BIOINFORMATICS

LEI LIU
ALI ABBAS
University of Illinois at Urbana-Champaign
Urbana, Illinois

1. INTRODUCTION

In the past two decades we have witnessed revolutionary changes in biomedical research and biotechnology and an explosive growth of biomedical data. High throughput technologies developed in automated DNA sequencing, functional genomics, proteomics, and metabolomics enable us to produce such high volume and complex data that the data analysis becomes a big challenge. Consequently, a promising new field, Bioinformatics has emerged and is growing rapidly. Combining biological studies with Computer Science, Mathematics and Statistics, Bioinformatics develops methods, solutions, and software to discover patterns, generate models, and gain insight knowledge of complex biological systems.

Before we discuss further of the field, let us briefly review the basic concepts in molecular biology, which are the foundations for bioinformatics studies. The genetic information is coded in DNA sequences. The physical form of a gene is a fragment of DNA. A genome is the complete set of DNA sequences that encode all the genetic information for an organism, which is often organized into one or more chromosomes. The genetic information is decoded through complex molecular machinery inside a cell composed of two major parts: transcription and translation, to produce functional protein and RNA products. These molecular genetic processes can be summarized precisely by the Central Dogma shown in Fig. 1. The proteins and active RNA molecules combined with other large and small biochemical molecules, organic and inorganic compounds form the complex dynamic network systems that maintain the living status of a cell. Proteins form complex 3-D structures that carry out functions. The 3-D structure of a protein is determined by the primary protein sequence and the local environment. The protein sequence is decoded from the DNA sequence of a gene through the genetic codes as shown in Table 1. These codes have been shown to be universal among all living forms on earth.

The high throughput data can be generated at many different levels in the biological system. The genomics data are generated from the genome sequencing that deciphers the complete DNA sequences of all the genetic information in an organism. We can measure the mRNA levels using microarray technology to monitor the gene expression of all the genes in a genome known as transcriptome. Proteome is the complete set of proteins in a cell at a certain stage, which can be measured by high throughput 2-D gel electrophoresis and Mass Spectrometry. We also can monitor all the metabolic compounds in a cell known as metabolome in a high throughput fashion. Many new terms ending with "ome" can be viewed as the complete set of entities in a cell. For example, the "interactome" refers to the complete set of protein-protein interactions in a cell.

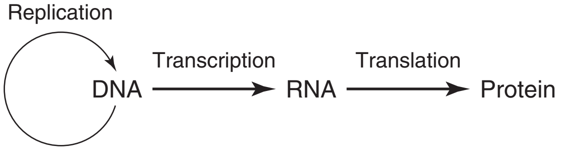


Figure 1. Central dogma of molecular biology.

The theory of evolution is also a fundamental base for many aspects of Bioinformatics, especially on sequence and phylogenetic analyses. According to the Darwin's theory of evolution, mutation and natural selection is the driving force during the evolution. In 1980s, the neutral theory of molecular evolution was proposed by Kimura (1) based on the observation that the mutation rates were not even on different parts of genomes. The places that change rapidly might not be under the natural selection. The assumptions on how genes are changing have profound influence on the analysis of biological sequences.

Bioinformatics is needed at all levels of high throughput systematic studies to facilitate the data analysis, mining, management, and visualization. But more importantly, the major task is to integrate data from different levels and prior biological knowledge to achieve system level understanding of biological phenomena. Since Bioinformatics touches on many areas of biological studies, it is impossible to cover every aspect in a short chapter. In this article, the authors will provide a general overview of the field and focus on several key areas including: sequence analysis, phylogenetic analysis, protein structure, genome analysis, microarray analysis, and network analysis.

Sequence analysis often refers to sequence alignment and pattern searching in both DNA and protein sequences. This area can be considered as the "classical" Bioinformatics, which can be dated back to 1960s, long before the word "Bioinformatics" appeared. It deals with the problems such as how to make an optimal alignment between two sequences; how to search sequence databases quickly with an unknown sequence. Phylogenetic analysis is closely related to sequence alignment. The idea is to use DNA or protein sequences comparison to infer evolution history. The first step in this analysis is to perform multiple sequence alignment. Then a phylogenetic tree is built based on the multiple alignments. The protein structure analysis involves the prediction of protein secondary and tertiary structures from the primary sequences. So far the analyses focus on individual sequences or handful of sequences. The next three areas are involved in system wide analysis. Genome analysis mainly deals with the sequencing of a complete or partial genome. The problems include genome assembly, gene structure prediction, gene function annotation, and so on. Many techniques of sequence analysis are used in genome analysis, but many new methods were developed for the unique problems. Microarray technologies provide an opportunity for biologist to study the gene expression at a system level. The problems faced in the analysis are completely different from se-

Table 1. The Genetic Code

		Second	Position		
First Position	T	C	A	G	Third Position
T	TTT Phe [F]	TCT Ser [S]	TAT Tyr [Y]	TGT Cys [C]	T
	TTC Phe [F]	TCA Ser [S]	TAC Tyr [Y]	TGC Cys [C]	C
	TTA Leu [L]	TCG Ser [S]	TAA *Stop* [end]	TGA *Stop* [end]	A
	TTG Leu [L]	TCC Ser [S]	TAG *Stop* [end]	TGG Trp [W]	G
C	CTT Leu [L]	CCT Pro [P]	CAT His [H]	CGT Arg [R]	T
	CTC Leu [L]	CCC Pro [P]	CAC His [H]	CGC Arg [R]	C
	CTA Leu [L]	CCA Pro [P]	CAA Gln [Q]	CGA Arg [R]	A
	CTG Leu [L]	CCG Pro [P]	CAG Gln [Q]	CGG Arg [R]	G
A	ATT Ile [I]	ACT Thr [T]	AAT Asn [N]	AGT Ser [S]	T
	ATC Ile[I]	ACC Thr [T]	AAC Asn [N]	AGC Ser [S]	C
	ATA Ile [I]	ACA Thr [T]	AAA Lys [K]	AGA Arg [R]	A
	ATG Met [M]	ACG Thr [T]	AAG Lys [K]	AGG Arg [R]	G
G	GTT Val [V]	GCT Ala [A]	GAT Asp [D]	GGT Gly [G]	T
	GTC Val [V]	GCC Ala [A]	GAC Asp [D]	GGC Gly [G]	C
	GTA Val [V]	GCA Ala [A]	GAA Glu [E]	GGA Gly [G]	A
	GTG Val [V]	GCG Ala [A]	GAG Glu [E]	GGG Gly [G]	G

quence analysis. Many statistical and data mining techniques are applied in the field. Network analysis is another system level study of biological system. Biological networks can be divided into three categories: metabolic network, protein-protein interaction network, and genetic network. The questions in this area include network modeling, network inference from high throughput data, such as microarray, and network properties study. In the following several sections, we will make a more in-depth discussion of each area.

2. SEQUENCE ALIGNMENT

2.1. Pair-wise Sequence Alignment

Sequence alignment can be described by the following problem. Given two strings of text, *X* and *Y*, (that may be DNA or amino acid sequences) find the optimal way of inserting dashes into the two sequences so as to maximize a given scoring function between them. The scoring function depends on both the length of the regions of consecutive dashes and on the pairs of characters that are in the same position when gaps have been inserted. The following example from Abbas and Holmes (2) illustrates the idea of sequence alignment for two strings of text. Consider the two sequences, COUNTING, and NTIG shown in Fig. 2a. Figures 2b–d, show possible alignments obtained by inserting gaps (dashes) at different positions in one of the sequences. Figure 2d shows the alignment with the highest number of matching elements. The "optimal alignment" between two sequences depends on the scoring function that is used. As we shall see, an optimal sequence alignment for a given scoring function may not be unique.

Now we have discussed what is meant by an optimal sequence alignment, we need to explain the motivation for doing it. Sequence alignment algorithms can detect mutations in the genome that lead to genetic disease, and also provide a similarity score, which can be used to determine the probability that the sequences are evolutionarily related. Knowledge of evolutionary relation between a newly identified protein sequence and a family of protein sequences in a database may provide the first clues about its three-dimensional structure and chemical function. Furthermore, by aligning families of proteins that have the same function (and may have very different sequences) we can observe a common subsequence of amino acids that is key to its particular function. These subsequences are termed protein motifs. Sequence alignment is also a first step in constructing phylogenetic trees that relate biological families of species.

A dynamic programming approach to sequence alignment was proposed by Needleman and Wunsch (3). The idea behind the dynamic programming approach can be explained using the two sequences, CCGAT and CA-AT, of Fig. 3a. Suppose we have an optimal alignment for the two sequences and an additive scoring system for their alignment. If we break this alignment into two parts (Fig. 3b), we have two alignments: the left is the alignment of the two sequences CCGA and CA-A, and the right is the alignment of the last elements T-T. If the scoring system is additive, then the score of the alignment of Fig. 3b is the sum of the scores of the four base-alignment on the left (CCGA and CA-A) plus the score of the alignment of the pair T-T on the right. If the alignment in Fig. 3a is optimal then the four-base alignment in the left hand side of Fig. 3b must also be optimal. If this were not the case (for example if a better alignment would be obtained by aligning A with G) then the optimal alignment of Fig. 3c would lead to a higher score than the alignment shown in Fig. 3a. The optimal alignment ending at any stage is therefore equal to the total (cumulative) score of the optimal alignment at the previous stage plus the score assigned to the aligned elements at that current stage.

```
(a)   Sequence 1   C  O  U  N  T  I  N  G
      Sequence 2   N  T  I  G

(b)   Sequence 1   C  O  U  N  T  I  N  G   Possible Alignment
      Sequence 2   -  -  -  N  T  I  G  -   (Shifting Sequence 2)

(c)   Sequence 1   C  O  U  N  T  I  N  G   Possible Alignment
      Sequence 2   -  -  -  N  T  -  I  G   (Shifting Sequence 2 and inserting a gap)

(d)   Sequence 1   C  O  U  N  T  I  N  G   Possible Alignment
      Sequence 2   -  -  -  N  T  I  -  G   (Shifting Sequence 2 and inserting a gap)
```

Figure 2. Possible alignments of two sequences.

```
C  C  G  A  T        C  C  G  A     T        C  C  G  A  T
|        |  |        |        |  +  |        |        |  |
C  A  –  A  T        C  A  –  A     T        C  –  A  A  T
     (a)                  (b)                     (c)
```

Figure 3. Overview of the dynamic programming approach.

