

Tutorial:

**Genetic network analysis —
From the bench to computers and back**

November 4, 2001

2nd International Conference on Systems Biology

© Zoltan Szallasi

How to read this tutorial

This tutorial was written both for scientists with a background in biology and for people with a more mathematical/computational background. Certain concepts or results will be rather obvious to one group but not to the other. Please, be patient or simply skip the familiar parts and move on to something more relevant for you. You should also understand, that the main purpose of this tutorial is providing a good heuristic "feeling" about what certain algorithms do, or what a certain biological research field is trying to achieve. Therefore, the detailed mathematical descriptions or precise biological definitions often had to be sacrificed in order to stay focused. These details can be found in the appropriate and often referenced literature. I have also included references to several truly excellent tutorial web-sites on particular techniques. These might give you the necessary details you are missing from these pages.

I would also like to advise you to visit my ever (although somewhat slowly) developing web-site (www.usuhs.mil/pha/faculty/zoltan.shtml) or (chip.org) for two reasons. First, the on-line tutorial and this printed tutorial are not exactly overlapping. Second, you can directly connect from there to several other related web-sites.

I first thought of giving this tutorial at the beginning of 1999, when there were only a handful of papers published on this topic. At that time I planned to cover every major issue in detail – for example, it was easy to explain all clustering algorithms that have been applied for gene expression analysis. As the field started to explode, it has become more and more difficult to cover every single approach published in the literature, especially considering that papers often tend to cover overlapping problems. Therefore, I decided to settle for the following goals:

- covering the major concepts of genetic network analysis
- trying to keep a fairly updated reference list focusing on the most exciting results.
- and attempting to keep track of all significant results where computational sciences helped biology

If you think that I have missed mentioning a truly important paper (especially if it is yours), do not hesitate to contact me at the following address: zszallasi@chip.org

The tutorial will be (loosely) structured around the following topics:

- 1. Introduction: combinatorial biology, information content in biology, reductionism**
- 2. The information content of massively parallel data sets in biology**
- 3. The conceptual framework of genetic network analysis**
- 4. Genetic network modeling, forward modeling or “in silico biology”**
- 5. Reverse engineering of genetic regulatory networks**
- 6. Classification and cluster analysis in gene expression matrices**
- 7. Generative models in the analysis of gene expression matrices.**
- 8. Systems approach to genetic networks and biology**

These issues being out of the way let us start with the

1.1. Definition of genetic network analysis

Most biological phenomena are caused by an ensemble of cooperating biochemical entities including mRNA, proteins, small molecules (such as hormones) or ions. Genetic network analysis exploits massively parallel measurements in order to determine the regulatory interactions between genes and their derivatives, i.e. proteins in their different states. Based on these regulatory interactions, it also intends to provide predictive power about the behavior of individual genes under given conditions and the overall behavior of the system (e.g. whether a cell will turn malignant or not.)

1.2. Putting genetic network analysis into the context of other disciplines

The study of genetic networks lies in the border area of molecular biology, computer simulations and the study of complex heterogeneous systems. The theoretical and experimental approach to genetic networks developed rather independently, the scientific interaction often being limited to polite curiosity from both sides. There has been an apparent lack of tangible problems requiring the united efforts of experimental biology and theory, since biological studies have produced a relatively low amount of information that was traditionally analyzed by simple decision trees. The limitations of technology lead to a somewhat tautologous strategy of understanding molecular biology. Genes or gene products were classified whether they had a dominant effect on a given endpoint in a certain phenotypic assay. If they did not, they were discarded as irrelevant. If they did, simple decision trees were satisfactory to predict what would happen if the gene or its product were modified, since the gene had a dominant effect in the given biological assay (by virtue of identification). This circular reasoning and the state of experimental tools of molecular biology did not necessitate dealing with combinatorial issues or complex network analysis. Analysis, decision making and prediction did not need to be delegated from “human thinking” to “machine thinking”. In cancer research, for example, all experimental tools are biased towards the identification of dominant oncogenes, and the question of non-dominant cooperating oncogenes has been largely ignored. In this latter case, several genes together induce neoplasia, but individually they have no detectable transforming ability. Identification of non-dominant cooperating oncogenes requires a large body of experimental information to start with, and powerful computer tools to analyze complex regulatory interactions. The lack of both of these requirements directed almost all attention towards dominant oncogenes, which in turn proved to be a fruitful research field, since these genes have readily detectable effects. However, as a result, the complex network nature of gene regulatory interactions has been ignored.

1.3. What has changed ?? (i.e. why are you reading this tutorial ?)

In a somewhat formalized way we can say that our knowledge about a biological system can be characterized by the ratio of information extracted relative to the potentially relevant information in the system. Prior to this decade this ratio was very low and this very fact defined the general strategy of biological research, which was focusing on obtaining data only about “important” parameters. This was done by truly ingenious experimental techniques whose main virtue was often their ability to reduce the amount of experimental information required to solve a given problem. (See e.g. in the next chapter the screening techniques for oncogenes.) I hope, nobody will be offended if we use the term "reductionism" in a different context from its original definition: It can also mean the "reduction" of information used to solve a particular question.

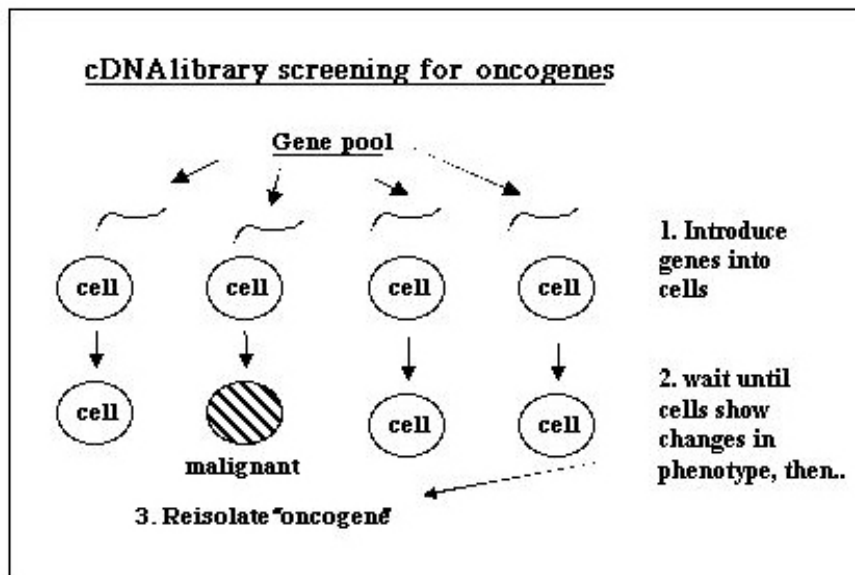
Recent technological developments, such as large-scale gene-expression measurements and the human genome project promise an increase of several orders of magnitude in the amount of experimental information. This sudden surge in actual data has renewed the interest in genetic networks from an experimental point of view. For example, it is manageable now for a single scientist to measure within a few months the expression level of all genes of a given organism, such as the approximately 6000 genes of yeast, in a time-dependent manner during the cell cycle. Consequently, biologists will produce less biased and significantly larger experimental databases that cannot possibly be analyzed by simple

decision trees. Therefore, experimentalists will face the question of how to make use of large-scale gene expression measurements. Can biological interactions be efficiently predicted based on massively parallel measurements? Can genetic network theory, in cooperation with computer simulations, make any useful, testable prediction about phenotypic changes in a given biological system? These changes in experimental biology has also prompted molecular biologists to recognize that, for quantitative analysis of gene-expression patterns, living organisms can be viewed as massively parallel computers. All these developments will lead to a markedly different "non-reductionist" approach to biology.

Genetic network analysis is expected to help experimental biology in many ways. First, massively parallel mRNA or protein quantitation can produce time-dependent measurements, termed expression matrices, on a significant portion of the members of a genetic network. These expression matrices are the result of the underlying regulatory network. Analytical methods, in particular reverse engineering, seek to extract information from time-series measurements in order to identify regulatory interactions in these genetic networks. Then the predicted individual regulatory interactions can be experimentally tested. Forward modeling, on the other hand, is expected to produce expression matrices that accurately predict the time-dependent gene-expression values. All forward modeling starts with an empirically determined database. For example, all molecular biologists studying human cells work on the same directed graph representing the human genes and gene products and their regulatory interactions. If the data generated by them is organized into an efficient database with sufficient understanding of the dynamic regulatory interactions, then, at least in theory, this database could predict time-dependent expression matrices given an initial parameter set. Then the predicted gene-expression patterns can be correlated with the experimentally determined gene-expression matrix and the observed phenotype. The most immediate use of computational analysis, however, is probably correlating phenotypes with genotypes, often using population and time-averaged measurements.

1.4. An example of why we need genetic network analysis.

What (if anything) is wrong with the "reductionist" approach? For the answer let us take an example from cancer research. Identification of oncogenes without previous knowledge about their function involved the development of ingenious screening procedures based on the introduction of a diverse gene pool into a biological system. As shown on the figure below, libraries of cDNAs or short random DNA fragments are transfected into an appropriate cell line, and then genes or gene fragments that induce a desired phenotypic change, such as neoplastic focus formation, are identified.



This is an excellent method as long as one is looking for dominant oncogenes, i.e. genetic changes that by themselves induce or trigger malignant transformation.

Unfortunately, human cancer is more complicated. First, cancer is the result of several cooperating genetic changes. Second, many of these changes cause malignant transformation only in a certain combination but individually do not induce any obviously detectable phenotypic changes. In other words, cancer is caused both by dominant and non-dominant cooperating oncogenes. Of course, we know much more about the former than the latter, since biological research has been strongly biased towards the identification of single dominant causes. What happens if we try to apply library screening methods for the identification of non-dominant cooperating oncogenes? Let us assume that three (or N) non-dominant cooperating oncogenes are causing cancer. Since we do not know which combination is important we would like to make sure that all possible combinations are represented in our screening process. This would call for a large number of transfected cells, on the order of $100,000^3 = 10^{15}$ transfectants (or 10^{5N} in the case of N cooperating genes), assuming that the human genome contains 100,000 genes. This would require tens of thousands of tissue culture dishes even by conservative estimates. The exponential increase in the number of samples to be tested will impose a practical, if not conceptual, limitation on the use of library-screening methods for identifying non-dominant cooperating oncogenes. This is only one of many possible examples that demonstrate why traditional experimental approaches are not really efficient to identify combinatorial causes.

These type of problems will constitute a new area of research that might be called combinatorial biology and which requires a less biased extraction of information from biological systems. This can be achieved with the help of "genomics" that has increased the ratio of extracted vs. potentially relevant information by several orders of magnitude by providing information about thousands of genes present in a given biological system.

1.5. Is the revolution really coming?

The current changes in molecular biology such as massively parallel gene expression measurements, the human genome project etc. are often touted as a "revolution" both in the lay and scientific media. The statement "scientific revolution", however, requires some support (see T. Kuhn's work etc.). e.g by a clear demonstration of quantity transforming into quality. In other words, finding a given gene within one postdoc-week instead of ten postdoc-years will not constitute a major paradigm shift by itself.

I will list here several potential breakthrough questions, the answers to which would clearly indicate a major step ahead in biology:

- Can we perform efficient and meaningful reverse engineering? - For example, can we efficiently identify regulatory interactions between genes from time-dependent gene expression measurements.
- Can we identify non-dominant cooperating factors responsible for a given phenotype? - especially if that cooperation constitutes a "tricky" rule, for example Exclusive OR type interaction.
- Can we predict truly new subclasses of tumors (- and not only "repredict" -) based on their gene expression patterns?
- Can we understand the robustness of biological systems in terms of the overall architecture of genetic regulatory networks?

In order to demonstrate the difficulty of solving the questions listed above we may for example consider, that the answer to the first question is not necessarily YES (as we will see later). The useful information content of massively parallel biological measurements may be too low to perform meaningful reverse engineering - and there may be a solid theoretical reason for that.

As a conclusion to this introduction let us outline the contrast between reductionist approaches and the non-reductionist genetic network analysis:

Classical	Genome-Scale
Biased data search, based on the most obvious and direct correlations between genes and phenotypes - single or oligoparametric	Uses a large body of unbiased information (e.g. the expression of all genes)
Human decision based (there is no need for powerful computation)	Data management and prediction is delegated to computers
Essentially non-combinatorial approach. Subtle and non-linear effects are not identified in screens and are ignored	No conceptual restrictions on the complexity of the problems to be solved (although there will be computational restrictions)

2. The useful information content of massively parallel measurements.

Beyond all technical details the most important issue for genetic network analysis is the precision and reproducibility of massively parallel measurements.

