic gene disruptions and gene overexpressions. In *Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 695–702, 1998.

[3] A. A. Alizadeh, M. B. Eisen, R. E. Davis, C. Ma, I. S. Lossos, A. Rosenwald, J. C. Boldrick, H. Sabet, T. Tran, X. Yu, J. I. Powell, L. Yang, G. E. Marti, T. Moore, J. Hudson, L. Lu, D. B. Lewis, R. Tibshirani, G. Sherlock, W. C. Chan, T. C. Greiner, D. D. Weisenburger, J. O. Armitage, R. Warnke, and L. M. Staudt. Distinct types of diffuse large B-cell lymphoma identified by gene expression profiling. *Nature*, 403(6769):503–511, 2000.

[4] U. Alon, N. Barkai, D. A. Notterman, G. Gish, S. Ybarra, D. Mack, and A. J. Levine. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *PNAS*, 96:6745–6750, June 1999.

[5] RJ. Cho et al. A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol Cell*, 2:65–73, 1998.

[6] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein. Cluster analysis and display of genome-wide expression patterns. *PNAS*, 95:14863–14868, 1998.

[7] T. R. Golub, D. K. Slonim, et al. Molecular classification of cancer: Class discovery and class prediction by gene expression monitoring. *Science*, 286:531–537, October 1999.

[8] E. Hartuv, A. Schmitt, J. Lange, S. Meier-Ewert, H. Lehrach, and R. Shamir. An algorithm for clustering cDNAs for gene expression analysis using short oligonucleotide fingerprints. In *Proceedings Third International Symposium on Computational Molecular Biology (RECOMB 99)*, pages 188–197, April 1999.

[9] R. Herwig, A.J. Poustka, C. Muller, C. Bull, H. Lehrach, and J. O'Brien. Large-scale clustering of cDNA-fingerprinting data. *Genome Research*, 9:1093–1105, 1999.

[10] TR. Hughes et al. Functional discovery via a compendium of expression profiles. *Cell*, 102:109–26, 2000.

[11] V.R. Iyer, M.B. Eisen, D.T. Ross, G. Schuler, T. Moore, J.C.F. Lee, J.M. Trent, L.M. Staudt, J. Hudson Jr., M.S. Boguski, D. Lashkari, D. Shalon, D. Botstein, and P.O. Brown. The transcriptional program in the response of human fibroblasts to serum. *Science*, 283 (1), 1999.

[12] S.A. Kauffman. *The Origins Of Order, Self Organization and Selection in Evoultion.* Oxford University Press, 1993.

[13] A. Krause, P. Nicodeme, E. Bornberg-Bauer, M. Rehmsmeier, and M. Vingron. WWW access to the SYSTERS protein sequence cluster set. *Bioinformatics*, 15(3):262–3, 1999.

[14] D. Pe'er, A. Regev, G. Elidan, and N. Friedman. Inferring subnetworks from perturbed expression profiles. *Bioinformatics*, 17 Suppl 1:S215–24, 2001.

[15] P. A. Pevzner. l-tuple DNA sequencing: computer analysis. *J. Biomol. Struct. Dyn.*, 7:63–73, 1989.

[16] R. Sharan and R. Shamir. A clustering algorithm with applications to gene expression analysis. In *Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB 2000)*, 2000. To appear.

[17] Roded Sharan, Rani Elkon, and Ron Shamir. Cluster analysis and its applications to gene expression data. In *Ernst Schering workshop on Bioinformatics and Genome Analysis*. Springer Verlag, 2002. to appear.

[18] R. Somogyi and CA. Sniegoski. Modeling the complexity of genetic networks. understanding multigene and pleiotropic regulation. *Complexity*, 1:45–50, 1996.

[19] P. T. Spellman, G. Sherlock, et al. Comprehensive identification of cell cycle-regulated genes of the yeast Saccharomyces cerevisiae by microarray hybridization. *Mol. Biol. Cell*, 9:3273–3297, 1998.

[20] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, ES. Lander, and TR. Golub. Interpreting patterns of gene expression with self organizing maps: Methods and application to hematopoietic differentiation. *Proceedings of the National Academy of Science USA*, 96:2907–2912, 1999.

[21] A. Tanay and R. Shamir. Computational expansion of genetic networks. *Bioinformatics*, 17(Suppl 1):S270–8, 2001.

[22] G. Yona, N. Linial, and M. Linial. Protomap: Automatic classification of protein sequences, a hierarchy of protein families, and local maps of the protein space. *Proteins: Structure, Function, and Genetics*, 37:360–378, 1999.