The optimal alignment of two sequences ends with the last two symbols aligned, the last symbol of one sequence aligned to a gap, or the last symbol of the other sequence aligned to a gap. In our analysis x_i refers to the i^{th} symbol in sequence 1 and y_j refers to the j^{th} symbol in sequence 2 before any alignment has been made. We will use the symbol $S(i, j)$ to refer to the cumulative score of the alignment up until symbols x_i and y_j, and the symbol $s(x_i, y_j)$ to refer to the score assigned to matching elements x_i and y_j. We will use d to refer to the cost associated with introducing a gap.

1. If the current stage of the alignment matches two symbols, x_i and y_j, then the score, $S(i, j)$, is equal to the previous score, $S(i-1, j-1)$, plus the score assigned to aligning the two symbols, $s(x_i, y_j)$.
2. If the current match is between symbol x_i in sequence 1 and a gap in sequence 2 then the new score is equal to the score up until symbol x_{i-1} and the same symbol y_j, $S(i-1, j)$, plus the penalty associated with introducing a gap, $-d$
3. If the current match is between symbol y_j in sequence 2 and a gap in sequence 1 then the new score is equal to the previous score up until symbol y_{j-1} and the same symbol x_i, $S(i, j-1)$, plus the gap penalty $-d$

The optimal cumulative score at symbols x_i and y_j is:

$$S(i,j) = \max \begin{cases} S(i-1, j-1) + s(x_i, y_j) \\ S(i-1, j) - d \\ S(i, j-1) - d. \end{cases}$$

The previous equation determines the new elements at each stage in the alignment by successive iterations from the previous stages. The maximum at any stage may not be unique. The optimal sequence alignment (s) is the one that provides the highest score. This is usually performed using a matrix representation, where the cells in the matrix are assigned an optimal score, and the optimal alignment is determined by a process called trace back (4,5).

The optimal alignment between two sequences depends on the scoring function that is used. This brings the need for a score that is biologically significant and relevant to the phenomenon being analyzed. Substitution matrices, present one method of achieving this using a "log-odds" scoring system. It lists the likelihood of change from one amino acid or nucleotide to another in homologous sequences during evolution. One of the first substitution matrices used to score amino acid sequences was developed by Dayhoff et al (6) and called Percent Accepted Mutation (PAM) Matrix, which was derived from a relatively small set of closely related proteins. Other matrices such as the BLOSUM50 matrix (7) were also developed and use databases of more distantly related proteins.

The Needleman- Wunsch (N-W) algorithm and its variation (4) provide the best *global* alignment for two given sequences. Smith and Waterman (8) presented another dynamic programming algorithm that deals with finding the best *local* alignment for smaller subsequences of two given sequences rather than the best global alignment of the two sequences. The local alignment algorithm identifies a pair of subsegments, one from each of the given sequences, such that there is no other pair of subsegments with greater similarity.

2.2. Heuristic Alignment Methods

Heuristic search methods for sequence alignment have gained popularity and extensive use in practice because of the complexity and large number of calculations in the dynamic programming approach. Heuristic approaches search for local alignments of subsegments and use these alignments as "seeds" in which to extend out to longer sequences. The most widely used heuristic search method available today is BLAST (Basic Local Alignment Search Tool) by Altschul et al (9). BLAST alignments define a measure of similarity called MSP (Maximal Segment Pair) as the highest scoring pair of identical length subsegments from two sequences. The lengths of the subsegments are chosen to maximize the MSP score.

2.3. Multiple Sequence Alignments

Multiple sequence alignments are alignments of more than two sequences. The inclusion of additional sequences can improve the accuracy of the alignment, find protein motifs, identify related protein sequences in a database, and predict protein secondary structure. Multiple sequence alignments are also the first step in constructing phylogenetic trees.

The most common approach for multiple alignments is progressive alignment, which involves choosing two sequences and performing a pairwise alignment of the first to the second. The third sequence is then aligned to the first and the process is repeated until all the sequences are aligned. The score of the multiple alignment is the sum of scores of the pairwise alignments. Pairwise dynamic programming can be generalized to perform multiple alignments using the progressive alignment approach; however, it is computationally impractical even when only a few sequences are involved (10). The sensitivity of progressive alignment was improved for divergent protein sequences using CLUSTAL-W (11), available at (*http://clustalw.genome.ad.jp/*).

Many other approaches to sequence alignment have been proposed in the literature. For example, a Bayesian approach was suggested for adaptive sequence alignments (12,13), (Zhu et al 1998). Another approach to sequence alignment that has found great success is hidden Markov models. We will refer to this approach in more detail in the section on genome annotation. The data that is now available from the human genome project has suggested the need for aligning whole genome sequences where large-scale changes can be studied as opposed to single-gene insertions, deletions, and nucleotide substitutions. MuMMer (14) follows this direction and performs alignments and comparisons of very large sequences.

3. PHYLOGENETIC TREES

Biologists have long built trees to classify species based on morphological data. The main objectives of phylogenetic tree studies are (1) to reconstruct the genealogical ties between organisms and (2) to estimate the time of divergence between organisms since they last shared a common ancestor. With the explosion of genetic data in the last few years, molecular based phylogenetic studies have been used in many applications such as the study of gene evolution, population subdivisions, analysis of mating systems, paternity testing, environmental surveillance, and the origins of diseases that have transferred species.

From a mathematical point of view, a phylogenetic tree is a rooted binary tree with labeled leaves. A tree is binary if each vertex has either one or three neighbors. A tree is rooted if a node, R, has been selected and termed the root. A root represents an ancestral sequence from which all other nodes descend. Two important aspects of a phylogenetic tree are its topology and branch length. The topology refers to the branching pattern of the tree and the branch length is the "evolutionary" time between the splitting events. Figure 4a shows a rooted binary tree with six leaves. Figure 4b shows all possible distinct rooted topologies for a tree with 3 leaves.

The data that is used to construct trees is usually in the form of contemporary sequences and is located at the leaves. For this reason trees are represented with all their leaves "on the ground level" rather than at different levels.

The tree-building analysis consists of two main steps. The first step, estimation, uses the data matrix to produce a tree, $\tilde{T}$, that estimates the unknown tree, T. The second step provides a confidence statement about the estimator $\tilde{T}$. This is often performed by bootstrapping methods.

Tree-building techniques can generally be classified into one of four types: distance-based methods, parsimony methods, maximum likelihood methods, and Bayesian methods. For a detailed discussion of each of these methods see Li (15). Now we will give a brief overview of each of the tree-building methods, more details of which can be found in Abbas and Holmes (2).

Distance-Based methods first calculate an "evolutionary distance", d_{xy}, between each two sequences X and Y in a multiple sequence alignment. The pairwise distance for N sequences results in a distance matrix of dimension $N \times N$, which is symmetric about its diagonal if the distance d_{xy} is symmetric. One of the widely used distance-based methods of phylogenetic tree construction the Jukes-Cantor model (16) that provides an estimate of the evolutionary distance between X and Y as

$$d_{xy} = -\frac{3}{4}\log\left(1-\frac{4}{3}\left(1-\left(\frac{\#AA}{K}+\frac{\#CC}{K}+\frac{\#GG}{K}+\frac{\#TT}{K}\right)\right)\right),$$

where K denotes the number of characters (columns) in the dataset, and $\#AA$ denotes the number of times a letter A in sequence X is matched with a letter A in sequence Y. Once the distance matrix is calculated, the phylogenetic tree is estimated using a clustering technique. Clustering

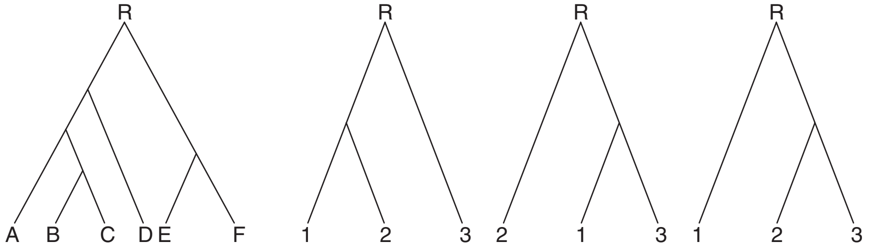


Figure 4. (a) Rooted tree with six leaves. (b) All possible topologies for three leaves.

is the task of segmenting the sequences into a number of homogeneous subgroups or clusters. The most commonly applied clustering methods are the unweighted pair group method with arithmetic mean (UPGMA), and the neighbor-joining method (17).

The Parsimony Method for constructing phylogenetic trees is based on the assumption that “evolution is parsimonious” which means that there should be no more evolutionary steps than necessary. As a result, the phylogenetic tree that is selected is the one with the minimum number of substitutions between ancestor and descendants. Maximum likelihood methods select the tree that has the highest probability of producing the observed data. Under this model the likelihood for each possible tree is separately computed for each sequence (row) in the data set. This requires computing the likelihood of all the possible trees and so the method is computationally expensive and requires efficient search procedures. For more details see Felsenstein (18). The Bayesian estimation methods start the tree construction with a very wide prior distribution on the space of all trees. The approach then uses Gibbs sampling and Monte Carlo Markov Chains to compute a posterior probability distribution on the tree conditioned on the dataset. To facilitate this task, Huelsenbeck and Ronquist (19) developed a software package called Mr. Bayes to perform Bayesian inference of phylogenetic trees using use MCMC simulation.

Tree-building methods can be compared using several criteria such as accuracy, consistency, efficiency, and robustness. To clarify some of these issues, we refer the reader to Holmes (20) where a geometric analysis of the problem is provided and these issues are further discussed. The second part of the tree-building analysis is concerned with how “close” we believe the estimated tree is to the true tree. Felsenstein (21) suggested the use of the bootstrap to answer this question of how much confidence should we have in the estimated trees. Another method builds on a probability distribution on the space of all trees. The difficult part of this problem is that there are exponentially many possible trees. A nonparametric approach using a multinomial probability model on the whole set of trees would not be feasible as the number of trees is (2N-3)!!. The Bayesian approach defines parametric priors on the space of trees, and then computes the posterior distribution on the same subset of the set of all trees. This analysis enables confidence statements in a Bayesian sense (22).

4. PROTEIN FOLDING, SIMULATION, AND STRUCTURE PREDICTION

The structure of a protein greatly influences its function. Knowledge of protein structure and function can help determine the chemical structure of drugs needed to reverse the symptoms that arise due to its malfunction. The bonds in a molecular structure contribute to its overall potential energy. We shall neglect all quantum mechanical effects in the following discussion and consider only the elements that contribute largely to the potential energy of a structure (as suggested by (23)).