This will be limited by at least two factors:

1. To a smaller extent by "conceptual issues": What are we normalizing for?

For example: if we normalize per unit RNA then decrease in the level of a given message(s) unavoidably leads to a relative increase in the level of other messages - determining absolute levels, such as number of RNA copies per cell, on the other hand might not be practical.

2. To a larger extent by practical/experimental issues such as array preparation, labeling efficiency, or that the initiation of reverse transcription is stochastic (especially for rare messages) etc. This leads to the somewhat alarming observation that cDNA microarray measurements on the very same RNA sample split into two parts will always yield close to 1 percent of differentially expressed genes. (This means measuring the same sample twice!!!!!!) In other words, the above mentioned experimental factors can cause an about 1.5 to 2-fold "pseudo-change" in the level of about 1% of the quantified population.

The general consensus in the field estimates that a reliable detection of 2-fold differences seems to be the practical limitation of massively parallel quantitation, especially if we consider cross experimental comparisons. This will in turn set a maximum on the useful information content of massively parallel measurements with practical consequences on e.g. reverse engineering. Since a rational experimenter will sample gene-expression according to a time-series in which each consecutive time point is expected to produce at least as large expression difference as the error of the measurement, this will set a practical limit for the time window of massively parallel measurements. (approx. 5 min intervals for yeast and maybe 15-20 min intervals in mammalian cells.) The error of measurements will also determine in bits the information content of a given data point. From these one can calculate a theoretical maximum of useful information content.

For example, for analyzing the cell cycle the useful information content of the gene expression matrix will depend on:

1. Measurement error
2. Kinetics of gene expression changes (an issue that might be deeply influenced by stochasticity)

3. The number of genes changing their expression level and the frequency of changes produced by individual genes
4. The time frame of the experiment.

Taken together one will find that time dependent gene expression measurements of the cell cycle produce 1-2 orders of magnitude less information than expected in an ideal case.

Before you send me an angry e-mail that your group can do better (which remains to be seen), I would like to point out that the useful information content of these measurements will not change much even if a reliable quantitation of e.g. 1.6-fold appears in the field. The point is that there is a limit of extractable information, which can be estimated and this will determine whether a certain predictive calculation can be performed or not.

One final thought. We can be certain that one way or another we will rely on artificial intelligence type learning algorithms to solve reverse engineering, clustering etc. problems. In order to solve these problems by computational tools one often wants to start with the largest possible appropriate data set (e.g. for the pattern recognition of hand written digits one can create a huge number of sufficiently different samples). This strategy generally helps to improve the performance of computational tools (e.g. artificial neural nets). In biology, however, the number of samples might be quite limited, at least relative to the complexity of the problems, because, for instance, the cell has to survive. This lack of data is probably the most frustrating aspect for computer scientists, but one that is not likely to change in the near future.

3. The conceptual framework of genetic network analysis

(Strictly for readers with no biology background whatsoever !!!!)

This section will provide a brief overview about the nature and the number of interacting parameters in genetic networks.

One of the nicest things about undergraduate physics is that the rule of interaction is the same between any two members of a given system. For example, the law of gravitation is the same between the Earth and the Moon and the Earth and the Sun. Unfortunately, this is not true for complex systems such as biology. The regulatory interactions between any two members of a genetic network can be different, therefore must be individually determined and used in complex calculations.

The cell (the frequently used experimental unit) is a network of “gene derivatives” and other biochemical entities (ions, small molecules, like hormones etc.). Gene derivatives include mRNA, proteins in their different inactive or active states. The term genetic network might be a bit misleading to a newcomer to the field - Although it is true that genes are regulating the states of other genes, but they do so by means of several intermediaries. A whole series of regulatory events exist between the activation of a certain gene and the effect of the same gene on a downstream regulated gene, and these regulatory events often receive multiple conditional inputs from an array of other elements in the network. Let us review a demonstrative example for the several steps involved from the production of mRNA of a transcription factor until the production of mRNA of a downstream-regulated gene. The product of the gene “c-jun” forms the AP-1 transcription factor complex, either with another identical c-jun molecule as a homodimer, or with one of several other proteins as a heterodimer . We start from the state when the mRNA of c-jun is already produced. In addition to the transcriptional regulation, the level of mRNA of this gene can be regulated by the stabilization or destabilization of mRNA (Figure 3.1). The level of protein production will be proportional to the net amount of c-jun mRNA and not to the transcriptional activation of this gene by itself. mRNA is the first regulated derivative of the c-jun gene. All proteins are produced in the cytoplasm in a non-modified form, and the jun protein has to be first translocated to the

nucleus to exert its transcriptional activity. The non-phosphorylated cytoplasmic and non-phosphorylated nuclear jun protein can be considered as two further derivatives of the c-jun gene, since there is evidence for the independent regulation of c-jun localization. The activity of the jun protein is significantly enhanced by phosphorylation at serine 73 and serine 63. The nuclear phosphorylated form of c-jun can be considered as an additional derivative, since both the function and the regulation by stabilization differs for the phosphorylated and non-phosphorylated form. For example, the non-phosphorylated form can effectively inhibit the glucocorticoid receptor activity, but it is much less effective in binding to and activating the so-called TPA-responsive element. As shown on Figure 3, c-jun has at least four independently regulated derivatives: its mRNA, the non-phosphorylated cytoplasmic, non-phosphorylated nuclear and the phosphorylated nuclear forms. The independent regulation of different levels of the genetic is clearly demonstrated by a recent study that found a correlation of 0.48 between the abundance of mRNA and its protein derivatives in human liver.

Biochemistry is constantly increasing our knowledge about the localization of proteins in different subcellular domains and about posttranslational modifications, often at multiple and independently regulated sites of the same proteins. Without doubt, this will further increase the number of independently regulated derivatives of each gene. In addition to the “specific regulatory inputs” of proteins the activity of biological molecules is often regulated by the concentration or concentration gradient of small molecules or ions, such as Ca^{++} , as well.

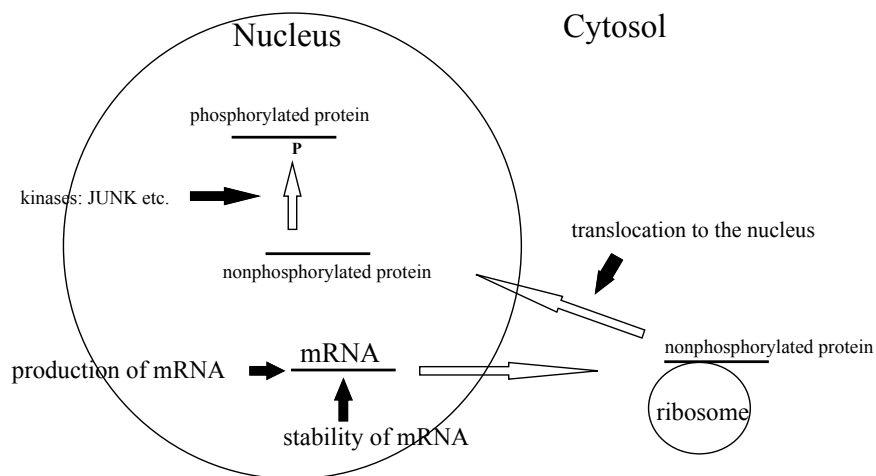


Figure 3. 1. Independently regulated derivatives of the c-jun gene. (For details see text.) The black arrows indicate independent regulatory inputs

From a genetic network aspect all these different biochemical entities could be termed as "biological information carriers" or "biological parameters".

A biological parameter can be defined as a biochemical entity, that:

- can be measured
- is chemically rather homogeneous
- determines by itself or in combination with something else the state of another biological parameter.

Using this definition we can make a first estimate on the complexity of genetic networks. Currently, the biochemical description of a single gene and its derivatives involves about 5 to 10 distinct biological parameters on average. This would result in a deterministic genetic network with about 10 times as many

members than the total number of active genes. There are about 10,000-20,000 active genes per cell. Therefore, based on current biochemical knowledge it seems that the number of biological parameters in a cellular network will be in the range of 10^5 - 10^6 biological parameters.

A final comment on terminology. Historically, theoreticians often discussed cells as a “network of genes”, in order to avoid constantly referring to mRNA, proteins etc. For the sake of simplicity, we will follow the same practice and from now on we will refer to all gene derivatives as “genes” - even if we know that the system is more complex and contains several levels of regulation. Similarly, for simplicity, we will talk about genetic networks, and gene regulatory interactions.

4. Genetic network modeling

Genetic network modeling was pioneered by a handful of scientists, the most widely known of them Stuart Kauffman, several decades ago. Computers were slowly coming of age, and it became clear that by computer modeling one can address issues about complex systems that could never be answered by analytical means. These scientists asked very exciting questions: What makes a genetic network stable? Are there certain genetic network architectures (i.e. wiring diagrams) that are more likely to be compatible with life than others? Is there any overall, fundamental reason for the average length of the cell cycle or for the number of types of differentiated cells in a complex organism? Although these issues were relegated to the mystical realm of theoretical biology, these early efforts in genetic network modeling led to several challenging concepts about how we understand the organization of complex genetic networks. The appearance of massively parallel data acquisition will certainly lead to renewed efforts in this interesting field.

The general view of genetic network modeling has long held, that the field produced many exciting results by a group of smart theoreticians BUT these results are of little use to experimental biology? This leads to the question of:

4.1. Why genetic network modeling?

Genetic networks can be viewed as massively parallel computers. We also know that we won't have easy closed formulae to predict what will happen in these systems.

The only hope for prediction is if we can:

- (1) extract enough information determining the system AND
- (2) we can reproduce their regulatory architecture AND
- (3) we can handle computationally the forward propagation of changes.

Genetic network modeling provides a preliminary test for points 2. and 3.

In fact, there are research groups who are currently attempting to deduce the regulatory architecture of a given part of the genetic network, forward model the behavior of this sub-network and then experimentally validate their predictions. (Hill et al., 2001)

Genetic network modeling will hopefully help to answer questions of great practical relevance as well, such as:

1. What kind of network architectures are compatible with the gene expression changes observed in vivo - such as the empirical ratio of cycling versus non-cycling genes, or the fact that genes do not get turned on and off more than twice during a cell cycle etc. ? (Szallasi & Liang, 1998)
2. Is there any “network architecture” reason for the maximal number of differentially expressed genes between different cell types ?

3. Are there theoretical constraints on the reversibility of attractor transitions in genetic networks ? (malignant conversion is considered by some theoreticians as an attractor transition)
4. Theoretically determined network constraints can aid forward modeling and reduce the computational requirement for reverse engineering. For example: if stability requires low connectivity of the genetic network then the number of possibilities to be tested computationally in reverse engineering or forward modeling will be significantly reduced.
5. Genetic network modeling may help to produce null-hypotheses for the statistical analysis of large-scale gene expression measurements.

4.2. Genetic Network modeling can be classified (if it is absolutely necessary) by two criteria:

- the size of the genetic network
 - Genetic networks can be studied on small scale, when only a few genes are studied. For example Becskei and Serrano (2000) showed that negative feedback may be an important factor to keep gene expression levels at a stable level without huge fluctuations. They showed this both experimentally and by modeling this small genetic network with differential equations.
 - Genetic networks can be modeled at an intermediate level, when the interaction of a couple of tens of genes are studied (Hill et al, 2001, Shapiro et al. 2000). These studies usually require rather "cool" software solutions such as automatic equations generators etc. This approach may provide an opportunity to investigate robustness and the effect of perturbations in these networks.
 - Large scale or "ensemble" modeling deals with realistic sized networks of thousands of interacting genes. They have been popularized by Kauffman (1993) in order to ask questions whether we can understand the general behavior of genetic networks. For example, is there any overall requirement on the network architecture that ensures stability etc.
- the nature of the regulatory interactions
 - (1) they can be approximated by Boolean rules OR
 - (2) they can be described by differential equations
 - (3) they can be modeled by stochastic equations. (see more on these issues below)

4.3. The principles of "ensemble modeling" of genetic networks:

They are deceptively simple. One starts with defining a set of genes and their interactions - e.g. by a directed graph in which each gene is a node and each directed vertex denotes a regulatory interaction (i.e. who regulates whom). We also need to define the function that describes the regulatory interaction between the genes that can be either Boolean, continuous, or stochastic. Then take an initial value set and observe how the system will behave when left alone. All that remains to be done is to make sense out of these observations.

(Of course, the observation itself can be a little troublesome when for example in a Boolean genetic network the number of possible gene expression states (for n gene) is 2^n . For yeast, this means about 10^{2000} states, which is a very-very large number.)