1. **Pair Bonds:** This is a bond that exists between atoms physically connected by a bond and separated by a distance b. It is like a spring action where energy is stored above and below an equilibrium distance, b_0. The energy associated with this bond is $U(b) = \frac{1}{2}K_b(b - b_o)^2$, where b_0 can be determined from the X-ray of the crystal structure showing the electron density maps, and K_b can be determined from spectroscopy.
2. **Bond Angles**: This bond exists when an angular deviation from an equilibrium angle, θ_o, occurs between three atoms. The bond angle energy associated with the triplet is $U(\theta) = \frac{1}{2}K_\theta(\theta - \theta_0)^2$.
3. **Torsion Angles:** This bond exists when a torsion angle, ϕ, exists between the first and fourth atoms on the axis of the second and third atoms. The energy associated with this bond is $U(\phi) = K_\phi(1 - \cos(n\phi + \delta))$, where δ is an initial torsion angle.
4. **Non-bonded pairs:** Bonds also exist between atoms that are not physically connected in the structure. These bonds include
   a. Van der Waal forces, which exist between non-bonded pairs and contribute to energy, $U(r) = \varepsilon[(\frac{r_0}{r})^{12} - 2(\frac{r_0}{r})^6]$, r_0 is an equilibrium distance and ε a constant.
   b. Electrostatic Interactions, which contribute to an energy of $U(r) = \alpha\frac{q_i q_j}{r}$, and
   c. Hydrogen bonds, which result from Van Der Waals forces and the geometry of the system, and contribute to the potential energy of the structure.

The total potential energy function of a given structure can thus be determined by the knowledge of the precise position of each atom. The three main techniques that are used for protein structure prediction: homology (comparative modeling), fold recognition and threading, and *Ab initio* folding.

4.1. Homology or Comparative Modeling

Comparative modeling techniques predict the structure of a given protein sequence based on its alignment to one or more protein sequences of known structure in a protein database. The approach uses sequence alignment techniques to establish a correspondence between the known structure “template” and the unknown structure. Protein structures are archived for public use in an Internet-accessible database known as the Protein Data Bank. *http://www.rcsb.org/pdb/* (24).

4.2. Fold Recognition and Threading

When the two sequences exhibit less similarity, the process of recognizing which folding template to use is more difficult. The first step in this case is to choose a structure from a library of templates in the protein databank. This is called fold recognition. The second step “threads” the given protein sequence into the chosen template. Several computer software programs are available for protein structure prediction using the fold recognition and threading technique such as PROSPECT (25).

4.3. Ab Initio (New Fold) Prediction

If no similarities exist with any of the sequences in the database, the ab initio prediction method is used. This method is one of the earliest structure prediction methods, and uses energy interaction principles to predict the protein structure (23,26,27). Some of these methods include optimization where the objective is to find a minimum-energy structure (a local minimum in the energy landscape has zero forces acting on the atoms and is therefore an equilibrium state).

Another type of analysis uses Molecular dynamics uses equations of motion to trace the position of each atom during folding of the protein (28). A single structure is used as a starting point for these calculations. The force acting on each atom is the negative of the gradient of the potential energy at that position. Accelerations, a_i, are related through masses, m_i, to forces, F_i, via Netwon's second law ($F_i = m_i a_i$). At each time step, new positions and velocities of each of the atoms are determined by solving equations of motion using the old positions, old velocities, and old accelerations. Beeman (29) showed that new atomic positions and velocities could be determined by the following equations of motion

$$x(t+\Delta t) = x(t) + v(t)\Delta t + [4a(t) - a(t+\Delta t)]\frac{(\Delta t)^2}{6},$$

$$v(t+\Delta t) = v(t) + [2a(t+\Delta t) + 5a(t) - a(t-\Delta t)]\frac{\Delta t}{6},$$

where $x(t)$ = position of the atom at time, t, $v(t)$ = velocity of the atom at time, t, $a(t)$ = acceleration at time, t, and Δt = time step in the order of 10^{-15} seconds for the simulation to be accurate.

In 1994, the first large-scale experiment to assess protein structure prediction methods was conducted. This experiment is known as CASP (Critical Assessment of techniques for protein Structure Prediction). The results of this experiment were published in a special issue of *Proteins* (1995). Further experiments were developed to evaluate the fully automatic web servers for fold recognition. These experiments are known as CAFASP (Critical Assessment of Fully Automated Structure Prediction). For a discussion on the limitations, challenges, and likely future developments on the evaluation of the field of protein folding and structure prediction, we refer the reader to (30).

5. GENOME ANALYSIS

Analysis of completely sequenced genomes has been one of the major driving forces for the development of bioinformatics field. The major challenges in this area include genome assembly, gene prediction, function annotation, promoter region prediction, identification of single nucleotide polymorphism (SNP), and comparative genomics of conserved regions. For a genome project, one must ask several fundamental questions: how can we put the whole genome together from many small pieces of sequences? where are the genes located on a chromosome? and what are other features we can extract from the completed genomes?

5.1. Genome Assembly

The first problem is pertaining to the genome mapping and sequence assembly. During the sequencing process, large DNA molecules with millions of base pairs, such as a human chromosome, are broken into smaller fragments (~100 kb) and cloned into vector such as bacterial artificial chromosome (BAC). These BAC clones can be tiled together by physical mapping techniques. Individual BACs can be further broken down into smaller random fragments of 1-2 kb. These fragments are sequenced and assembled based on overlapping fragments. With more fragments sequenced, there will be enough overlaps to cover most of the sequence. This method is often referred as "shotgun sequencing". Computer tools were developed to assemble the small random fragments into large contigs based on the overlapping ends among the fragments using similar algorithms as the ones used in the basic sequence alignment. The widely used ones include PHRAP/Consed (31,32) and CAP3 (33). Most of prokaryotic genomes can be sequenced directly by the "shotgun sequencing" strategy with special techniques for gap closure. For large genomes, such as human genome, there are two strategies. One is to assemble large contigs first and then tile together the contigs based on the physical map to form the complete chromosome (34) Another strategy is called Whole Genome Shotgun Sequencing (WGS) strategy, which assemble the genome directly from the "shotgun sequencing" data in combination with mapping information (35). WGS is a faster strategy to finish a large genome, but the challenge of WGS is how to deal with the large number of repetitive sequences in a genome. Nevertheless, WGS has been successfully used in completing the *Drosophila* and human genomes (36,37).

5.2. Genome Annotation

The second problem is related to deciphering the information coded in a genome, which is often called genome annotation. The process includes the prediction of gene structures and other features on a chromosome and the function annotation of the genes. There are two basic types of genes in a genome: RNA genes and protein encoding genes. RNA genes produce active RNA molecules such as ribosomal RNA, tRNA, small RNA. Majority of genes in a genome are protein encoding genes. Therefore, the big challenge is how to find the protein encoding region in a genome. The simplest way to search for a protein encoding region is to search for open reading frames (ORF), which is a contiguous set of codons between two stop codons. There are six possible reading frames for a given DNA sequence. Three of them start at the first, second, and third base. The other three reading frames are at the complementary strand. The longest ORFs between the start codon and the stop codon in the same reading frame provide good, but not sufficient evidence of a protein encoding region. Gene prediction is generally easier and more accurate in prokaryotic than eukaryotic organisms due to the intron/exon structure in eukaryote genes. In prokaryotic organisms (bacteria and archaea), the translation and transcription are highly coupled. The protein is synthesized from the RNA even before the transcription of the gene is finished. The coding region spans the whole RNA without interruption. On the other hand, in eukaryotic organisms, transcription is more complicated. After the initial RNA product is generated from the gene, it goes through a process called splicing to get rid of non-coding regions (intron) and concatenate the coding regions (exon) together. Computational methods of gene prediction based on Hidden Markov Model (HMM) have been quite successful, especially in prokaryote genome. These methods involve training a gene model to recognize genes in a particular organism. Because of the variations in codon usage, a model must be trained for each new genome. In prokaryote genome, genes are packed densely with relatively short intergenic sequences. The model reads through a sequence with unknown gene composition and find the regions flanked by start and stop codons. The codon composition of a gene is different from that of an intergenic region and can be used as a discriminator for gene prediction. Several software tools, such as GeneMark (38) and Glimmer (14) are widely used HMM methods in prokaryotic genome annotation. Similar ideas are also applied to eukaryote gene prediction. Because of the intron/exon structure, the model is much more complex with more attention on the boundary of intron and exon. Programs such as GeneScan (39) and GenomeScan (40) are HMM methods for eukaryote gene prediction. Neural network based methods have also been applied in eukaryote gene prediction, such as Grial (41). Additional information for gene prediction can be found using expressed sequence tags (ESTs), which are the sequences from cDNA libraries. Because cDNA is derived from mRNA, a match to an EST is a good indication that the genomic region encodes a gene. Functional annotation of the predicted genes is another major task in genome annotation. This process can be also viewed as gene classification with different functional classification systems such as Enzyme Commission Numbers (EC number) system, protein families, metabolic pathways, and Gene Ontology. The simplest way is to infer annotation from the sequence similarity to a known gene, e.g. BLAST search against a well-annotated protein database such as SWISS-PROT. A better way can be a search against protein family databases (e.g. Pfam (42)), which are built based on profile HMMs. The widely used HMM alignment tools include HMMER (43) and SAM (44). All automated annotation methods can produce mistakes. More accurate and precise annotation requires experimental verification and combination of information from different sources.

Besides the gene structures, other features such as promoters can be better analyzed with a finished genome. In prokaryotic organisms, genes involved in the same pathway are often organized in an operon structure, in which the genes contiguous on the chromosome and transcribed together. Finding operons in a finished genome provides information on the gene regulation. For eukaryotic organisms, the completed genomes provide upstream sequences for promoter region search and prediction. Promoter region prediction and detection has been a very challenging bioinformatics problem. The promoter regions are the binding sites for transcription factors (TF). Promoter prediction is to discover the sequence patterns which are specific for TF binding. Different motif finding algorithms have been applied including scoring matrix method (45), Gibbs sampling (46), and Multiple EM for Motif Elicitation (MEME) (47). The results are not quite satisfactory. Recent studies using comparative genomics methods on the problem have produced some promising results and demonstrated that the promoters are conserved among closely related species (48). In addition, microarray studies can provide additional information for promoter discoveries (see, the section 6).

5.3. Comparative Genomics

With more and more genomes being completely sequenced, comparative analysis becomes increasingly valuable and provides more insights of genome organization and evolution. One comparative analysis is based on the orthologous genes, called clusters of orthologous groups (COG) (49). Two genes from two different organisms are considered orthologous genes if they are believed to come from a common ancestor gene. Another term, paralogous genes, refers to genes in one organism and related to each other by gene duplication events. In COG, proteins from all completed genomes are compared. All matching proteins in all the organisms are identified and grouped into orthologous groups by speciation and gene duplication events. Related orthologous groups are then clustered to form a COG that includes both orthologs and paralogs. These clusters correspond to classes of functions. Another type of comparative analysis is based on the alignment of the genomes and studies the gene orders and chromosomal rearrangements. A set of orthologous genes that show the same gene order along the chromosomes in two closely related species is called a synteny group. The corresponding region of the chromosomes is called synteny blocks (50). In closely related species, such as mammalian species, the gene orders in synteny regions are generally conserved with many rearrangements. The chromosomal rearrangements include inversion, translocation, fusion and fission. By comparing completely sequenced genomes, for example, human and mouse genomes, we can reveal the rearrangement events. One challenging problem is to reconstruct the ancestral genome from the multiple genome comparisons and estimate the number and types of the rearrangements (51). Detailed comparisons of genomes often reveal not only the differences between species but much more insight of evolution and function and regulation of genes. Recent publication of chimpanzee genome and its comparison to human genome shows approximately thirty-five million single-nucleotide changes, five million insertion/deletion events, and various chromosomal rearrangements (52).

6. MICROARRAY ANALYSIS

Microarray technologies allow biologists to monitor genome-wide patterns of gene expression in a high throughput fashion. Gene expression refers to the process of transcription. Gene expression for a particular gene can be measured as the fluctuation of the amount of messenger RNA produced from the transcription process of that gene in different conditions or samples.

DNA microarrays are typically composed of thousands of DNA sequences, called probes, fixed to a glass or silicon substrate. The DNA sequences can be long (500–1500 bp) cDNA sequences or shorter (25–70 mer) oligonucleotide sequences. The probes can be deposited with a pin or piezoelectric spray on a glass slide, known as spotted array technology. Oligonucleotide sequences can also be synthesized in situ on a silicon chip by photolithographic technology (i.e., Affymetrix GeneChip). Relative quantitative detection of gene expression can be carried out between two samples on one array (spotted array) or by single samples comparing multiple arrays (Affymetrix GeneChip). In spotted array experiments, samples from two sources are labeled with different fluorescent molecules (Cy3 and Cy5) and hybridized together on the same array. The relative fluorescence between each dye on each spot is then recorded and a composite image may be produced. The relative intensities of each channel represent the relative abundance of the RNA or DNA product in each of the two samples. In Affymetrix GeneChip experiments, each sample is labeled with the same dye and hybridized to different arrays. The absolute fluorescent values of each spot may then be scaled and compared with the same spot across arrays. Figure 5 gives an example of a composite image from one spotted array.

Microarray analyses usually include several steps: image analysis and data extraction, data quantification and normalization, identification of differentially expressed genes, and knowledge discovery by data mining techniques such as clustering and classification. Image analysis and data extraction is fully automated and mainly carried out using a commercial software package or a freeware depending on the technology platforms. For example, Affymetrix developed a standard data processing procedures and software for its GeneChips (for detailed information *http://www.affymetrix.com*); GenePix is widely used image analysis software for spotted arrays. For the rest of steps, the detailed procedures may vary depending on the experiment design and goals. We will discuss some of the procedures below.

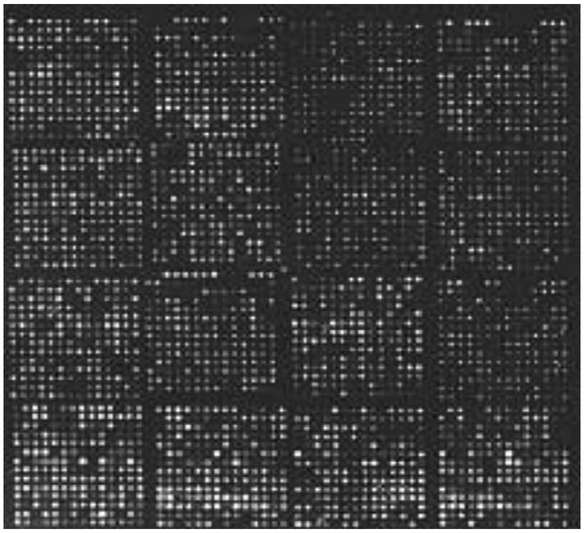


Figure 5. An image from a spotted array after laser scanning. Each spot on the image represents a gene and the intensity of a spot reflects the gene expression.

6.1. Statistical Analysis

The purpose of normalization is to adjust for systematic variations, primarily for labeling and hybridization efficiency, so that we can discover true biological variations as defined by the microarray experiment (53,54). For example, as shown in the self-hybridization scatter plot (Fig. 6) for a two-dye spotted array, variations (dye bias) between dyes is obvious and related to spot intensities. To correct, the dye bias, one can apply the following model:

$$\log_2(R/G) -> \log_2(R/G) - c(A),$$

where R and G are the intensities of the dyes; A is the signal strength ($\log_2(R^*G)/2$); M is the logarithm ratio ($\log_2(R/G)$); c(A) is the locally weighted polynomial regression (LOWESS) fit to the MA plot (55,56).

After correction of systematic variations, we want to determine which genes are significantly changed during the experiment and to assign appropriately adjusted p-values to the genes. For each gene, we wish to test the null hypothesis that the gene is not differentially expressed. P-value is the probability of finding a result by chance. If P-

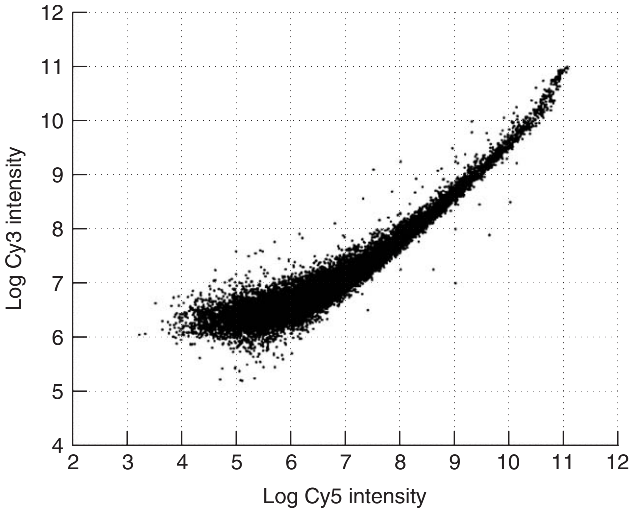


Figure 6. Self-hybridization scatter plot. Y-axis is the intensity from one dye; X-axis is the intensity from the other dye. Each spot is a gene.

value is less than a cut-off (e.g., 0.05), one would reject the null hypothesis and state that the gene is differentially expressed (57). Analysis of variance (ANOVA) is usually used to model the factors for a particular experiment. For example,

$$\log(m_{ijk}) = \mu + A_i + D_j + V_k + \varepsilon_{ijk},$$

where m_{ijk} is the ratio of intensities from the two dye-labelled samples for a gene; μ is the mean of ratios from all replicates; A is the effect of different arrays; D is the dye effects; V is the treatment effects (58). Through F-test, we will determine if the gene exhibits differential expression between any V_k. For a typical microarray, there are thousands of genes. We need to perform thousands of tests in an experiment at the same time, which introduce the statistical problem of multiple testing and adjustment of p-value. Many methods exist for such as purpose including Bonferroni adjustment and False discovery rate (FDR) (59).

For Affymetrix GeneChips analysis, even though the basic steps are the same as spotted microarrays, because of the difference in technology, different statistical methods were developed. Besides the statistical methods provided by Affymetrix, several popular methods are packaged into software such as dChip (60) and RMA (53) in Bioconductor (*http://www.bioconductor.org*). With rapid accumulation of microarray data, one challenging problem is how to compare microarray data across different technology platforms. Some recent studies on data agreements have provided some guidance (61–63).

6.2. Clustering and Classification

Once obtained from the statistical test a list of significant genes, we would apply different data mining techniques to find interesting patterns. At this step the microarray data set is organized as a matrix. Each column represents a condition; each row represents a gene. An entry is the expression level of the gene under the corresponding condition. If a set of genes exhibit the similar fluctuation under all of the conditions, it may indicate that these genes are co-regulated. One way to discover the co-regulated genes is to cluster genes with similar fluctuation patterns using various clustering algorithm. Hierarchical clustering was the first clustering method applied to the problem (64). The result of hierarchical clustering forms a two-dimensional dendrogram as shown in Fig. 7. The measurement used in the clustering process can be either a similarity such as Pearson's correlation coefficient or a distance such as Euclidian distance.

Many different clustering methods have been applied later on, such as, k-means (65), self-organizing map (66), and support vector machine (67). Another type of microarray study involves classification techniques. For example, we can use the gene expression profile to classify cancer types. Golub et al (68) first reported using classification techniques to classify two different types of leukemia as shown in Fig. 8. Many commercial software packages, e.g., GeneSpring and Spotfire, offer the use of these algorithms for microarray analyses.

7. COMPUTATIONAL MODELING AND ANALYSIS OF BIOLOGICAL NETWORKS

Biological system is a complex system involving hundreds of thousands of elements. The interaction among the elements forms an extremely complex networks. With the development of high throughput technologies in functional genomics, proteomics, and metabolomics, one can start looking into the system level mechanisms governing the interactions and properties of biological networks. Network modeling has been used extensively in social and economical fields for many years (69). Many methods can be applied to biological network studies.

The cellular system involves complex interactions between proteins, DNA, RNA, and smaller molecules and can be categorized in three broad subsystem, metabolic networks or pathway, protein networks, and genetic or gene regulatory networks. *Metabolic networks* represent the enzymatic processes within the cell, which provide energy and building blocks for cells. It is formed by the combination of a substrate with an enzyme in a biosynthesis or degradation reaction. Considerable information about metabolic reactions has been accumulated through many years and are organized into large databases, such as KEGG (70), EcoCyc (71) and WIT (72). *Protein networks* refer to the signaling networks where the basic reaction is between two proteins. Protein-protein interactions can be determined systematically using techniques such as yeast two-hybrid system (73) or derived from the text mining of literatures (74). *Genetic networks or regulatory networks* refer to the functional inference of direct causal gene interactions (75). One can conceptualize gene expression as a genetic feedback networks. The networks can be inferred from the gene expression data generated from microarray or proteomics studies in combination with computation modeling.