The essence of "classical genetic network" modeling is that we are NOT "reverse engineering" the network, i.e. keep changing the rules or architecture until they fit a certain set of experimental observations. Instead, we try to set certain "overall features" of the network e.g. the average number of regulatory inputs/gene and then observe the overall behavior of the network.

Boolean genetic network models offer a good starting point for two reasons. First, it is easier to understand the basic concepts through them. Second, for computational reasons Boolean networks have provided the only results on large ($n > 100$) genetic networks. Boolean networks already impose some formidable computational problems. Introducing continuous and stochastic models render even smaller networks computationally intractable.

Nevertheless, there have been heroic efforts to introduce more realistic genetic network models.

4.8. Continuous (differential equations) models

In these models regulatory interactions are more accurately approximated by differential equations. Unfortunately, this leads to serious computational problems that could be possibly circumvented by various methods: See e.g. (Novak & Tyson, 1997; Glass & Kauffman, 1973; McAdams & Shapiro, 1995)

4.9. Stochastic models

There is little doubt that these models are the closest to reality, therefore we will provide a more detailed introduction to this topic.

For analytical purposes, genetic networks can be viewed either as deterministic or stochastic systems. As we have seen in the case of Boolean models, a deterministic network is a rigid system, where the gene-expression state at a given time-point and the regulatory interactions between them unambiguously determine the gene-expression state at the next time-point. In such a network, there is only one path leading from a certain gene-expression state to another, since none of the gene-expression states can have two different successive outcomes (Figure 4.4.). In a stochastic system, on the other hand, a given gene-expression state can generate more than one successive gene-expression

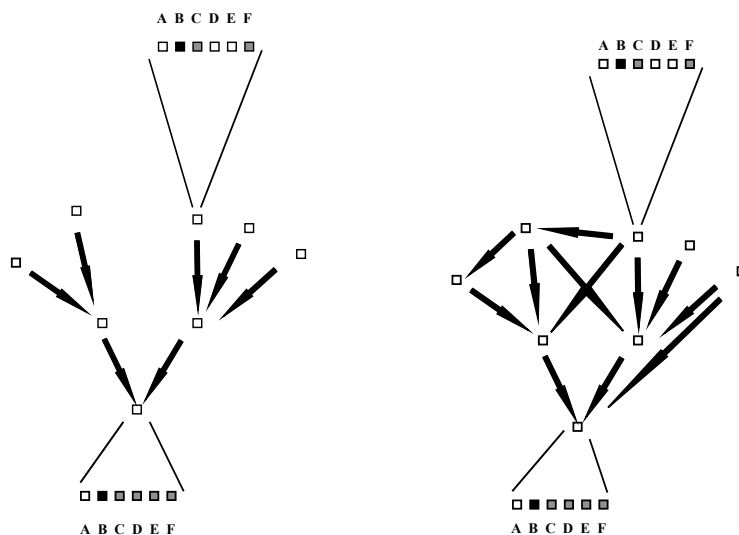


Figure 4.4. Graphic representation of deterministic and stochastic genetic networks. Every square-shaped node represents an entire gene-expression state, defined by the expression level of thousands of genes. Here we show only six (A to F) genes, displaying levels from no expression (empty squares) to maximum level of expression (full squares). Connective directed "edges" represent state transition paths showing the direction of the transition. Note, that in the deterministic network of figure A there is only one state transition path exiting each state, whereas more than one path can lead into the same gene-expression state. In the stochastic network of figure B, a multitude of transition paths exit each node, and there are several paths entering each gene-expression state as well.

states, and therefore, different cells of the same population may follow a different gene-expression path from one state of gene-expression to another. The fact that in reality genetic networks are stochastic, is supported both by theoretical considerations and experimental results. The number of transcription factors in a cell nucleus is often low, about a couple of hundred; the environment in which the gene regulatory interactions occur is far from free solution; and the reaction kinetics is relatively slow. As a result of all these factors stochastic mechanisms describe the kinetics of gene regulation more accurately than a deterministic description (such as a set of continuous differential equations). In addition, there is a growing number of experimental results both in prokaryotes (reviewed in , McAdams and Arkin, 1997)

and eukaryotes, such as hematopoiesis in mammals (Abkowitz et al. 1996) that could be best explained and modeled by stochastic mechanisms. In general, stochasticity allows significant variations in the sequence of activation and inactivation of genes. In extreme cases, this can result in two cells undergoing the same phenotypic change (e.g. proceeding from a certain point in the cell cycle to a later one) but having the sequence of activation for two genes reversed (Figure 4.5.). Conceptually this would not be a problem if we were able to analyze the gene-expression matrixes derived from individual cells, although the computational cost of analyzing a large number of alternative time-series data, considering the stochastic generation of gene-expression matrixes is probably high. More importantly, time-series measurements will always be obtained as population-averaged data. Current technology for massively parallel gene-expression measurements requires starting materials derived from a large number, often up to tens of thousands or millions, of cells. Even if it were possible to extract all quantitative parameters from a single cell at a given time point (e.g. using highly sensitive, PCR-based technology) the measurement itself kills or profoundly alters the organism, and we have no idea which transitions the organism would have proceeded through at later time points. In extreme cases, like the one in Figure 4.5., the population-averaged measurements will mask the real regulatory interactions. In reality, gene X and gene Y might be in a regulatory interaction with each other and never be activated at the same time. The time-averaged measurements, however, would suggest that they are activated at the same time. Stochastic simulations of large genetic networks are needed to assess how often we can expect the masking of regulatory information by the above-described mechanism.

Stochasticity can be one of the main reasons for the lack of sharp switch-on and -off kinetics of gene-expression, as often observed in experimental systems. This will put an empirical limit on the number of informative time-points obtained in time-series measurements as discussed in section 2. (For more detailed discussions of stochastic genetic networks see Arkin et al. 1998, McAdams and Arkin, 1997 & 1998)

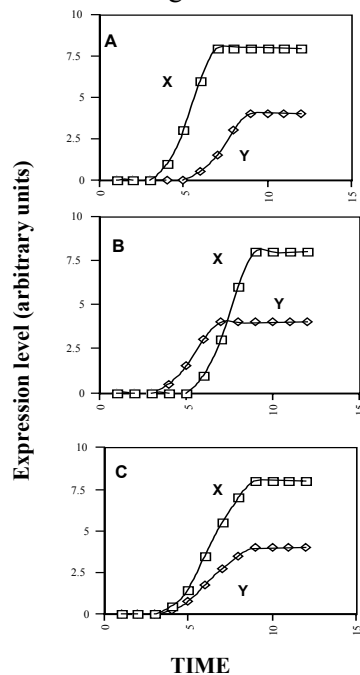


Figure 4.5. Time dependent expression changes of genes X and Y in two individual stochastic genetic networks (figures A and B) and the experimentally measured, population-averaged values (figure C). As described in the text, stochastic networks can get from a given gene-expression state to another one via several alternative paths. In the genetic network represented in figure A, gene X is activated before gene Y. Another cell (figure B) might follow a gene-expression path in which the sequence of the activation of these genes is reversed, even if gene X and Y are in regulatory interaction. Experimentally, we can measure only the population-averaged values. Based on the expression kinetics displayed by the genetic networks in figure A and figure B, both genes will show the same time dependence of gene-expression changes, although the maximum level of expression remains different.

4.10. Forward modeling or genetic network simulations.

(also known as the "The ultimate goal of computational biology" - or the "in silico" cell.)

This is one of the trickiest and riskiest part of genetic network analysis with a potentially significant return (IF IT WORKS ?????!!!!!!!). This approach is certainly not for the weak-hearted and

should not be attempted without a projected continuous support for many years ahead. Therefore, this field is mainly practiced by either industry (Entelos, Physiome Sciences, Gene Network Sciences etc.) or by government backed programs such as the "e-cell project" in Japan. To be absolutely fair, we must mention that intrepid individuals published nice reports as well, such as Eric Mjolsness on the MAP-kinase pathway. (In an ideal world, forward modeling of genetic networks could be envisioned as a distributed effort amongst all interested scientists if an appropriate organization, e.g. NIH, provided a reliable, centralized computational and maintenance environment, and would also ensure the high quality of data input.) At the moment, (based on their marketed products) Physiome, Entelos and e-cell is focusing more on physiological and biochemical processes that can be described with continuous differential equations. They have wisely avoided the murky issues of stochasticity that will certainly reemerge as scientific groups or companies, such as Gene Network Sciences or Molecular Mining, will attempt gene regulation based network modeling.

Forward modeling is exactly what is suggested by its name. It starts with building a regulatory architecture of interacting elements where the nature and parameters of interaction is extracted from the literature (Yes, there is hope for re-employment for biologists after massively parallel biology takes away the job of hard-working bench scientists.) It would, of course, be nice to employ natural language processing for this job, but these approaches seem to be quite hopeless at the moment and we need to resort to "human experts". It is also possible to fill in probable regulatory interactions by other means, for example exploiting reverse engineering as described in that chapter of this tutorial. Once a more or less complete regulatory network is created, it is fed into an appropriate modeling environment (more on that later). Then an initial state of parameters can be set and the model will produce a time-series read-out of the parameters of the regulatory network. Given a reliable functioning model one can envision a whole array of exciting questions. What will happen with the cell if a given gene is perturbed? What do we need to change in terms of over-expression or knock-out of genes to change the behavior of the cell? In short, one can run experiments "in silico" and obtain testable hypotheses for further experiments. The real excitement will come from the "killer problems", such as searching for combinatorial therapy. Currently we have no way of predicting how gene expression patterns will change if two drugs are co-applied, even if we know precisely the gene expression patterns induced by the individual agents. A successful forward modeling environment would be able to do that. Then one can search for combinations of drugs with known gene expression pattern changes that will produce a desired gene expression pattern, which could not be induced by any single compound. This may sound like "science fiction" at the moment, but projects like this could be the ultimate pay-off of bioinformatics. (It should also be noted that it does not really matter what looks sci-fi now - after reaping the low hanging fruit of massively parallel biology biotech will need to escape forward and try even the most far-fetched ideas. The huge steam engine of biotech is gaining more and more momentum and few will dare to stop.)

Let us take a look at the details.

1. Creating the regulatory architecture:

Currently, (as I mentioned above) the best way to achieve this is expert-based input. This means a group of humans who will read the relevant literature, identify the regulatory interactions between two elements of the network and also fill in all other details such as kinetic constants of a given regulatory interaction. This sounds like a lot of work, but may not be hopeless: In our work-horse model, the NK genetic network (N genes with K regulatory inputs per gene) the number of interactions to feed into the model is on the order of NK. A good expert, if pushing very hard, can put in 5-10 interactions/day. This would require on the order of NK/10 man-day effort to build a model given an ideal literature to start with. Given the contradictions and ambiguities of scientific papers, NK/5 is probably a better estimate. There is little doubt that most scientists would be interested to see the results produced by a network of 1000 genes - this would take on the order of a 500-1000 man-day effort which is about a year's worth of effort for an appropriately managed group of 6-10 experts. This does not seem too far-fetched so far.

Of course, these regulatory architectures should be built following the common sense guidelines of large scale modeling. For example, libraries of entries and interactions should be created separately and be accessed independently. Therefore, later corrections can be administered easily.

2. Building the modeling environment:

This involves choosing the types of equations to model the regulatory interactions. Continuous differential equations (diff.eq.) are commonly used for obvious reasons. In addition to that, there are two other methods that will certainly be given serious consideration. First, stochastic differential equations: As we have seen before, gene regulatory interactions are stochastic processes. Therefore, from a scientific point of view stochastic diff. eq.'s are a much better description than the continuous ones. Their major drawback is that they are computationally much more expensive. One of the most important questions of theoretical genetic network analysis is determining whether in living systems, the stochastic processes could be for all practical purposes efficiently approximated with continuous equations due to for example the robustness of genetic networks. Second, dynamic Bayesian nets: the "black box" type approach of creating time series changes. In theory, in this approach the actual details of biochemical/ molecular biological regulatory interactions could be altogether dispensed with. (This is only to highlight the nature of this approach and not what I am proposing to do.) Dynamic Bayesian Nets operate on the principle that the state of a given member of the network will cause a state of another member (or itself) with a certain probability. When an appropriate time-step is incorporated into the system, the probabilistic network of causative interactions will result in time series changes similar to the one produced by differential equations. The probabilistic description of a regulatory interaction can be derived from the actual biochemical underlying mechanism but could also be added by experts (their "belief") or generated by reverse engineering when fitting to actual time series measurements. Their greatest advantage is the possibility of incorporating assumed regulatory interactions when the biochemical understanding is insufficient.

Equations are not hardwired with the kinetic constants and interacting biochemical entities, but rather created by an automatic "equation generator". This provides much desired flexibility. The modeler needs to provide only a set of interacting entities and then the equation generator takes care of everything else - if the model is updated, one does not need to rewrite any of the equations, but only to provide an updated list of regulatory interactions.

3. Robustness of forward modeling:

One of the most often cited criticism against forward modeling is the uncertainty of parameters. Interestingly, while biologists do not worry about the precision of kinetic constants while working on a handful of enzymes, they become fully aware of error amplification when massively parallel forward modeling is at hand. In other words, how much do we need to worry about the facts that kinetic constants are derived from some spiked free solution measurements, and things may be quite different in the cell? This, very appropriate, issue is addressed by a two step process in forward modeling. First, if time dependent measurements are available from the modeled regulatory network, then the parameters can be corrected in order to better fit the measured values. This may be one of the key criteria based on which you may prefer one modeling environment over another one. Fitting the kinetic constants to the data will certainly require arduous numerical approximations. There are many smart ways of searching the "rugged landscape" involved in these approximative processes and speed and overall precision will distinguish a good program from a mediocre one. Second, modeling processes should be routinely tested for robustness, i.e. how sensitive are the results to small changes in the input value of kinetic parameters. Robustness can be quantified and a desire level of it may indicate that the overall topology of the regulatory network has been correctly described.

An efficient modeling environment is expected to predict time series data accurately under a variety conditions/perturbations. Interestingly, even if this is achieved, the scientific value of "in silico" biology may still remain questionable. It is not enough that modeling provide accurate predictions, but

Similarly to genetic network modeling, reverse engineering have been studied both on **continuous** and **discrete** systems .

5.2. Reverse engineering on continuous data

Reverse engineering approaches for continuous data are essentially variations on a theme. All attempts start with the assumption that the expression level of gene i at a given time point, $x_i(t+1)$, correlates with the weighted sum of a subset of gene expression levels at the previous time point, $\sum w_{ij}x_j(t)$ according to a certain equation g . (This equation is usually introduced to make the approach look more scientific)

$$(1) \quad x_i(t+1) = g(b_i + \sum w_{ij}x_j(t))$$

where w_{ij} is the weight of gene i 's influence on gene j and b_i is a base-line synthesis constant. The function g can be either a simple linear correlation as in the linear model of D'Haeseleer et al. (1999) resulting in:

$$(2) \quad x_i(t+1) = b_i + \sum w_{ij}x_j(t)$$

or a more "biology-like" dose-response function such as:

$$(3) \quad g(z) = 1/(1+e^{-kz})$$

that was used in "the weight matrix model" by Weaver et al. (1999) or in "coarse-grained reverse engineering" by Wahde & Hertz (1999). Weaver et al showed that this sort of quasi-linear model can be solved by linear algebra as well, by first applying the inverse of the squashing function:

$$(4) \quad g^{-1}(x_i(t+1)) = \sum w_{ji} x_j(t) + b_i$$

They also showed that randomly generated networks can be accurately reconstructed using this modeling technique.

Mjolsness, Reinitz and Sharp (1991) have used a similar approach to model small gene networks involved in pattern formation during the blastoderm stage of development in *Drosophila*. They added a simplified cellular model, with synchronized cell divisions (cell divisions are under the control of a maternal clock at this stage) along a longitudinal axis, alternated with updating the gene expression levels. Because of the more complex hybrid model, simulated annealing was used to find a least-squares fit to real gene expression data. The model was able to successfully replicate the pattern of *eve* stripes in *Drosophila*, as well as some mutant patterns on which the model was not explicitly trained.

Various groups have coined different names for this sort of models: connectionist model (Mjolsness, Reinitz and Sharp), linear model (D'haeseleer), linear transcription model (Chen et al), weight matrix model (Weaver et al). Considering the core of these models contain a weighted sum to implement gene regulation it was (very appropriately) suggested, to call them "additive models" (D'Haeseleer et al., 1999).

In all of the above cases, reverse engineering is reduced into determining all " w_{ij} " and " b_i " values.

The number crunching can be performed by well established techniques such as:

- Genetic algorithms (Wahde & Hertz, 1999)
- Solving weight matrices (singular value decomposition etc.) (Weaver et al., 1999)
- Least square fit for the linear modeling (D'Haeseleer et al., 1999) etc.

(It should be noted that these methods usually require at least as many time points as genes i.e. $T-1 > N+2$.) or probably by other computational methods as well such as:

- Dynamic Bayesian Networks (Murphy & Mian, 1999)

There is no clear winner yet.

5.3. The information requirement of successful reverse engineering

The amount of data (and the consequential experimental cost) depends on the actual genetic network. Table 5.3. provides a list of the useful information content, in terms of independent time points or gene expression states, that is required to perform successful reverse engineering on a given type of genetic network. (The network consists of N genes and the average number of regulatory inputs per gene is K .)

<u>How much information is needed for reverse engineering? (in terms of independently measured gene expression states)</u>	
Boolean fully connected	2^N
Boolean, connectivity K	$K 2^K \log(N)$
Boolean, connectivity K, linearly separable rules	$K \log(N/K)$
Continuous, connectivity K, additive	$K \log(N/K)$
Pairwise correlation	$\log(N)$
N = number of genes	
K = average regulatory input/gene	

Table 5.3. Orders of complexity for Boolean networks

For details (especially if you are craving for some real math complete with theorems etc.) see the work of Akutsu et al (1999, 2000) on discrete nets.

For continuous networks the solution looks deceptively simple, although it took some heavy-duty math to derive it (Hertz, 1998).

Unfortunately, the story is not complete with the elegant mathematical solutions. The above quoted papers implicitly assume that the gene expression space is relatively freely and randomly sampled. This is probably very far from true. For example, "knock-out" or "knock-in" experiments are expected to yield a vast portion of the required data. In these time-dependent (for example along the cell cycle) gene expression measurements are performed before and after a given gene has been removed or constitutively overexpressed in a given cell type. We know, that about 83% of genes can be knocked out from yeast without fatal harm to the organism (i.e. in these cases we can obtain data). BUT we do not know how much information is carried by the 17% of genes that cannot be removed without killing the organism relative to the other 83%. In addition, we have little idea about how strongly connected are the genetic networks before and after knocking out a gene. In other words it might very well happen that the gene expression series before and after removing a gene will be very similar because the gene to be knocked out is at the end of a regulatory sub-network. (That is why it can be removed). Taken together, the results in Table 5.4. determine the required number of independent gene expression states BUT there is no guarantee that biological systems will provide that much information. In fact our ability to perform reverse engineering will depend on at least four factors:

1. the stochastic nature of genetic networks ,
2. the effective size of genetic networks ,
3. the compartmentalization of genetic networks,
4. the information content of gene expression matrices,

As a first approximation we can say that time dependent gene expression measurements tend to yield 1-2 order of magnitude less information than expected in an ideal case. (for details see Szallasi, 1999)

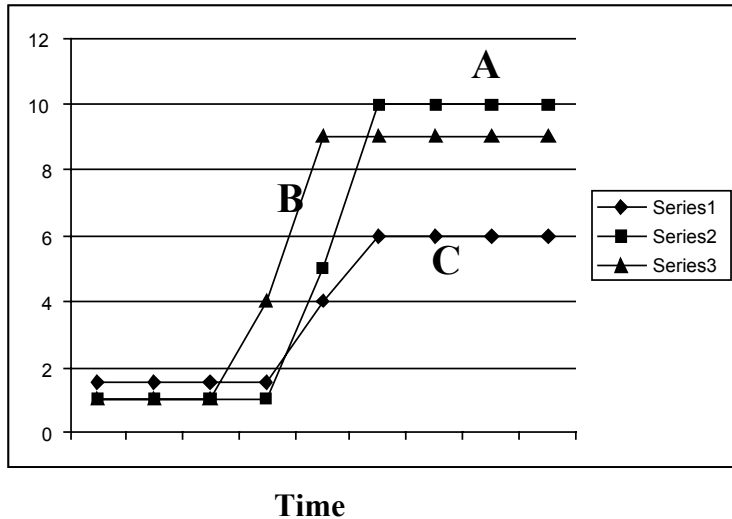


Figure 5.4. Time series measurements for 3 genes. The three genes show a correlated pattern that could hint at control structures.

5.4. Correlation analysis

We will also discuss here an elegant application of correlation analysis for reverse engineering. Observation of correlations between variables has long been used in biology to predict causal relationships. Although correlation can never be the proof of a causal relationship, it can lead us to propose hypotheses that can be tested by other means. In terms of gene regulation, a high correlation (or anti-correlation) between A and B can be caused by (1) gene A regulating gene B, (2) gene B regulating gene A, (3) gene A and B being co-regulated by a third gene C, or (4) accident. Of course, all of these regulatory interactions can be indirect, through one or more intermediates. Nevertheless, a sufficiently high correlation between two genes (taking into account number of data points, error levels on the data, general regulation trends, etc.) warrants an investigation of the genes in question.

Figure 5.4. shows the time dependent concentration changes of three substances, (A, B and C) in a complex chemical reaction. If a chemical reaction takes 1 unit of time, then the "B producing A" reaction will be a more likely candidate than the "C producing A" reaction to explain the observed time dependent changes.

Arkin et al (1997) used a similar way of thinking in order to reverse engineer the "reaction path network" in a multi component chemical system. The system (a reactor vessel with chemicals implementing glycolysis) is driven using random (and independent) inputs for some of the chemical species, while the concentration of all the species is monitored over time. First, the time-lagged-correlation (cross correlation) matrix is calculated according to equations:

$$(1) S_{ij}(\tau) = \langle [x_i(t) - \bar{x}_i] * [x_j(t + \tau) - \bar{x}_j] \rangle$$

and

$$(2) r_{ij}(\tau) = \frac{S_{ij}(\tau)}{\sqrt{S_{ii}(\tau) * S_{jj}(\tau)}}$$

where

$\langle \dots \rangle$: designates the time average over all the measurements

$x_i(t)$: t-th time point of the time series generated for species i

\bar{x}_i : time average of the i-th time series.

The correlation matrix will show how much does a change in the level of species "i" correlate with a change τ time later in the level of species "j" ? From this a distance matrix is constructed based on the maximum correlation between any two chemical species. This distance matrix is then fed into a simple clustering algorithm to generate a tree of connections between the species. To visualize the results, the chemical species and the tree connecting them is displayed using multidimensional scaling (MDS), mapping each species to a point in 2D space while trying to preserve the distances between each prescribed in the distance matrix. It is also possible to use the information regarding the time lag between species at which the highest correlation was found, which could be useful to infer causal relationships.

6. Classification and cluster analysis in gene expression matrices

Cluster analysis is probably the best-known part of genetic network analysis for at least two reasons: First, it is relatively easy to perform (Import your data matrix, then click a button). Second, it always yields something that looks like a result. (See key messages 1 and 2 at the beginning of this tutorial).

The aim of cluster analysis is dividing the genes and/or samples (i.e. rows or columns) into subgroups in a manner that the elements in the same cluster are always highly similar to each other (or more similar than a threshold) AND elements from different clusters always have a low similarity (or are less similar than a threshold). This is easier said than being done. (As so often the case in computer sciences, in order to solve clustering problems one should often either cheat or the computation will take too long. BUT, as you will see it, there are some smart ways around the problems.)

Once a clustering of sufficient quality is achieved, one can attach meaning to it. For example, co-clustering of genes in time dependent measurements may help to reveal mutual regulatory inputs, shared promoter binding sites, or suggest thus far new functions. Clustering of tumor samples may help to identify thus far unrecognized tumor subclasses.

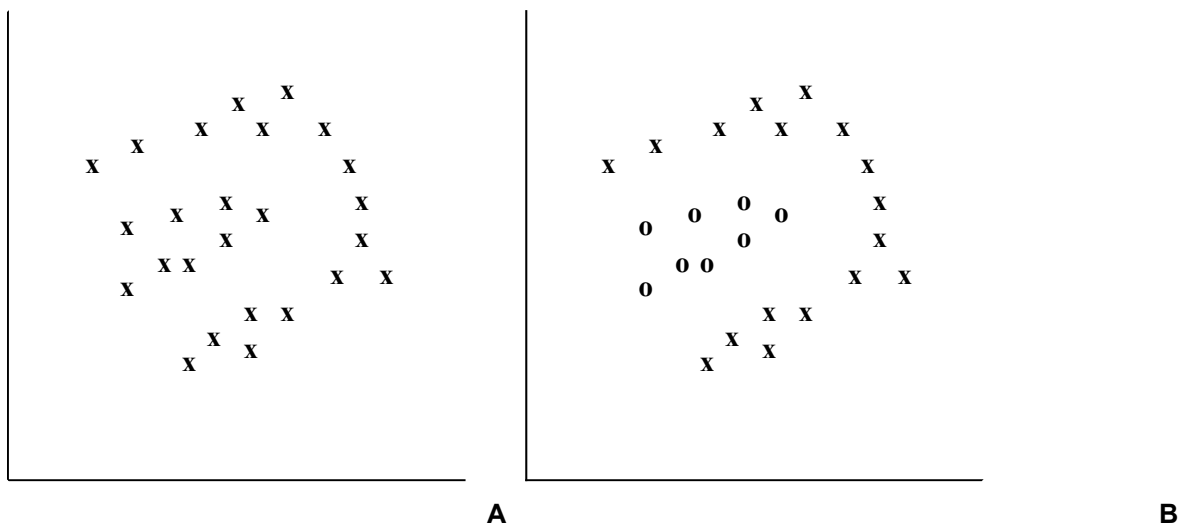


Figure 6-1 In the left window is a representation of samples with two features or variables: a traditional scatterplot. On the right window, the points are the same, but they are each identified as belonging to a *class* or *cluster*. The goal of clustering is sometimes seen as identifying the class structure that arises “naturally” in data.