Metabolic networks are typically represented as a graph with vertex being all the compounds (substrates) and the edges being reactions linking the substrates. With such representation, one can study the general properties of the metabolic networks. It has been shown that metabolic networks exhibit typical property of small world or scale-free network (76,77). The distribution of compound connectivity follows a power law as shown in Fig. 9. There are nodes serving as hubs in the networks. Such property makes the networks quite robust to random deletion of nodes, but vulnerable to selected deletion of nodes. For example, deletion of hub nodes will cause the network collapse very quickly. A recent study also shows that the metabolic networks are organized in modules based on the connectivity, which are correlated with functional classification (78).

Flux analysis is another important aspect in metabolic network study. Building on the stoichiometric network analysis, which only uses the well-characterized network topology, the concept of elementary flux modes was introduced (79,80). An elementary mode is a minimal set of enzymes that could operate at steady state, with the enzymes weighted by the relative flux they need to carry out the mode to function. The total number of elementary modes for given conditions has been used as a quantitative

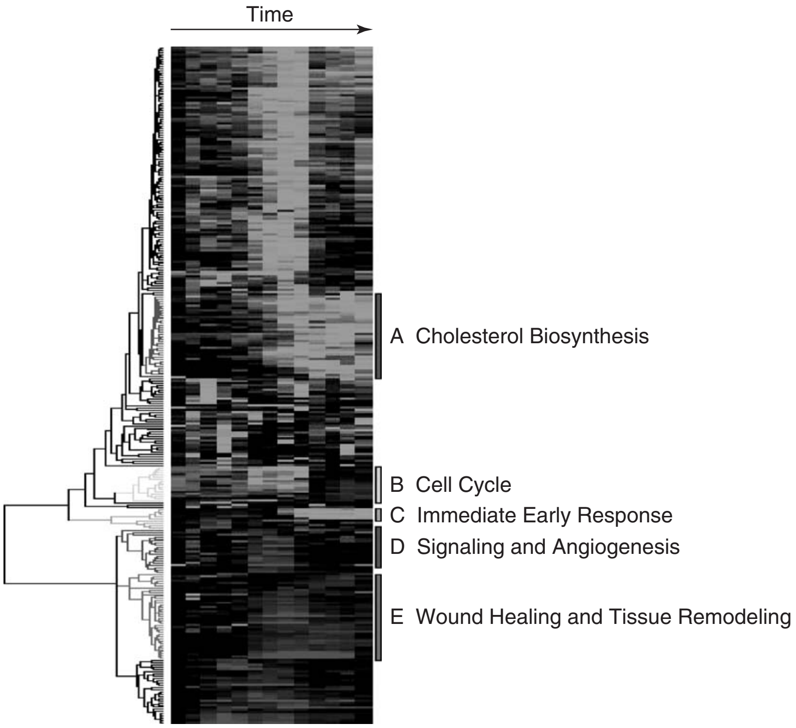


Figure 7. Hierarchical clustering of microarry data. Rows are genes. Columns are RNA samples at different time points. Values are the signals (expression levels), which are represented by the color spectrum. Green represents down-regulation while red represents up-regulation. The color bars beside the dendrogram show the clusters of genes which exhibit similar expression profiles (patterns). The bars are labeled with letters and description of possible biological processes involving the genes in the clusters. (Reprinted from Eisen et al, (64)).

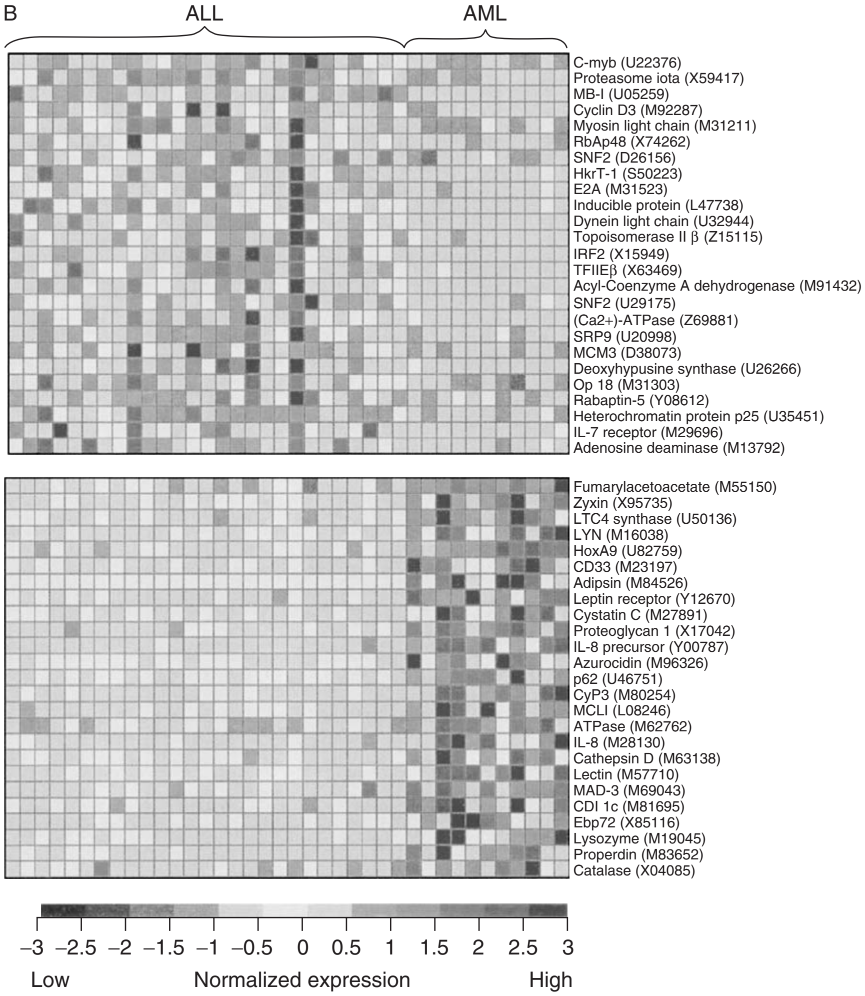


Figure 8. An example of microarray classification. Genes distinguishing acute myeloid leukemia (AML) and acute lymphoblastic leukemia (ALL). The 50 genes most highly correlated with the ALL-AML class distinction are shown. Each row corresponds to a gene, with the columns corresponding to expression levels in different samples. Expression levels for each gene are normalized across the samples such that the mean is 0 and the SD is 1. Expression levels greater than the mean are shaded in red, and those below the mean are shaded in blue. The scale indicates SDs above or below the mean. The top panel shows genes highly expressed in ALL, the bottom panel shows genes more highly expressed in AML. (Reprinted from Golub et al, (68)).

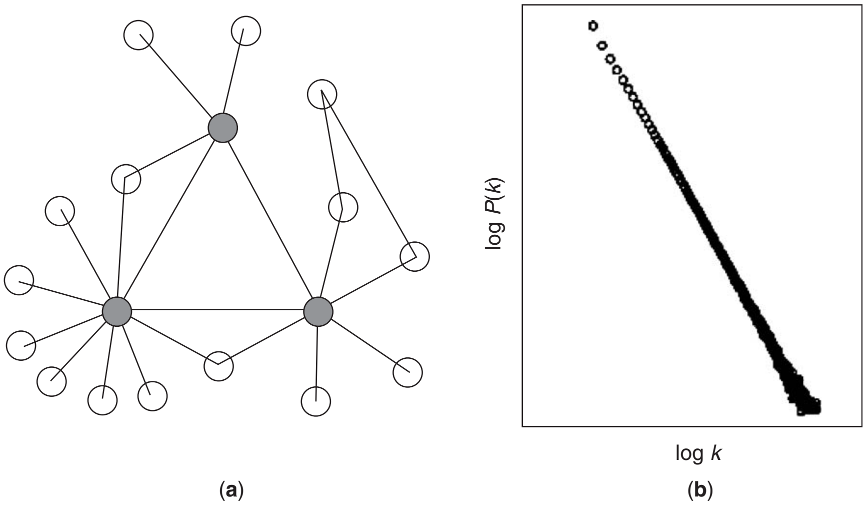


Figure 9. A. In the scale-free network most nodes have only a few links, but a few nodes, called hubs (filled circle), have a very large number of links. B. The network connectivity can be characterized by the probability, $P(k)$, that a node has k links. $P(k)$ for a scale-free network has no well-defined peak, and for large k it decays as a power-law, $P(k) \approx k^{-\gamma}$, appearing as a straight line with slope $-\gamma$ on a log–log plot (Reprinted from Jeong et al, (76)).

measure of network flexibility and as an estimate of fault-tolerance (81,82).

A system approach to model regulatory networks is essential to understand their dynamics. Recently several high-level models have been proposed for the regulatory network including Boolean models, continuous systems of coupled differential equations, and probabilistic model. *Boolean networks* assume that a protein or a gene can be in one of two states, active or inactive, represented by 1 or 0. This binary state varies in time and depends on the state of the other genes and proteins in the network through a discrete equation:

$$X_i(t+1) = F_i[X_i(t), \ldots, X_N(t)].$$

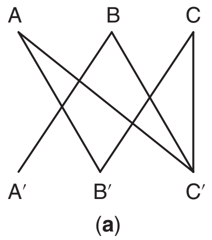
Thus the function F_i is a Boolean function for the update of the *i*th element as a function of the state of the network at time *t* (75). Figure 10 gives a simple example.

Gene expression patterns contain much of the state information of the genetic network and can be measured experimentally. We are facing the challenge of inferring or reverse engineering the internal structure of this genetic network from measurements of its output. Genes with similar temporal expression patterns may share common genetic control processes and may therefore be related functionally. Clustering gene expression patterns according to a similarity or distance measure is the first step toward constructing a wiring diagram for a genetic network (84).

Differential equations can be an alternative model to the Boolean network and applied when the state variables *X* are continuous and satisfy a system of differential equations of the form

$$\frac{dX_i}{dt} = F_i[X_1(t), \ldots, X_N(t), I(t)],$$

where the vector *I(t)* represents some external input into the system. The variables X_i can be interpreted as representing concentrations of proteins or mRNAs. Such model has been used to model biochemical reactions in the metabolic pathways and gene regulation (75).