Figure 6.1. is an intuitive example that may help to demonstrate the gist of several key issues in cluster analysis, and as such I will shamelessly exploit it throughout this part of the tutorial. The left hand-side figure represents a set of measurements on the expression levels of two genes corresponding to the horizontal and vertical axis. Every point represents a set of measurements from a different sample (imagine e.g. a cell line with a given phenotype). After plotting the data we could ask the question whether there is any reason to assume that there are more than one types of cell lines incorporated in the study based on the gene expression data. We might be inclined to say yes, since we can easily classify the data points visually into two clusters: a central blob surrounded by a horseshoe. (as demonstrated on the right hand side panel where data points of the central blob turned from crosses into circles.)

Before one discards the horseshoe-and-blob example as irrelevant, I would like to mention that it is not an accident that I brought up this very example. Contrary to popular belief, XOR (exclusive OR) type rules

seem to exist in biology even at the level of individual gene regulation, which may create clustering problems of this type.

Creating a computer algorithm to identify such finicky shaped clusters is quite difficult, although we can do this very easily visually. Now, imagine doing the same in three dimensions and then in a hundred dimensions. The same visual pattern recognition quickly turns into a very difficult problem, which is often termed as *The curse of dimensionality*: (Although the expression, which was first used by Huber if I am not mistaken, had originally a slightly different meaning. It described the fact that high-dimensional data sets are often too sparse to find the interesting substructures in them.)

The Curse of Dimensionality in our case simply means that a task which is easy to perform in a low-dimensional space becomes exponentially more difficult with increasing dimensionality i.e. with increasing the numbers of measured parameters. This underlines the importance of reducing the number of dimensions as we will discuss later.

As we have seen on figure 6.1., we have a good reason to assume that there might be two types of cells in our study. This assumption is coming from the well-distinguishable shapes of clusters. The existence or lack of such well distinguishable clusters is determined by the **internal data structure**. In our simple example we can visually confirm the presence of the "blob" and the "horseshoe", but again, what shall we do with an N-dimensional data set. This leads to the question whether biology, as we understand it, supports the existence of certain features of the internal data structure. For example, it is not necessarily true that different types of cancers are associated with isolated gene expression clusters. It could just as well be true that gene expression measurements in cancer (considering the significant noise of measurements) will provide a continuous gene expression "blob" with protrusions in certain dimensions. Currently, we have little idea whether biology will dictate any specific internal data structure. Maybe modeling efforts, e.g. exploiting the idea of attractor transitions in cancer will reveal key features of gene expression patterns. I would like to point out that the "continuous blob" -version does not mean the end of exploiting gene expression measurements in cancer, but it will certainly influence the choice of analytical tools. For example, in this case supervised learning methods will probably perform better than unsupervised ones, since the latter ones often assume the presence of some easily recognizable data structure e.g. nice, isolated and often spherical clusters.

Meaningful cluster analysis of gene expression data, therefore, requires the solution of two problems: first, performing a "mathematically correct" clustering algorithm and second, translating the clustering results into biological knowledge. ("Mathematically correct" roughly means that objects in the same cluster are more similar to each other than to objects in any other clusters, and also that you would not expect those clusters to show up by chance.)

Computer scientists have several decades of advantage over biologists in this field, therefore it does not come as a surprise that we are much better prepared to solve the first problem.

6.1 Cluster analysis of gene expression measurements requires three steps:

1. Preprocessing data, e.g. certain clustering algorithms require the generation of a similarity or distance measure that quantifies the similarity (or lack thereof) of the expression patterns of gene pairs along a whole series of measurements. Preprocessing also deals with reducing the dimensionality of data sets. (Of course you can also generate similarity measures between gene triplicates etc. but that is rarely done)
2. Determining the clusters of genes in a "mathematically correct way" and generating a visual representation of the clustering, e.g. a dendrogram, in which genes showing similar expression patterns will be close to each other.
3. Determining the biological meaning of co-clustering.

6.2 Data Preprocessing: similarity measures and reducing dimensions

Many clustering algorithms require the generation of some sort of similarity measure. This may create the first problem since a whole series of data points are getting replaced by a single number, the similarity measure. Furthermore, as you will see below, similarity/distance measures are not created equal.

The most frequently used similarity measures are (a non-exhaustive list):

Euclidean distance (which is the actual distance between two points in an n-dimensional space), is computed as the square root of the sum over all genes (samples) of the squared differences between expression levels.:

$$D_{\text{Euclidean}} = \sqrt{\sum_1^N (x_i - y_i)^2}$$

City-block (Manhattan): The distance between two samples (genes) is computed as the sum over all genes (samples) of the absolute value of the differences between expression levels:

$$D_{\text{Manhattan}} = \sum_1^N |x_i - y_i|$$

Chord: The vectors representing samples (genes) are normalized to have unit length. The distance between two samples (genes) is then computed as the Euclidean distance between the unit vectors. The total counts in any column (row) are therefore irrelevant for the chord distance. This is as if we were measuring the length of a cord in an n-dimensional sphere.

Infinity norm (Chessboard): The distance between two samples (genes) is computed as the maximum over all genes (samples) of the absolute value of the differences between expression levels.

$$D_{\text{infinity norm}} = \max_1^N (|x_i - y_i|)$$

Pearson: The Pearson correlation coefficient is calculated as:

$$r = \frac{\sum_1^N (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\frac{\sum_1^N (x_i - \bar{x})^2}{N}} \sqrt{\frac{\sum_1^N (y_i - \bar{y})^2}{N}}}$$

for similarity measure the following distance is used:

$$D_{\text{Pearson}} = 1 - r^2$$

Spearman: The Spearman correlation replaces the exact expression values with rankings (non parametric), and then computes the Pearson correlation of the rankings, i.e.

$$r_{\text{Spearman}} = 1 - \frac{6 \sum_1^N D_i^2}{N(N^2 - 1)}$$

where D is the rank difference between gene/sample pairs.

Again, for similarity measure the following distance is used:

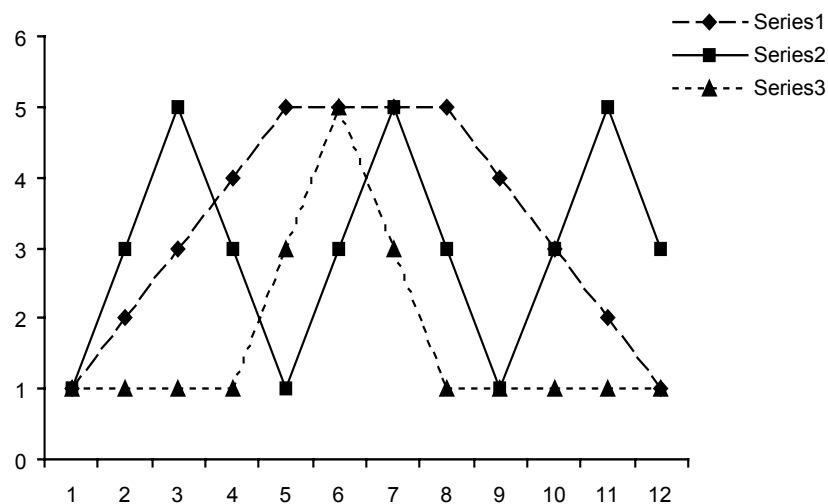
$$D_{\text{Spearman}} = 1 - r_{\text{Spearman}}^2$$

Mutual information (see chapter 5)

6.2.1 Similarity/distance measures are not created equal.

Let us consider now the experiment on figure 6.2, where the time-dependent expression level changes were measured for three genes. As you can see above, there is a variety of distance measures that could be employed: e.g. if we use the normalized Euclidean distance, then we will find that similarity measure for both gene pairs 1 & 2 and for gene pairs 1 & 3 is the same, namely 1.5. However, applying the Pearson correlation coefficient gives a different result, (giving a similarity measure of 0 for genes 1 & 2 ; and 0.6 for genes 1 & 3) showing a higher correlation between genes 1 and 3 than between genes 1 and 2. However, an unbiased biologist would say the following: Isn't it striking that gene 1 always changes its expression when gene 2 changes it expression the opposite direction (this is, of course, not true the other way around)? Maybe gene 2 has a direct input on gene 1 according to an AND function where the other input, gene X has not been identified yet. This seems to be a better working hypothesis than forcing a correlation between gene 3 and 1. Mutual information based distance would place genes 1 and 2 closer than genes 1 and 3.

Figure 6-2



Of course the question is: Which similarity measure is most relevant to biology? We do not have an answer for that, and this might not be a relevant question after all, at least not in this form.

Currently, the most prudent approach is to try several different similarity measures, which is usually offered by most commercially available clustering programs, and compare the results.

It may be worth developing algorithms that would test e.g. the robustness of results depending on the distance metric employed and therefore provide a “test of goodness” of the metric used. (Periwal, 2001, personal communication)

6.2.2 Reducing the dimensions

The second step of data preprocessing is **reducing the number of dimensions**. Projecting or mapping high dimensional data points into a lower dimensional space is a common tool to deal with the curse of dimensionality. During this approach the algorithm tries to retain certain features of the original data set.

For example in **multidimensional scaling** (MDS) the order of the distances between each pairs of data points is retained as much as possible. In **principal component analysis (PCA)** low dimensional subspaces are searched where most of the variance of the data is retained relative to the original data set. (In order to save space I would like to refer to some of several excellent on-line tutorials on these issues such as: <http://www.statsoftinc.com/textbook/stmulzca.html> and <http://www.analytictech.com/networks/mds.htm> on MDS, or <http://www.casaxps.com/FactorAnalysis.htm> on PCA.)

Reducing dimensions is a surprisingly rarely used tool, probably because of its inherent risk of losing important features of the data set. However, it has three significant potential uses:

- First, if we have a well behaving data set and most of its features can be summed up in significantly fewer dimensions, then the computational demand becomes lower.
- Second, we might understand something about the data. For example, imagine if one of the derivative dimensions contains only genes with similar functions. In this case one may attempt to attach a biological meaning to the joint involvement of that group of genes in a given biological process, e.g. malignant transformation.
- Third, we can always take a look at our clustering results at lower dimensions. In time dependent gene expression data, for example, the first two derivative dimensions account for 90% and the first three derivative dimensions account for 95% of the variance. (Raychaudhuri et al. 2000) One can easily check visually in so few dimensions the quality of a clustering result provided by a given algorithm. (e.g. the results of hierarchical clustering was checked by Raychaudhuri et al. 2000)

6.3. Clustering algorithms

The next step is the actual **Cluster analysis** of the preprocessed gene expression data.

There is an increasing number of papers on clustering gene expression data. (It seems that experts in the "hard sciences" finally found a tangible biological problem to work on.) In addition to the application of a whole array of "off-the-shelf" methods, several ingenious, novel algorithms were introduced. The large number of papers allows only a brief review of the most important ones of these efforts.

I will also offer a classification here that the "purists in computer science" may not subscribe to but which may give a good intuitive feeling about the nature of the listed clustering algorithms.

The first two of these papers, dated from historic 1998, used **bottom up hierarchical clustering**, which belongs to the first class of clustering algorithms, which could be called:

6.3.1. Obvious (or non-optimized) algorithms. These algorithms will perform a series of steps, that group similar objects together based on some common sense logic. For example in bottom up hierarchical clustering the two most similar objects are joined and replaced by their average. These algorithms proceed either for a specified number of times (e.g. in self-organizing maps) or until they run out of objects (e.g. in bottom up hierarchical clustering) **BUT** these algorithms, most of the time, do not look for an optimal solution or for some sort of measure of the goodness of the clustering. I.e. you will have no idea how far you are from an optimal solution.

6.3.1.1. Euclidean-distance based bottom up hierarchical clustering.

Wen et al. (1998) calculated the Euclidean distance of a gene expression matrix containing measurements over a series of time points. Then they fed this similarity matrix into the FITCH algorithm developed by Felsenstein (1993) for creating evolutionary trees.

6.3.1.2. Correlation coefficient based clustering

In a very similar effort Eisen et al. (1998) used a different metric, the Pearson correlation coefficient and then employed a greedy (non-optimized) version of Bottom up hierarchical clustering (running in time $O(N^2)$)

These papers were great first efforts, but one should be aware of the fact that in general there are many potential problems and pitfalls with non optimized bottom-up hierarchical clustering, including:

- Genes clustered next to each other are not hierarchically related
- lack of robustness
- genes can cluster together based on local decisions
- accidental features can be easily locked.

Gene expression matrices can be often reduced to a manageable size after appropriate pre-filtering (e.g. for a certain level of variance). Therefore, it might be worth getting hold of the fastest computer around and try an exhaustive version of hierarchical clustering, running in about $O(N^3)$. In the resultant tree you will find that the distance between each pair of genes/samples (i.e. the number of nodes through which you get from one to the other) will be proportional to the corresponding distance measure employed. The only problem with this approach is that it is often difficult to see the overall relative position of the different main clusters.

6.3.1.3. Top Down hierarchical clustering

The Deterministic-annealing algorithm was used to cluster time-averaged gene expression data from a series of tumor and normal tissues by Alon et al., 1999. In this, each gene, k is represented with a normalized vector (V_k), whose components correspond to the expression level of the gene in individual samples. The algorithm proceeds from top to down: The genes are split into two clusters around two cluster centroids. The probability of a gene belonging to a cluster is determined by:

$$P_j(V_k) = \exp(-\beta|V_k - C_j|^2) / \sum_j \exp(-\beta|V_k - C_j|^2)$$

and the centroids are determined by

$$C_j = \frac{\sum_k V_k * P_j(V_k)}{\sum_k P_j(V_k)}$$

which is solved by iteration.

Each gene is assigned to the centroid with the larger $P_j(V_k)$

These steps are repeated until all genes are clustered into a binary tree.

6.3.1.4. Self organizing maps (SOM)

Gene expression measurements are mapped into a k -dimensional “gene expression space” in which the i th coordinate represents the expression level in the i th sample, e.g. time point. (In this space each data point contains all the information whatever happened to that particular gene. I.e. no similarity measures are introduced). Start with a certain geometry of nodes (e.g. an $m \times n$ grid used by Tamayo et al., 1999), where each node represents the “center” of a cluster. The nodes are mapped into the gene expression space initially random THEN iteratively adjusted. Each iteration chooses a data point randomly and moves the nodes closer to it in a fashion that the nearest node is moved the most towards the data point and the rest of the nodes are moved less depending on their distance from the data point at the beginning of the iteration.

REPEAT this tens of thousands of times.

This is not a "bona fide" clustering algorithm, therefore it is very difficult to see where the algorithm is heading to, what the calculations has achieved when the program stops etc. Its great advantage is that it is running fast, but this is a certainly a method one should be very cautious about.

6.3.2. Optimization algorithms

These algorithms offer an advantage in the form of some sort of a measure that is getting optimized during clustering. For example, in the case of k-means the algorithm stops when the total distance of data points from the corresponding cluster center is minimal. This, at least, provides a good mathematical handle on what you have achieved by the algorithm.

6.3.2.1 K-means clustering

The widely used K-means algorithm was used by George Church's group (Tavazoie et al., 1999) to analyze the gene expression matrix derived from the whole transcriptome of *S. cerevisiae* under a set of various growth conditions. In this, one needs to make an assumption about the number of clusters (designated as K) then the algorithm keeps partitioning and repartitioning the data until the total distance of the data points from their corresponding centroids is minimized.

k-means is a popular and rather simple algorithm. Of course, this simplicity comes at a price:

- it can perform poorly with overlapping clusters,
- it lacks robustness to outliers
- it assigns each point to one and only one cluster.

This last point might lead to misleading clustering and it is often more informative to determine the probability of a data point belonging to each cluster. (Imagine a data point at the border of two clusters - k-means will assign this data point to only one of the two clusters, whereas e.g. the next algorithm below might assign the data point to one cluster with 49% and to the other with 51% probability.)

6.3.2.2. Gaussian distribution fit by Expectation-Maximization

Mjolsness et al (1999) proposed a "softer" version of the k-means clustering. Each set of clusters is modeled by a Gaussian distribution fit to the data by Expectation-Maximization algorithm using cross-validation to choose the optimal number of clusters. This can be considered a "mathematically correct" version of k-means, improving on some of the shortcomings of bottom up hierarchical clustering. Although it provides a probabilistic partitioning of the data points, it does not provide any estimate on how unlikely is the emerging clustering pattern.

6.3.3 Optimization by probabilistic considerations

These methods provide a further advantage over the algorithms of the previous class. To put it simply, they offer an estimate whether a certain emerging pattern during clustering is due to chance. These are often **newly developed clustering algorithms** or a fundamentally new application of previously established methods. They often employ smart shortcuts in computation and also circumvent some of the inherent problems of other algorithms such as the need for an "a priori" assumption about the number of clusters

6.3.3.1 CLICK (Cluster Identification via Connectivity Kernels) was developed by Ron Shamir's group. (Shamir and Sharan,2000) This takes a graph theory approach to clustering. They represent the similarity matrix as a weighted graph where the vertices are the elements (e.g. genes) and the weight of the edges correspond to the similarity measure between these elements. The algorithm keeps partitioning the graph by minimum weight cuts until a set of sub-graphs (clusters) is created that satisfy the criteria

that the average intra-cluster correlation coefficient is maximal and the average inter-cluster correlation coefficient is minimal (of course at the same time). The algorithm employs several smart short-cuts and tricks and as a consequence runs very fast. (It seemed to me running in time $O(N)$ based on the data in the paper).

6.3.3.2 Gene Shaving (Hastie et al.2000). This algorithm creates a sequence of nested clusters. At each step the largest principal component of the current cluster of genes is computed and a certain fraction of genes (say 10%), whose expression vector is not pointing in the principal components direction, is shaved off. In the next step each nested cluster is getting tested against a randomized set of data, whether they are due to chance by using a combined measure derived from the intracuster (V_W) and intercluster (V_B) variance:

$$\frac{\left(\frac{V_B}{V_W}\right)}{1 + \left(\frac{V_B}{V_W}\right)}$$

the most likely optimal cluster size is thus determined and in the next round each row of the whole data set is orthogonalized with respect to the average gene of the last "optimal" cluster and the algorithm runs through the whole procedure again. This algorithm runs fast as well. The choice of the size of the nested clusters seems somewhat arbitrary and I am not sure whether this algorithm provides any advantage over the last group of advanced clustering algorithms.

6.3.3.3 In a paper presented at PSB2001, Sasik et al. employed **percolation clustering** for dictyostelium development. The core idea is similar to the so-called "**superparamagnetic clustering**" by Blatt et al. (1996), which seems to work well on non-spherical clusters. It starts with a high-dimensional (n-dimensional) representation of the data points and points that are within a certain "d" threshold distance of each other are connected. As the d threshold is increasing first tight clusters emerge, then these clusters are connected into superclusters, then at the end, at a certain threshold, every point is connected. This is pretty similar to hierarchical clustering so far, including its major weakness: if two points belonging to two, otherwise separate clusters happen to be close to each other due to e.g. noise of the measurements, then these clusters will get connected for no good reason apart from the measurement error. In this case, the very fact that only those two points are connected between the two clusters, but within each cluster the rest of the points are multiply connected will give a strong indication of the "real" cluster structure (i.e. two isolated clusters connected by a pair of "noisy points".) This algorithm exploits a similar argument. First, the overall possible "cluster structure" is explored based on probabilistic considerations, then, based on this information, the actual tree is constructed avoiding merging clusters based on the proximity of only a few points that are probably placed close to each other due to noise or some other error. The nice feature of this algorithm (in addition to its speed) is that it seems to be able to handle "horseshoe-and-blob" type clustering problems. These two algorithms are your best bets to find complicated cluster shapes, none of the above described methods comes close to them in achieving that goal.

Conclusion: It is hard to see whether further algorithms will provide any advantage over the methods listed in 6.3.3. These algorithms are fast, based on probabilistic considerations and require almost no initial assumptions about the clustering (e.g. the "k" number of clusters). It seems that these algorithms provide ample analytical power to analyze gene expression data. Therefore, lack of biological results cannot possibly be blamed on computer scientists any more.

6.4 Biclustering

From the perspective of a practicing biologist there is an even more interesting problem than clustering: biclustering. This was recognized recently by Cheng and Church (2000), when they reintroduced this approach for the analysis of gene expression matrices. (This algorithm also ran under a couple of aliases such as “direct clustering” in Hartigan, 1972 and “box clustering” in Mirkin, 1996.) The ultimate goal of biclustering is finding a cluster of genes the status of which could best distinguish a cluster of samples from other samples. (The “bi” prefix obviously refers to the fact that this algorithm is searching for a cluster of genes and a cluster of samples simultaneously, i.e. clustering is done in two directions) The practical consequences of succeeding in doing so is of great practical use. Imagine, for example, that you have a large set of gene expression measurements in a wide variety of cancer samples or with a large number of different drugs. Biclustering would not only discover new subtypes of cancer but also would implicate the actual genes that are involved in determining the newly discovered cancer subclass. Similarly, biclustering could give you the marker genes for a given class of treatment mechanism, therefore devise a rational strategy to screen for drug leads with similar action.

The algorithm proposed by Cheng and Church does much more than simply reorganizing the matrix in both directions. (That was suggested and done by other groups as well such as Weinstein et al, 1997 or Alon et al, 1999) Reorganizing the matrix in both directions is essentially running two consecutive clustering algorithms. Biclustering, on the other hand, is running clustering in both directions at the same time, i.e. how clustering goes in one direction will have a continuous effect on clustering in the other direction. Finally, biclustering will cut the matrix as well into submatrices within which the behaviour of genes show a very good correlation with the samples involved. In fact, this is the way most biclustering algorithms operate. First, define a measure of how well the behavior of a subset of genes correlate with a subset of samples. Cheng and Church used the so called *mean residue score* for a submatrix with I rows and J columns calculated as:

$$H(I, J) = \frac{1}{|I||J|} \sum_{i \in I, j \in J} (a_{ij} - a_{i\cdot} - a_{\cdot j} + a_{\cdot\cdot})^2$$

where a_{ij} is the expression level in the i th row and j th column, $a_{i\cdot}$ is the mean expression in the corresponding column within the submatrix, $a_{\cdot j}$ is the mean expression in the corresponding row within the submatrix, and $a_{\cdot\cdot}$ is the mean expression in the submatrix. When this measure is low, then the genes show a rather coordinated behaviour within the submatrix of highly similar samples. All you need to do is to go through all possible submatrices and find the largest submatrices with the lowest corresponding mean residue score. Unfortunately, this is an NP-hard problem, therefore this approach can be executed only through greedy algorithms that do not guarantee the best global solution. Nevertheless, this is definitely worth trying, because you will undoubtedly end up with a “deeper” working hypothesis than by clustering only in one direction.

Initial rumors suggest that there might be a shortcut through the “NP-hardness” of the problem in a biologically meaningful way. Periwal at Gene Network Sciences suggested an (unfortunately) proprietary, deterministic algorithm that will produce a set of non-exclusive biclusters of highly coherent genes and samples based on the global analysis of the gene expression matrix. The relative merits of their approach remains to be seen but it is certainly intriguing to test their claim for the most efficient algorithm for finding biclusters.

6.5 Comparing different clustering algorithms.

When one comes up with a new algorithm, it is usually expected to compare it against other algorithms used on similar data sets. There is a whole array of possible criteria to be used depending on e.g. whether the true clustering is known or unknown. Shamir and Sharan (2001) compared CLICK to

several other algorithms including SOM and k-means, and found that this algorithm performed superior judged by the average and minimal intra-cluster correlation coefficient and the average and maximal inter-cluster correlation coefficients.

Another test for validating clustering algorithms, which is based on the jackknife approach, was proposed by Yeung et al (2001). They applied clustering algorithms to all but one experimental condition in a data-set and then used the left-out condition to assess the predictive power of the clustering algorithm.

It is recommended to visit Terry Speeds web-site for some sobering experience regarding microarray analysis. (<http://www.stat.berkeley.edu/users/terry/zarray/Html/>) They have also published a detailed comparison of different methods for the classification of tumors using gene expression data (Dutoit et al., 2000, technical report #576. Available from the above mentioned web site.)