A′= B
B′ = A or C
C′ = (A and B) or (B and C) or (A and C)

(b)

Input			Output		
A	B	C	A′	B′	C′
0	0	0	0	0	0
0	0	1	0	1	0
0	1	0	1	0	0
0	1	1	1	1	1
1	0	0	0	1	0
1	0	1	0	1	1

(c)

Figure 10. Target Boolean network for reverse engineering. (a) The network wiring and (b) logical rules determine (c) the dynamic output. The challenge lies in inferring (a) and (b) from (c) (Reprinted from Liang et al, (83)).

Bayesian networks are provided by the theory of graphical models in statistics. The basic idea is to approximate a complex multi-dimensional probability distribution using a product of simpler local probability distributions. Generally, a Bayesian network model is based on a directed acyclic graph (DAG) with *N* nodes. In genetic network, the nodes may represent genes or proteins and the random variables X_i levels of activity. The parameters of the model are the local conditional distributions of each random variable given the random variables associated with the parent nodes, whose product yields the joint distribution of all of the random variables:

$$P(X_1, \ldots, X_N) = \prod_i P(X_i | X_j : j \in N^-(i)),$$

where $N^{-}(i)$ denotes all the parents of vertex i. Given a data set D representing expression levels derived using DNA microarray experiments; it is possible to use learning techniques with heuristic approximation methods to infer the network architecture and parameters. Because data from microarray experiments are still limited and insufficient to completely determine a single model, people have developed heuristics for learning classes of models rather than single models, for instance, for a set of co-regulated genes (75) Bayesian networks have been used to combine heterogeneous data sets and applied recently to genetic networks using microarray data (85,86).

In this chapter we reviewed some major development in the field of bioinformatics and introduced some basic concepts in the field covering six areas: sequence analysis, phylogenetic analysis, protein structure analysis, genome analysis, microarray analysis, and network analysis. Due to the limited space, some topics have been left out. One of such topics is text mining, which using Natural Language Processing (NLP) techniques to extract information from the vast amount of literatures in biological research. Text mining has become an integral part in bioinformatics. With the continuing development and maturing of new technologies in many system level studies, the way that we conduct biological research is undergoing revolutionary change. Systems biology is becoming a major theme and driving force. The challenges for bioinformatics in the post-genomics era lie on the integration of data and knowledge from heterogeneous sources and system level modeling and simulation providing molecular mechanism for physiological phenomena.

BIBLIOGRAPHY

1. M. Kimura, *The neutral theory of molecular evolution*. Cambridge University Press, 1983.
2. A. Abbas, S. Holmes, Bioinformatics and Management Science. Some Common Tools and Techniques. *Operations Research*. 2004; **52**(2):165–190.
3. S. B. Needleman, C. D. Wunsch, A General Method Applicable to the Search for Similarities in Amino Acid Sequence of Two Proteins. *J. Mol. Biol*. 1970; **48**:443–453.
4. O. Gotoh, An Improved Algorithm for Matching Biological Sequences. *J. Mol. Biol*. 1982; **162**:705–708.
5. S. Durbin, S. Eddy, A. Krogh, G. Mitchison, *Biological Sequence Analysis*: *Probabilistic models of proteins and nucleic acids*. Cambridge, U.K.: Cambridge University Press, 1998.
6. M. O. Dayhoff, R.M. Schwartz, B. C. Orcut, *A model of evolutionary change in proteins*. Atlas of Protein Sequence and Structure. Vol 5, supplement 3. National Biomedical Research Foundation. Washington, D.C., 1978, pp. 345–352.
7. S. Henikoff, J. G. Henikoff, Amino Acid Substitution Matrices from Protein Blocks. *Proc. Natl. Acad. Sci. USA* 1992; **89**:10915–10919.
8. T. F. Smith and M.S. Waterman, Identification of Common Molecular Subsequences. *J. Mol. Biol*. 1981; **147**:195–197.
9. S. F. Altschul, W. Gish, W. Miller, E. Myers, and J. Lipman, Basic Local Alignment Search Tool. *J. Mol. Biol*. 1990; **215**:403–410.
10. J. D. Lipman, S. F. Altschul, and J. D. Kececioglu, A Tool for Multiple Sequence Alignment. *Proc. Natl. Acad. Sci*. 1989; **86**:4412–4415.
11. J. D. Thompson, D. G. Higgins, and T. J. Gibson, CLUSTAL W: improving the sensitivity of progressive multiple sequence alignment through sequence weighting, position specific gap penalties and weight matrix choice. *Nucleic Acids Research*. 1994; **22**:4673–4680.
12. C. E. Lawrence, S. F. Altschul, M. S. Boguski, J. S. Liu, A. N. Neuwald, J. Wootton, Detecting subtle sequence signals: A Gibbs sampling strategy for multiple alignment. *Science* 1993; **262**:208–214.
13. J. Zhu, J. S. Liu, and C. E. Lawrence, Bayesian Adaptive Sequence Alignment Algorithms. *Bioinformatics*. 1998; 14,25–39.
14. A. L. Delcher, D. Harmon, S. Kasif, O. White, and S. L. Salzberg, Improved microbial gene identification with GLIMMER. *Nucleic Acids Res*. 1999; **27**(23):4636–4641.
15. W. H. Li, 1997. *Molecular Evolution*. Sinauer Assoc, Boston.
16. T. Jukes, C. Cantor, Evolution of Protein Molecules, in: H. N. Munro, ed., *Mammalian Protein Metabolism*. New York: Academic Press, 1969, pp. 21–132.
17. N. Saitou, M. Nei, The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol. Biol. Evol*. 1987; **4**(4):406–425.
18. J. Felsenstein, Evolutionary trees from DNA sequences: a maximum likelihood approach. *J. Mol. Evol*. 1981; **17**(6):368–376
19. J. Huelsenbeck, F. Ronquist, 2002. Mr. Bayes. Bayesian Inference of Phylogeny. http://morphbank.ebc.uu.se/mrbayes/links.php
20. S. Holmes, 2002. Bootstrapping Phylogenetic Trees. To appear in *Statistical Science*. Submitted in (2002).
21. J. Felsenstein, Confidence limits on phylogenies: An approach using the bootstrap. *Evolution* 1985; **39**:783–791.
22. S. Li, D. K. Pearl, H. Doss. Phylogenetic tree construction using MCMC. *J. Am. Statistical Association* 2000; **95**:493–503.
23. M. Levitt, S. Lifson, Refinement of Protein Confirmations using a Macromolecular Energy Minimization Procedure. *J. Mol. Biol*. 1969; **46**:269–279.
24. H. M. Berman, J. Westbrook, Z. Feng, G. Gillil, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne, The Protein Data Bank, *Nucleic Acids Research* 2000; **28**:235–242.
25. Y. Xu, D. Xu, Protein threading using PROSPECT: Design and evaluation. *Proteins: Structure, Function, and Genetics*. 2000; **40**:343–354.
26. M. Levitt and A. Warshel, Computer Simulation of Protein Folding. *Nature* 1975; **253**:694–698.
27. G. Nemethy and H. A. Scheraga, Theoretical Determination of Sterically Allowed Conformations of a Polypeptide Chain by a Computer Method. *Biopolymers* 1965; **3**:155–184.
28. M. Levitt, Molecular Dynamics of Native Protein: Computer Simulation of the Trajectories. *J. Mol. Biol*. 1983; **168**:595–620.
29. D. Beeman, Some Multi-step Methods for Use in Molecular Dynamics Calculations. *J. Comput. Phys*. 1976; **20**:130–139.
30. P. E. Bourne, CASP and CAFASP experiments and their findings. *Methods Biochem. Anal*. 2003; **44**:501–507.