6.6. Supervised versus unsupervised learning

So far we have discussed algorithms that do not take advantage of the information we have about the actual biological behavior of the individual samples represented by the data points (unsupervised methods). Let us consider figure 6.1. again. As we have mentioned, it is difficult to create an algorithm that would efficiently recognize the “horseshoe” and the “blob” hidden in the data. BUT what if we already know that the circles on panel B are coming from tumors and the crosses are coming from normal samples. In this case, the so-called supervised learning methods, such as support vector machines will perform extremely well (Brown et al. 1999). The price of this performance is obvious. You need to know ahead of time the phenotypic clusters, which is obviously difficult if you want to discover e.g. novel classes of tumors based on gene expression patterns.

6.7 The use of cluster analysis in biology

The last step is making sense out of clustering results. Here we will review a couple of concrete examples when cluster analysis was used to address important questions in biology.

Tavazoie et al., (1999) used k-means clustering in order to identify new sets of coregulated genes (regulons) and their (putative) cis-regulatory elements. They started with a previously published data set of time dependent expression of 6,000 genes (complete transcriptome) of *S. cerevisiae*. They performed k-means clustering on these time dependent data and found about 30 clusters (49-186 genes/ cluster) after repeated attempts. The clusters were enriched for genes with similar function; and timing of the peak of some clusters fits well the time when the activity of that gene is needed;

This is good news so far, but not very unexpected.

Then, they searched for upstream DNA sequence motifs that are common to members within a cluster using the AlignACE algorithm from the same group (Roth et al., 1998). They have found 18 short sequence motifs in 12 clusters that were highly represented within their own clusters but were absent or very rarely detected in other clusters, i.e. most of these motifs were highly selective for the clusters they were identified from.

This is good.

Furthermore they have showed that “tighter clusters” show a good correlation with the presence of significant motifs.

This is exciting !

This is a nice example how two computational methods, i.e. cluster analysis and motif search by sequence alignment can complement each other's shortcomings.

In another paper **Golub et al., (1999)** used self organizing maps (SOM) in order to (re)predict different classes of leukemias. A two centroid SOM successfully reproduced the acute lymphoid leukemia (ALL)/ acute myeloid leukemia (AML) classification.

A four centroid SOM suggested an AML/T-lineage ALL/ B-lineage ALL/B-lineage ALL clustering suggesting a three centroid reclustering into AML/T-ALL/B-ALL.

This is nice but not very unexpected - after all, we expect this much from alternative differentiation states - (There MUST be a certain set of genes that will distinguish between myeloid and B lymphoid cells.etc.)

Here comes the big test: When they tried to correlate SOM classification with the clinical response of AML patients (responsive and non-responsive to chemotherapy) they failed to find any strong correlation between the gene expression patterns and the clinical behavior.

Of course, it is too early to give up hope. There are plenty of explanations that can be blamed for this temporary failure: E.g. The phenotype in question it is caused by a group of genes (e.g. multi-drug resistance gene) that is not on the chip.

An interesting theoretical consequence can be inferred in cancer biology from a clustering paper by **Perou et al. (2000)** They have performed cDNA microarray measurements on a series of surgically removed breast tumors. Twenty tumors were sampled twice, before and after the patient underwent chemotherapy. In addition, gene expression measurements from two primary tumors were also paired with their corresponding lymph node metastases. Not surprisingly the gene expression patterns from different tumors were rather distinct. However, classification by hierarchical cluster analysis showed that tumor samples removed from the same patient, despite presumed profound perturbation by chemotherapy, were always much more similar to each other than to any other sample from another patient. The same findings applied to primary tumors and their metastases. These results suggest that once a tumor reaches a stable gene expression state, it cannot be driven to a significantly different state by chemotherapy or by the dispersion of primary tumors. For those who are familiar genetic network modeling and especially the early efforts of Kauffman and coworkers (Kauffman 1971) this must ring a bell. The tumor samples display the behavior that is expected from gene-network attractors. During the initial steps of malignant transformation the genetic network of a cell undergoes a major perturbation leading to an unstable state. From here, according to theory, the cell will search the “gene expression space” in order to find a stable yet dynamic state of the genetic network. This stable state has been termed “attractor”. The theory further postulates that the attractor will stabilize the gene expression pattern of the cell, as further perturbations will induce feedback regulatory mechanisms that will return the cell to this newly found stable state. Therefore, in this case clustering analysis provided initial proof for an interesting hypothesis in tumor biology.

7. Generative models in the analysis of gene expression matrices.

The analysis of gene expression matrices by statistics or clustering algorithms will require the assessment of what is likely or unlikely to appear in a given data set. This may not be such an easy question to answer and will certainly require that we take into consideration the internal data structure of gene expression matrices. In order to highlight the importance of this issue I would like to start with the following introductory problem. Given a set of cDNA microarray measurements on breast cancer cell lines we have observed a high diversity of gene expression profiles. On average, 10% of all, "N" genes measured was mis-regulated (i.e. up- or down-regulated) in any cell line relative to non-malignant cells. At the same time, after E cell lines measured we still had K genes that were consistently mis-regulated. Could this K mis-regulated genes be due to chance because of the high diversity of gene expression patterns? As a first approximation, one can translate this question into a straightforward combinatorics problem: Let us pick M elements randomly and independently out of N elements in E consecutive experiments. How likely is it that at least K elements will be picked in all E experiments? For the detailed solution of the problem see (Wahde et al., 2001). However, one should be aware of the fact that the above mentioned combinatorial calculation is applicable only if the mis-regulated genes are randomly and independently selected. There are many handy ways to assess whether this is true for a given gene expression data set and in our case it was clear (Klus et al., 2000) that genes in breast cancer are not independently mis-regulated (Klus et al. 2001). We have further demonstrated that the unjustified assumption of random and independent selection of mis-regulated genes may lead to errors of several orders of magnitude in the statistical analysis of cancer associated gene expression matrices (Wahde and Szallasi, 2000). Therefore, it seems that simple randomization of data matrices is an incorrect way of assessing the significance of observed separators or clusters. A more appropriate approach uses random matrices that retain the internal data structure produced by e.g. the co-regulation of genes. For example, our generative model (Wahde and Szallasi, 2000) creates random matrices with a pair-wise mutual information distribution similar to the one observed in the original data set.

8. Systems approach to genetic networks and biology

Systems biology approaches genetic networks from an engineering point of view. (The difficulties lying ahead justify just about any approach.) It is attempting to understand the principles of gene regulatory networks starting from control theory. What overall design principles ensure robustness? What is the ratio of the number of control parts vs. the number of effectors? In engineering stability is achieved by feedback, redundancy and modular design - we suspect that the very same principles are at work in living systems - is there anything more to the design of genetic networks? (Kitano, 2000)

One can also add other interesting questions such as: What are the implications of robustness in biology? Does it preclude efficient reverse engineering?

This is an emerging field in biology that is certainly worth following. (It has a yearly meeting under the title of International Conference on Systems Biology, which brings together researchers from all fields of science, from pure mathematicians to molecular biologists, in a quite fruitful way - no minor feat in itself.)

9. Bibliography

Publications by the "tutor" can be downloaded from: www.usuhs.mil/pha/faculty/zoltan.shtml

- Szallasi, Z., and Liang, S. Modeling the normal and neoplastic cell cycle with "realistic Boolean genetic networks": Their application for understanding carcinogenesis and assessing therapeutic strategies. Pacific Symposium on Biocomputing. 3:66-76, 1998.
- Szallasi, Z. Gene expression patterns and cancer. Nature Biotech. 16:1292-1293, 1998.
- Szallasi, Z. Genetic network analysis in light of massively parallel biological data acquisition. Pacific Symposium on Biocomputing. 4:5-16, 1999.
- Klus, G., Song, A., Schick, A., Wahde, M. and Szallasi, Z. 2001 Mutual information analysis as a tool to assess the role of aneuploidy in the generation of cancer-associated differential gene expression patterns. Pac. Sym. on Biocomput , 6:42-51, 2001.
- Wahde, M. and Szallasi Z. Generative model based analysis of cancer associated gene expression matrices. Proceedings of the First International Conference on Systems Biology. Pages: 39-45. 2000
- Klus, G.T., Bittner, M.L., Chen, Y, Wahde, M. and Szallasi, Z. Use of overall quantitative features of cDNA microarray measurements in cancer research. (2000) (Technical paper #1 at www.usuhs.mil/pha/faculty/zoltan.shtml)
- Wahde, M., Klus, G., Bittner, M., Chen, Y and Szallasi, Z. Assessing the significance of consistently mis-regulated genes in cancer associated gene expression matrices. Bioinformatics. (submitted)

Overview

- Kauffman, S.A. (1993) The Origins of Order, Self-Organization and Selection in Evolution. Oxford University Press.
- Bryant, A. Milosavljevic and R. Somogyi (1998) Gene Expression and Genetic Networks. Pacific Symposium on Biocomputing 3:3-5 (1998).
- Somogyi R, Sniegowski CA (1996) Modeling the complexity of genetic networks: understanding multigenic and pleiotropic regulation. Complexity 1(6):45-63.
- Kitano, H. (2000) Perspectives on systems biology. Proceedings of the First International Conference on Systems Biology. pages: 3-21.

Data

- Anderson, L., and Seilhamer, J. (1997) A comparison of selected mRNA and protein abundances in human liver. Electrophoresis 18(4):533-537.
- Appel, R. D., Sanchez, J. C., Bairoch, A., Golaz, O., Miu, M., Vargas, J. R., and Hochstrasser, D. F. (1993) SWISS-2DPAGE: a database of two-dimensional gel electrophoresis images. Electrophoresis 14(11):1232-1238. (<http://www.expasy.ch/ch2d/ch2d-top.html> and <http://www.expasy.ch/ch2d/2d-index.html>)
- DeRisi, J. L., Iyer, V. R., and Brown, P. O. (1997) Exploring the metabolic and genetic control of gene expression on a genomic scale. Science 278:680-686. (<http://cmgm.Stanford.EDU/pbrown/explore/>)
- Velculescu, V. E., Zhang, L., Zhou, W., Vogelstein, J., Basrai, M. A., Jr., D. E. B., Hieter, P., Vogelstein, B., and Kinzler, K. W. (1997) Characterization of the yeast transcriptome. Cell 88:243-251. (<http://www.sagenet.org/yeast/yeastintro.htm>)
- Zhang, L., Zhou, W., Velculescu, V. E., Kerm, S. E., Hruban, R. H., Hamilton, S. R., Vogelstein, B., and Kinzler, K. W. (1997) Gene expression profiles in normal and cancer cells. Science 276:1286-1272. (<http://welchlink.welch.jhu.edu/~molgen-g/home.htm>)
- Wen X., Fuhrman S., Michaels G.S., Carr D.B., Smith S., Barker J.L., Somogyi R. (1998) Large-Scale Temporal Gene Expression Mapping of CNS Development. Proc Natl Acad Sci 95:334-339.
- P. Spellman, G. Sherlock, M. Zhang, V. Iyer, K. Anders, M. Eisen, P. Brown, D. Botstein, and B. Futcher (1998) Comprehensive Identification of Cell Cycle-regulated Genes of the Yeast

Saccharomyces cerevisiae by Microarray Hybridization. *Molecular Biology of the Cell* 9:3273-3297. (<http://genome-www.stanford.edu/cellcycle/>)

- Cho RJ, Campbell MJ, Winzeler EA, Steinmetz L, Conway A, Wodicka L, Wolfsberg TG, Gabrielian AE, Landsman D, Lockhart DJ, Davis RW (1998) A genome-wide transcriptional analysis of the mitotic cell cycle. *Mol Cell* 2(1):65-73. (<http://genomics.stanford.edu/yeast/cellcycle.html>)
- Araceli M. Huerta, Heladia Salgado, Denis Thieffry, Julio Collado-Vides (1998) RegulonDB: a database on transcriptional regulation in *Escherichia coli*. *Nucleic Acids Res* 26(1):55-9. (http://www.cifn.unam.mx/Computational_Biology/regulondb/)
- Perou et al. (2000) Molecular portraits of human breast tumours. *Nature* 406:747-752