31. D. Gordon, C. Abajian, and P. Green, Consed: a graphical tool for sequence finishing. *Genome Res.* 1998; **8**(3):195–202.
32. D. Gordon, C. Desmarais, and P. Green, Automated finishing with autofinish. *Genome Res.* 2001; **11**(4):614–625.
33. X. Huang and A. Madan, CAP3: A DNA sequence assembly program. *Genome Res.* 1999; **9**(9):868–877.
34. R. H. Waterston, E. S. Lander, and J. E. Sulston, On the sequencing of the human genome. *Proc. Natl. Acad. Sci. USA* 2002; **99**(6):3712–3716.
35. E. W. Myers, et al., A whole-genome assembly of *Drosophila*. 2000; **287**(5461):2196–2204.
36. M. D. Adams, et al., The genome sequence of *Drosophila melanogaster*. *Science* 2000; **287**(5461):2185–2195.
37. J. C. Venter, et al., 2001. The sequence of the human genome. *Science* 2001; **29**:1304–1351.
38. A. V. Lukashin, M. Borodovsky, GeneMark.hmm: new solutions for gene finding. *Nucleic Acids Res.* 1998; **26**(4):1107–1115.
39. C. Burge, and S. Karlin, Prediction of complete gene structures in human genomic DNA. *J. Mol. Biol.* 1997; **268**:78–94.
40. R.-F. Yeh, L. P. Lim, and C. B. Burge, Computational inference of homologous gene structures in the human genome. *Genome Res.* 2001; **11**:803–816.
41. Y. Xu and C. E. Uberbacher, 1997. Automated Gene Identification in Large-Scale Genomic Sequences. *J. Comp. Biol*. 1997; 4:325–338.
42. A. Bateman, L. Coin, R. Durbin, R. D. Finn, V. Hollich, S. Griffiths-Jones, A. Khanna, M. Marshall, S. Moxon, E. L. L. Sonnhammer, D. J. Studholme, C. Yeats, and S. R. Eddy, The Pfam Protein Families Database. *Nucleic Acids Research* 2004; **32**:D138–D141.
43. S. Eddy, Profile hidden Markov models. *Bioinformatics* 1998; **14**:755–763.
44. A. Krogh, M. Brown, I. S. Mian, K. Juolander, and D. Haussler, Hidden Markov models in computational biology applications to protein modeling. *J. Mol. Biol*. 1994; **235**:1501–1531.
45. G. D. Stomo and G. W. Hartzell, Identifying protein-binding sites from unaligned DNA fragments. *Proc. Natl. Acad. Sci*. 1989; **86**:1183–1187.
46. C. E. Lawrence and A. A. Reilly, An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences. *Proteins Struct. Funct. Genet*. 1990; **7**:41–51.
47. L. T. Bailey and C. Elkan, 1994. Fitting a mixture model by expectation maximization to discover motifs in biopolymers. *Proceedings of the Second International Conference on Intelligent Systems for Molecular Biology.* pp. 28-36.
48. M. Kellis, B. W. Birren, and E. S. Lander, Proof and evolutionary analysis of ancient genome duplication in the yeast *Saccharomyces cerevisiae*. *Nature* 2004; **428**:617–624.
49. R. L. Tatusov, E. V. Koonin, and D. J. Lipman, A genomic perspective on protein families. *Science* 1997: 631–637.
50. S. J. O'Brien, M. Menotti-Raymond, W. J. Murphy, W. G. Nash, J. Wienberg, R. Stanyon, N. G. Copeland, N. A. Jenkins, J. E. Womack, and J. A. M. Graves, The promise of comparative genomics in mammals. *Science* 1999; **286**:458–481.
51. G. Bourque and A. P. Pevzner, Genome-scale evolution: reconstructing gene orders in the ancestral species. *Genome Res* 2002; **12**:26–36.
52. Chimpanzee Sequencing and Analysis Consortium. Initial sequence of the chimpanzee genome and comparison with the human genome. *Nature* 2005; 437:69–87.
53. B. M. Bolstad, R. A. Irizarry, M. Astrand, and T. P. Speed, A Comparison of Normalization Methods for High Density oligonucleotide Array Data Based on Bias and Variance. *Bioinformatics* 2003; **19**(2):185–193.
54. Peter Bajcsy, Lei Liu, and Mark Band (2005) DNA microarray image processing in "DNA Array Image Analysis: Nuts&Bolts" Ed. Gerda Kamberova, DNA Press.
55. Y. H. Yang and N. Thorne, 2003. Normalization for Two-color cDNA Microarray Data. Science and Statistics: A Festschrift for Terry Speed. In: D. Goldstein eds., *IMS Lecture Notes, Monograph Series*. Vol 40, pp. 403–418.
56. Y. H. Yang, S. Dudoit, P. Luu, D. M. Lin, V. Peng, J. Ngai, and T. P. Speed. Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. *Nucleic Acids Research* 2002; **30**(4):e15.
57. G. K. Smyth, Y. H. Yang, and T. P. Speed. Statistical issues in microarray data analysis. In: Functional Genomics: Methods and Protocols. In: M. J. Brownstein and A. B. Khodursky eds., *Methods in Molecular Biology*, Volume 224, Totowa, NJ: Humana Press, 2003, pp. 111–136.
58. M. Kerr and G. Churchil, Analysis of variance for gene expression microarray data. *J Comp Biol*. 2000; **7**:819–837.
59. Y. Benjamini and Y. Hochberg, Controlling the false discovery rate: a practical and powerful approach to multiple testing. *J. Royal Statistical Society, Series B*. 1995; **57**(1):289–300.
60. C. Li and W. J. Wong, Model-based analysis of oligonucleotide arrays: expression index computation and outlier detection. *Proc. Natl. Acad. Sci*. 2001; **98**:31–36.
61. H. Wang, X. He, M. Band, C. Wilson, and L. Liu, A study of inter-lab and inter-platform agreement of DNA microarray data. *BMC Genomics* 2005; **6**(1):71.
62. A. C. Culhane, G. Perriere, and D. G. Higgins, Cross-platform comparison and visualisation of gene expression data using co-inertia analysis. *BMC Bioinformatics*. 2003; **4**:59.
63. A. Jarvinen, S. Hautaniemi, H. Edgren, P. Auvinen, J. Saarela, O. Kallioniemi, and O. Monni, Are data from different gene expression microarray platforms comparable? *Genomics* 2004; **83**:1164–1168.
64. M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, Cluster analysis and display of genome-wide expression patterns. *Proc. Natl. Acad. Sci. USA* 1998 95(25):14863–14868.
65. A. Ben-Dor, R. Shamir, and Z. Yakhini, Clustering gene expression patterns. *J. Comp. Biol*. 1999; **6**(3/4):281–297.
66. P. Tamayo, D. Solni, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. S. Lander, and T. R. Golub, Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. *Proc. Natl. Acad. Sci. USA*. 1999; **96**(6):2907–2912.
67. O. Alter, P. O. Brown, and D. Bostein, Singular value decomposition for genome-wide expression data processing and modeling. *Proc. Natl. Acad. Sci. USA* 2000; **97**(18):10101–10106.
68. T. R. Golub, D. K. Slonim, P. Tamayo, C. Huard, M. Gaasenbeek, J. P. Mesirov, H. Coller, M. L. Loh, J. R. Downing, M. A. Caligiuri, C. D. Bloomfield, and E. S. Lander, Molecular Classification of Cancer: Class Discovery and Cass Prediction by Gene Expression Monitoring. *Science* 1999; **286**:531–537.

69. R. V. Sole, R. Ferrer-Cancho, J. M. Montoya, and S. Valverde, Selection, tinkering, and emergence in complex networks. *Complexity* 2003; **8**:20–33.

70. M. Kanehisa, A database for post-genome analysis. *Trends Genet*. 1997; **13**:375–376.

71. I. M. Keseler, J. Collado-Vides, S. Gama-Castro, J. Ingraham, S. Paley, I. T. Paulsen, M. Peralta-Gil, and P. D. Karp, EcoCyc: A comprehensive database resource for Escherichia coli. *Nucleic Acids Research*. 2005; **33**:D334–D337.

72. R. Overbeek, N. Larsen, G. D. Pusch, M. D'Souza, E. Selkov, N. Kyrpides, M. Fonstein, N. Maltsev, and E. Selkov,WIT: integrated system for high-throughput genome sequence analysis and metabolic reconstruction. *Nucleic Acids Res*. 2000; **28**(1):123–125.

73. S. Fields and O. K. Song, A novel genetic system to detect protein-protein interactions. *Nature* 1989; **340**:245–246.

74. N. Daraselia, A. Yuryev, S. Egorov, S. Novichkova, A. Nikitin, and I. Mazo, Extraction of human protein interactions from MEDLINE using full-sentence parser. *Bioinformatics* 2003; **19**:1–8.

75. P. Baldi and G. W. Hatfield, *Microarrays and Gene Expression*. Cambridge, UK: Cambridge University Press, 2001.

76. H. Jeong, B. Tombor, R. Albert1, Z. N. Oltvai, and A. L. Barabási, The large-scale organization of metabolic networks. *Nature* 2000; 407:651–654.

77. A. Wagner and D. A. Fell, The small world inside large metabolic networks. *Proc. R. Soc. Lond. B*. 2001; **268**:1803–1810.

78. R. Guimera and A. L. Nunes Ameral, Functional cartography of complex metabolic networks. *Nature* 2005; **433**:895–900.

79. S. Schuster, C. Hilgetag, J. H. Woods, and D. A. Fell, Reaction routes in biochemical reaction systems: algebraic properties, validated calculation procedure and example from nucleotide metabolism. *J. Math. Biol*. 2002; **45**(2):153–181.

80. S. Schuster, D. A. Fell, and T. Dandekar, A general definition of metabolic pathways useful for systematic organization and analysis of complex metabolic networks. *Nature* 2000; **18**:326–332.

81. J. Stelling, S. Klamt, K. Bettenbrock, S. Schuster, and E. D. Gilles, Metabolic network structure determines key aspects of functionality and regulation. *Nature* 2002; **420**:190–193.

82. T. Cakir, B. Kirdar, and K. O. Ulgen. Metabolic pathway analysis of yeast strengthens the bridge between transcriptomics and metabolic networks. *Biotechnol. Bioeng.* 2004; **86**:251–260.

83. S. Liang, S. Fuhrman, and R. Somogyi, REVEAL, a general reverse engineering algorithm for inference of genetic network architectures. *Pacific Symposium on Biocomputing* 1998; **3**:18–29.

84. R. Somogyi, S. Fuhrman, and X. Wen, *Genetic network inference in computational models and applications to large-scale gene expression data*. Cambridge, MA: MIT Press, 2001.

85. Segal et al., 2003.

86. O. G. Troyanskaya, K. Dolinski, A. B. Owen, R. B. Altman, and D. Botstein, A Bayesian framework for combining heterogeneous data sources for gene function prediction (in *Saccharomyces cerevisiae*). *Proc. Natl. Acad. Sci.* 2003; **100**:8348–8353.

FURTHER READING

S. F. Altschul, T. L. Madden, A. A. Schaffer, J. Zhang, Z. Zhang, W. Miller, D. J. Lipman, Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. *Nucleic Acids Res*. 1997; 25:3389–3402.

P. Baldi, Y. Chauvin, T. Hunkapillar, M. McClure, Hidden Markov Models of Biological Primary Sequence Information. *Proceedings of the National Academy of Sciences of the USA* 1994; **91**:1059–1063.

P. Baldi and S. Brunak, *Bioinformatics: The Machine Learning Approach*. 2nd ed. Cambridge, MA: MIT Press, 2001

P. Bork, T. Dandekar, Y. Diaz-Lazcoz, F. Eisenhaber, M. Huynen, and Y. Yuan, Predicting Function: From Genes to Genomes and Back. *J. Mol. Biol*. 1998; **283**:707–725.

J. Bower and H. Bolouri, *Computational Modeling of Genetic and Biochemical Networks.* Cambridge, MA: MIT Press, 2001.

N. Bray, I. Dubchak, and L. Pachter, AVID: A global alignment program. *Genome Res*. 2003; **13**(1): 97–102.

P. O. Brown and D. Botstein, Exploring the new world of the genome with DNA microarrays. *Nature Genetics* 1999; **21**:33–7.

M. Brudno, C. B. Do, G. M. Cooper, M. F. Kim, E. Davydov, E. D. Green, A. Sidow, and A. Batzoglou, LAGAN and Multi-LAGAN: efficient tools for large-scale multiple alignment of genomic DNA. *Genome Res.* 2003(a); **13**(4):721–31.

M. Brudno, S. Malde, A. Poiakov, C. Do, O. Couronne, I. Dubchak, and A. Batzoglou, Glocal Alignment: Finding Rearrangements during alignment. *Bioinformatics.* Special Issue on the Proceedings of the ISMB 2003, 2003(b); **19**:54i–62i,

S. H. Bryant and S. F. Altschul, Statistics of sequence-structure Threading. *Current Opinion in Structural Biology* 1995; **5**:236–244.

F. E. Cohen, Protein misfolding and prion diseases. *J. Mol. Biol.* 1999; **293**:313–320.

P. Diaconis and S. Holmes, Random walks on trees and matchings. *Electronic J. Probability* 2002; **7**:1–17.

J. C. Doyle, Robustness and dynamics in biological networks. In: *The First International Conference on Systems Biology.* New York/NY: Japan Science and Technology Corporation, MIT Press, 2000

S. Dudoit, J. Fridlyand, and T. P. Speed, Comparison of discrimination methods for the classification of tumors using gene expression data. *J. Am. Statist. Assoc.* 2002; **97**:77–87.

S. Eddy, G. Mitchison, and R. Durbin, Maximum Discrimination Hidden Markov Models of Sequence Consensus. *J. Comput. Biol*. 1995; **2**:9–23.

S. R. Eddy, Non-coding RNA genes and the modern RNA world. *Nature Reviews Genetics*. 2001; **2**:919–929.

B. Efron, E. Halloran, and S. Holmes, Bootstrap confidence levels for phylogenetic trees. *Proc. National Academy Sciences* 1996; **93**:13429–34.

J. S. Farris, The logical basis of phylogenetic analysis, In: N. Platnick and V. Funk, eds., *Advances in Cladistics*. vol. 2, 1983, pp. 7–36.

A. N. Fedorov and T. O. Baldwin, Contranslational Protein Folding. *J. Biol. Chem.* 1997; **272**(52):32715–32718.

J. Felsenstein, PHYLIP, 1993. (Phylogeny Inference Package) version 3.5c., Distributed by the author. Department of Genetics, University of Washington, Seattle. Available: *http://evolution.genetics.washington.edu/phylip.html*

D. Fischer, C. Barret, K. Bryson, A. Elofsson, A. Godzik, D. Jones, K. J. Karplus, L. A. Kelley, R. M. MacCallum, K. Pawowski, B.

Rost, L. Rychlewski, and M. Sternberg, CAFASP-1: critical assessment of fully automated structure prediction methods. *Proteins* 1999; 3:209–17.

W. M. Fitch and E. Margoliash, Construction of phylogenetic trees. *Science* 1967; **155**:279–284.

L. R. Foulds and R. L. Graham, The Steiner problem in Phylogeny is NP-complete. *Advanced Applied Mathematics*. 1982; **3**:43–49.

N. Friedman, M, Linial, I. Nachman, and D. Peter, Using Bayesian Networks to analyze expression data. *J. Comp. Biol.* 2000; 7:601–620.

M. Gardner, *The Last Recreations*. NY: Copernicus-Springer Verlag, 1997.

S. Geman and D. Geman, Stochastic relaxation, Gibbs distribution and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 1984; **6**:721–741.

K. D. Gibson and H. A. Scheraga, Revised algorithms for the build-up procedure for predicting protein conformations by energy minimization. *J. of Comp. Chem.* 1987; **9**:327–355.

P. A. Goloboff, 1995 SPA. (S)ankoff (P)arsimony (A)nalysis, version 1.1. Computer program distributed by J. M. Carpenter, Dept. of Entomology, American Museum of Natural History, New York.

S. Gribaldo and P. Cammarano, The root of the universal tree of life inferred from anciently duplicated genes encoding components of the protein-targeting machinery. *J. Mol. Evol.* 1998; 47(5):508–516.

E. Haeckel, Generelle, Morphologie der Organismen: *Allgemeine Grundzuge der organischen FormenWissenschaft, mechanisch begrundet durch die von Charles Darwin reformirte Descendenz-Theorie.* Berlin: Georg Riemer, 1866.

S. Hannenhalli and P. A. Pevzner, Transforming cabbage into turnip: polynomial algorithm for sorting signed permutations by reversals. *STOC* 1995: 178–189.

J. V. Helden, B. Andre, and J. Collado-Vides, Extracting regulatory sites from the upstream region of yeast genes by computational analysis of oligonucleotide frequencies. *J. Mol. Biol.* 1998; 281:827–842.

E. Hooper, *The River.* Boston: Little, Brown, 1999.

S. Karlin and S. F. Altschul, Methods for assessing the statistical significance of molecular sequences features by using general scoring schemes. *Proc. Nat. Ac. Sci. USA* 1990; **87**(6) 2264–2268.

J. M. Keith, P. Adams, D. Bryant, D. P. Kroese, K. R. Mitchelson, D. A. E. Cochran, G. H. Lala, A simulated annealing algorithm for finding consensus sequences. *Bioinformatics* 2002; 18:1494–1499.

W. J. Kent, BLAT–the BLAST-like alignment tool. *Genome Res.* 2002; **12**(4):656–64.

S. Kirkpatrick, C. D. Gelatt, Jr., and M. P. Vecchi, Optimization by simulated annealing. *Science* 1983; **220**:671–680.

I. Korf, P. Flicek, D. Duan, and M. R. Brent, Integrating genomic homology into gene structure prediction. *Bioinformatics* 2001; **17**:S140–S148.

M. Levitt, Protein Folding by Restrained Energy Minimization and Molecular Dynamics. *J. Mol. Biol.* 1983; **170**:723–764.

D. H. Ly and D. J. Lockhart, R. A. Lerner, and P. G. Schultz, Mitotic misregulation and human aging. *Science* 2000; **287**:1241–1248.

B. Ma, J. Tromp, and M. Li, PatternHunter: Faster And More Sensitive Homology Search. *Bioinformatics* 2002; **18**:440–445.

B. Ma, Z. Wang, and K. Zhang, Alignment between Two Multiple Alignments. In *Combinatorial Pattern Matching*: 14th Annual Symposium, CPM 2003, Morelia, Michoacán, Mexico, June 25-27. *Lecture Notes in Computer Science.* Volume 2676 / 2003, Springer-Verlag Heidelberg, 2003

D. Maddison and W. Maddison. MacClade. 2002. Sinauer. Available: *http://phylogeny.arizona.edu/macclade/*

H. McAdams and L. Shapiro, Circuit Simulation of Genetic Networks. *Science* 1995; **269**:650–656.

N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller. Simulated Annealing. *J. Chem. Phys.* 1953; 21:1087–1092.

E. Mjolsness, D. H. Sharp, and J. Rinetz. A connectionsit model of development. *J. Theor. Biol.* 1991; 152:429–453.

L. B. Morales, R. Garduno-Juarez, and D. Romero, Applications of simulated annealing to the multiple-minima problem in small peptides. *J. Biomol. Struc. Dyn.* 1991; **8**:721–735.

B. Morgenstern, Dialign2: improvement of the segment-to-segment approach to multiple sequence alignment. *Bioinformatics* 1999; **15**:211–218.

J. L. Mountain and L. L. Cavalli-Sforza, Inference of human evolution through cladistic analysis of nuclear DNA restriction polymorphisms. *Proc. Natl. Acad. Sci. USA* 1994; **91**:6515–6519.

U. Muckstein, I. L. Hofacker, and P. F. Stadler, Stochastic pairwise alignments. *Bioinformatics* 2002; **18**(sup. 2):S153–S160.

C. Notredame, D. Higgins, and J. Heringa, T-Coffee: A novel method for multiple sequence alignments. *Journal of Molecular Biology* 2000; **302**:205–217.

M. C. Peitsch, ProMod and Swiss-Model: Internet-based tools for automated comparative protein modeling. *Biochem. Soc. Trans.* 1996; **24**:274–279.

P. A. Pevzner, *Computational molecular biology, an algorithmic approach.* Cambridge, MA: MIT Press, 2000.

U. Pieper, N. Eswar, V. A. Ilyin, A. Stuart, and A. Sali, ModBase, a database of annotated comparative protein structure models. *Nucleic Acids Res.* 2002; **30**:255–259.

G. N. Ramachandran and V. Sasisekharan, Conformation of polypeptides and proteins. *Adv Protein Chem.* 1968; **23**:283–438.

B. Rannala, Z. Yang, Probability distribution of molecular evolutionary trees: a new method of phylogenetic inference. *J. Mol. Evol.* 1996; **43**:304–311.

F. M. Richards, 1991. The Protein Folding Problem. *Scientific American,* pp. 54-63, January.

T. Schlick,. Optimization methods in computational chemistry. In *Reviews in Computational Chemistry,* III, VCH Publishers, 1992, 1-71.

I. Schmulevich, E. Dougherty, S. Kim, and W. Zhang, Probabilistic Boolean Networks: A rule-based uncertainty model for gene regulatory networks. *Bioinformatics* 2002; **18**:261–274.

E. Schröder, Vier combinatorische Probleme. *Z. Math. Phys.* 1870; **15**:361–376.

C. E. Shannon, A mathematical theory of communication. *Bell sys. Tech. Journal* 1948; **27**(379–423):623–656.

M. E. Snow, Powerful simulated annealing algorithm locates global minima of protein folding potentials from multiple starting conformations. *J. Comput. Chem.* 1992; **13**:579–584.

R. Stanley, *Enumerative Combinatorics*. Vol. I, 2nd ed. Cambridge University Press, 1996

J. Stuart, E. Segal, D. Koller, and S. Kim, A Gene Co-Expression Network for Global Discovery of Conserved Genetic Modules. *Science* 2003; **302**(5643):249–55.

D. L. Swofford. PAUP. Phylogenetic analysis using parsimony. V4.0. 2001. Available from Sinauer Associates. Boston, Massachusetts

A. Tozeren and S.W. Byers, *New Biology for Engineers and Computer Scientists*. Englewood Cliffs, NJ: Prentice Hall, 2003.

L. S. Wang, R. Jansen, B. Moret, L. Raubeson, and T. Warnow, Fast phylogenetic methods for the analysis of genome rearrangement data: An empirical study. *Proc. of 7th Pacific Symposium on Biocomputing* 2002.

J. D. Watson and F. H. Crick, A Structure for Deoxyribose Nucleic Acid. *Nature* 1953; (April).

K. P. White, S.A. Rifkin, P. Hurban, and D. D. Hogness, Microanalysis of *drosphila* development during metamorphosis. *Science* 1999; **286**:2179–2184.

H. Winkler, *Verbeitung und Ursache der Parthenogenesis im Pflanzen und Tierreiche*. Jena: Verlag Fischer, 1920.

J. Xu and A. Hagler, Review: Chemoinformatics and drug discovery. *Molecules* 2002; 7:566–600.

Z. Yang and B. Rannala, Bayesian phylogenetic inference using DNA sequences: a Markov chain Monte Carlo method. *Mol. Biol. Evol.* 1997; **14**:717–724.