High throughput methods

- DeRisi, J. L., Penland, L., Brown, P. O., Bittner, M. L., Meltzer, P. S., Ray, M., Chen, Y., Su, Y. A., and Trent, J. M. (1996) Use of a cDNA microarray to analyze gene expression patterns in human cancer. *Nature Genetics* 14:457-460.
- Fields, S., and Song, O. (1989) A novel genetic system to detect protein protein interactions. *Nature* 340:245-246.
- Fodor, S. P. A., Rava, R. P., Huang, X. C., Pease, A. C., Holmes, C. P., and Adams, C. L. (1993) Multiplexed biochemical assays with biological chips. *Nature* 364:555-556.
- Schena, M., Shalon, D., Davis, R. W., and Brown, P. O. (1995) Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science* 270(5235):467-470.
- Schena, M., Shalon, D., Heller, R., Chai, A., Brown, P. O., and Davis, R. W. (1996) Parallel human genome analysis: Microarray-based expression monitoring of 1000 genes. *Proc. Natl. Acad. Sci.* 93:10614-10619.
- Velculescu, V. E., Zhang, L., Vogelstein, B., and Kinzler, K. W. (1995) Serial analysis of gene expression. *Science* 270(5235):484-487.
- Lipshutz, R.J., Fodor, S.P., Gingeras, T.R., Lockhart, D.J. (1999) High density synthetic oligonucleotide arrays. *Nat Genet* (1 Suppl):20-4
- Lockhart, D.J., Dong, H., Byrne, M.C., Follettie, M.T., Gallo, M.V., Chee, M.S., Mittmann, M., Wang, C., Kobayashi, M., Horton, H., and Brown EL (1996) Expression monitoring by hybridization to high-density oligonucleotide arrays. *Nat Biotechnol* 13:1675-80

Genetic network modeling

- C.C. Hill, J.M. Tomasi, and J.P. Sethna, (2001) From Stochastic to Deterministic Descriptions of Gene Expression, www.ccmr.cornell.edu/~colin/
- Kauffman, S.A. (1971) Differentiation of malignant to benign cells. *J. Theoret. Biol.* 31:429-451

Asynchronous Boolean networks

- Thieffry, D. and Thomas, R. (1998) Qualitative Analysis of Gene Networks. *Pacific Symposium on Biocomputing* 3:66-76.
- Thomas, R. (1991) Regulatory Networks Seen as Asynchronous Automata: A Logical Description. *J Theor Biol.* 153: 1-23.
- Snoussi, E. H., and Thomas, R. (1993) Logical identification of all steady states: the concept of feedback loop characteristic states. *Bull. of Math. Biol.* 55:973-991.

Continuous logical networks

- Glass, L., and Kauffman, S. A. (1972) Co-operative components, spatial localization and oscillatory cellular dynamics. *J. Theor. Biol.* 34:219-237.
- Glass, L. and Kauffman, S.A. (1973) The Logical Analysis of Continuous, Non-Linear Biochemical Control Networks. *J. Theor. Biol.* 39:103-129.
- Glass, L. (1975). Classification of Biological Networks by Their Qualitative Dynamics. *J. Theor. Biol.* 54:85-107.

Stochastic behavior of networks:

- McAdams, H.H., Arkin, A. (1997) Stochastic Mechanisms in Gene Expression. PNAS, USA 94(3):814.
- Arkin, A., Ross, J., and McAdams, H. H. (1998) Stochastic kinetic analysis of developmental pathway bifurcation in phage lambda-infected Escherichia coli cells. Genetics 149(4):1633-48.
- McAdams H.H., Arkin A. (1998) Simulation of prokaryotic genetic circuits. Annu Rev Biophys Biomol Struct. 27:199-224.
- Abkowitz JL, Catlin SN, Gutter P. (1996) Evidence that hematopoiesis may be a stochastic process in vivo. Nat Med. 2:190-197

Biological constraints on genetic feedback networks:

- Barkal, N., and Leibler, S. (1997) Robustness in simple biochemical networks. Nature 387:913-917.
- Hlavacek, W. S., and Savageau, M. A. (1995) Subunit structure of regulator proteins influences the design of gene circuitry: analysis of perfectly coupled and completely uncoupled circuits. J. Mol. Biol. 248(4):739-755.
- Savageau, M. A. (1977) Design of molecular control mechanisms and the demand for gene expression. Proc. Natl. Acad. Sci. 74:5647-5651.
- Savageau, M.A. (1998) Rules for the Evolution of Gene Circuitry. Pacific Symposium on Biocomputing 3:54-65.
- Becskei A, Serrano L (2000) Engineering stability in gene networks by autoregulation Nature 405:590-593.

Reverse engineering

- Arkin, A., Shen, P., Ross, J. (1997) A Test Case of Correlation Metric Construction of a Reaction Pathway from Measurements. Science 277:1275-1279.
- Arkin, A., and Ross, J. (1997) Statistical Construction of Chemical Reaction Mechanisms from Measured Time-Series. J. Phys. Chem. 99:970-979.
- Akutsu T, Miyano S, Kuhara S. (2000) Algorithms for inferring qualitative models of biological networks. Pac Symp Biocomput. 5:293-304.
- Akutsu T, Miyano S, Kuhara S. (1999) Identification of genetic networks from a small number of gene expression patterns under the Boolean network model. Pac Symp Biocomput. 4:17-28.
- D'haeseleer, P., X. Wen, S. Fuhrman, and R. Somogyi (1999) Linear Modeling of mRNA Expression Levels During CNS Development and Injury. Pacific Symposium on Biocomputing.
- Liang S, Fuhrman S, Somogyi R (1998) REVEAL, A General Reverse Engineering Algorithm for Inference of Genetic Network Architectures. Pacific Symposium on Biocomputing 3:18-29.

Bottom-up modeling of small genetic networks

- McAdams, H.H., Shapiro, S. (1995) Circuit Simulation of Genetic Networks. Science. 269:650-656.

Genetic network modeling of spatio-temporal patterns in development (combined bottom up and top-down approach)

- Mjolsness, E., Sharp, D. H. and Reinitz, J. (1991) A connectionist model of development. J. Theor. Biol. 152: 429-453.
- Reinitz, J., Mjolsness, E., and Sharp, D.H. (1995) Model for cooperative control of positional information in Drosophila by bicoid and maternal hunchback. J. Exp. Zool. 271: 47-56.
- Reinitz, J. and Sharp, D.H. (1995) Mechanism of eve stripe formation. Mech. Dev.49: 133-158.

Cis-regulatory structures

- Arnone, M.I. and Davidson, E. (1997) The Hardwiring of Development: Organization and Function of Genomic Regulatory Systems. *Development* 124:1851-1864.

Linear or quasi-linear models

- Mjolsness, E., Sharp, D. H., and Reinitz, J. (1991) A connectionist model of development. *J. Theor. Biol.* 152(4):429-454.
- T. Chen, H. L. He, and G.M. Church (1999) Modeling Gene Expression with Differential Equations. Pacific Symposium on Biocomputing.
- D.C. Weaver, C.T. Workman, G.D. Stormo (1999) Modeling Regulatory Networks with Weight Matrices. Pacific Symposium on Biocomputing.

General systems theory

- Bonnländer, B. V., and Weigend, A. S. (1994) Selecting input variables using mutual information and nonparametric density estimation. In Proceedings of the 1994 International Symposium on Artificial Neural Networks, 42-50.
- Cavallo, R. E., and Klir, G. J. (1981) Reconstructability analysis: overview and bibliography. *Int. J. Gen. Sys.* 7:1-6.

Neural networks

- Bray, D. (1990) Intracellular signaling as a parallel distributed process. *J. Theor. Biol.* 143:215-231.
- Mjolsness, E., Sharp, D. H., and Reinitz, J. (1991) A connectionist model of development. *J. Theor. Biol.* 152(4):429-454.
- Heckerman, D. A (1995) tutorial on learning with bayesian networks. Tech. Rep. MSR-TR-95-06, Microsoft Research, Redmond, WA. Available via ftp from ftp.research.microsoft.com in /pub/Tech-Reports/Winter94- 95/TR-95-06.PS.
- Hofmann, R., and Tresp, V. (1996) Discovering structure in continuous variables using bayesian networks. In *Advances in Neural Information Processing Systems 8*, D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, Eds., MIT Press, 500-506.

Cluster analysis of gene expression matrices

- Eisen, M.B., Spellman, P.T., Brown, P.O., and Bottstein, D. (1998) Cluster analysis and display of genome-wide expression patterns. *Proc Natl Acad Sci USA*, 95:14863-14868.
- K. Y. Yeung, D. R. Haynor, and W. L. Ruzzo (2001) Validating clustering for gene expression data. *Bioinformatics* 17: 309-318.
- Blatt, M., Wisemann, S, and Domany, E. (1996) Super-paramagnetic clustering of data. *Phys. Rev. Lett.* 76:3251-3254.
- Sásik, R, T. Hwa, N. Iranfar, and W.F. Loomis Percolation Clustering: A Novel Algorithm Applied to the Clustering of Gene Expression Patterns in Dictyostelium Development Pacific Symposium on Biocomputing 6:335-347 (2001).
- Heyer LJ, Kruglyak S, Yooseph S. (1999) Exploring expression data: identification and analysis of coexpressed genes. *Genome Res.* 11:1106-1115
- Hastie, T., Tibshirani, R. , Eisen, M.B., Alizadeh, A., Levy, R., Staudt, L., Chan, W.C., Botstein, D., Brown, P. (2000) Gene shaving' as a method for identifying distinct sets of genes with similar expression patterns *Genome Biology* 2000 1: research0003.1-0003.21
- Cohen BA, Mitra RD, Hughes JD, Church GM (2000) A computational analysis of whole-genome expression data reveals chromosomal domains of gene expression. *Nat Genet.* (2):183-186.

- Wen X., Fuhrman S., Michaels G.S., Carr D.B., Smith S., Barker J.L., Somogyi R. (1998) Large-Scale Temporal Gene Expression Mapping of CNS Development. *Proc Natl Acad Sci* 95:334-339. (<http://rsb.info.nih.gov/mol-physiol/PNAS/GEMtable.html>)
- Brown, M., Grundy, W., Lin, D., Cristianini, N., Sugnet, C., Furey, T., Ares, M., and D. Haussler, D. (1999) Knowledge-based analysis of microarray gene expression data using support vector machines. *Proceedings of the National Academy of Science*, 97: 262-267 (1999).
- Shamir, R and Sharan, R. (2000) CLICK: A Clustering Algorithm for Gene Expression Analysis *Proc. 8th International Conference on Intelligent Systems for Molecular Biology (ISMB '00)*, pp., 260--268, AAAI Press, Menlo Park, CA (2000)
- Shamir, R and Sharan, R. (2001) Algorithmic Approaches to Clustering Gene Expression Data Submitted to *Current Topics in Computational Biology* T.Jiang, T. Smith, Y. Xu, M. Q. Zhang (editors) MIT Press
- Michaels G, Carr DB, Wen X, Fuhrman S, Askenazi M, Somogyi R (1998) Cluster Analysis and Data Visualization of Large-Scale Gene Expression Data. *Pacific Symposium on Biocomputing* 3:42-53.
- Tavazoie, S., Hughes, J.D., Campbell, M.J., Cho, R.J, and Church, G.M. (1999) Systematic determination of genetic network architecture. *Nature Genetics*, 22:281-285.
- Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrowsky, E., Lander, E.S., and Golub, T.R. (1999) Interpreting patterns of gene expression with self-organizing maps: Methods and applications to hematopoietic differentiation. *Proc Natl Acad Sci USA*, 96:2907-2912.
- Wen X, Fuhrman S, Michaels GS, Carr DB, Smith S, Barker JL, Somogyi R (1998) Large-Scale Temporal Gene Expression Mapping of CNS Development. *Proc Natl Acad Sci USA*, 95:334-339.
- A huge list of papers about self-organizing maps is available by S. Kaski, J. Kangas and T. Kohonen at the following website: <http://www.icsi.berkeley.edu/~jagota/NCS>
- Cheng Y, Church GM (2000) Biclustering of expression data. *Proc Int Conf Intell Syst Mol Biol* 8:93-103
- Weinstein, J.N. et al. (1997) An information-intensive approach to the molecular pharmacology of cancer. *Science* 275:343-349
- Hartigan, J.A. (1972) Direct clustering of a data matrix. *JASA* 67:123-129
- Alon, U., Barkai, N., Notterman, D.A., Gish, K., Ybarra, S., Mack, D., Levine, A.J., (1999) Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. *Proc Natl Acad Sci USA*, 96:6745-6750.
- Mjolsness, E., Castano, R., Mann, T., Roden, J., Gray, A., and Wold, B. (1999) Clustering methods for the analysis of *C. elegans* gene expression array data. Technical report JPL-ICTR-99-1, available online at: www.aig.jpl.nasa.gov/mls/mls_papers.html

Recommended books:

Mirkin, B. (1996) *Mathematical Classification and Clustering*. Dordrecht: Kluwer

Bishop, C.M (1995) *Neural Networks for Pattern Recognition*, Clarendon Press, Oxford