

CHAPTER 1

BIOINFORMATICS: MINING THE MASSIVE DATA FROM HIGH THROUGHPUT GENOMICS EXPERIMENTS

Haixu Tang* and Sun Kim[†]

*School of Informatics
Center for Genomics and Bioinformatics
Indiana University, Bloomington, IN 47408, USA
*hatang@indiana.edu
[†]sunkim2@indiana.edu*

The recent accomplishment of Human Genome Project (HGP) has revolutionized the life sciences and the medical sciences in many ways. Consequently, the field of bioinformatics has emerged as a new multidisciplinary field that attempts to solve biological and medical problems by analyzing data from high throughput experiments with computational methods. In this chapter we will review the recent development of classical and emerging topics in bioinformatics.

1. Introduction

Human Genome Project (HGP), in which researchers across the world collaborated to determine the whole genetic information in human body, i.e. the human genome, has revolutionized the life and medical sciences in many ways.¹ Among them, an emerging shift of the paradigm in biological research is probably most influencing. Conventionally, biology knowledge was accumulated mainly through a *hypothesis-driven* approach, in which biologists conceive theory for a particular biological problem and then carry out an experiment to test it. In a *hypothesis-driven* approach, experiments are intentionally designed to collect data only relevant to the to-be-tested hypothesis. As a result, these “intentional” data collection often requires several complementary experimental platforms, but produces only a small amount of data. The “hypothesis-driven” approach have been extremely successful and resulted in many critical discoveries in life sciences.

Human genome project, however, has demonstrated a different model of successful biology research. First, the anticipation of this project is great but rather non-specific, and, more importantly, it started without a clear hypothetical theory! Instead, biologists anticipate the generation of a large amount of data (i.e. genome sequences) that may verify or disprove some old hypothesis, and inspire many valuable new theories. Second, this project of “blind” data collection centers on a single technique platform, i.e. DNA sequencing. Indeed, one of the HGP’s goals was to advance the DNA sequencing technology itself for a more efficient data collection. Finally, for the first time in biology research, many biological laboratories across the world work closely and collaboratively on the same project. They carefully planned the project, split the efforts, and share the technologies and results openly with the entire biology community.

Now, the model of HGP is often named as *technology-driven* or *data driven* approach. As HGP has shown, this approach has several distinct features in comparing with the conventional “hypothesis-driven” approach: a high throughput technique platform, a blind collection of large amount of data, and a plan of free data sharing to the community. The success of this model has been copied to several other biological projects, such as the sequencing-based population genetics (HapMap),² microarray-based transcriptomics³ and mass-spectrometry-based proteomics,⁴ thus has given rise to a new kind of life science, often called *genomics*.⁵

The large amount of data generated by genomics indicate a new pathway for biological findings, through computational analysis instead of laboratory experiments. A new multidisciplinary field (now called bioinformatics) emerges, which combines life sciences, computer science and physical sciences to solve biological and medical problems. Bioinformatics offers a playground for the applications of novel approaches to data analysis and data mining. Surprisingly, in spite of its short history, several core algorithms in bioinformatics were developed long before the formation of the discipline. For example, computer scientists have started developing algorithms for comparing DNA sequences with several megabases long time ago, even before HGP was initiated. Nevertheless, the advancement of genome technology always poses new challenges for bioinformaticists. To provide an overview of the current status in bioinformatics research, in this chapter, we will first review the recent development of several classical topics in

bioinformatics, and then introduce a couple of new emerging problems from genome technologies.

2. Recent Development of Classical Topics

It is arguable that the origin of bioinformatics history can be traced back to Mendel's discovery of genetic inheritance in 1865. However, bioinformatics research in a real sense started in late 1960s, symbolized by Dayhoff's *atlas of protein sequences*,⁶ and the early modeling analysis of protein⁷ and RNA⁸ structures. In fact, these early works represented two distinct provenances of bioinformatics: evolution and biochemistry, which still largely define the current bioinformatics research topics.

Bioinformatics is in nature strongly linked to the advancement of genome technologies. As some technologies are proved infeasible in practice or replaced by newer ones, the related bioinformatics topics become outdated. Nevertheless, some classical topics remain important. We will review recent progresses of some of these topics.

2.1. Sequence alignment

The most frequently used computer procedure nowadays in life sciences is sequence alignment, which is also one of the most extensively studied problems in bioinformatics. Important biological molecules, such as nucleic acids (DNAs and RNAs) and proteins, are all linear polymers composed of a limited number of building units (monomers). Hence, they can be often represented as sequences on a small alphabet. For example, a DNA molecule can be represented as a sequence of letters A, C, G and T representing 4 nucleotides {A,C,G,T}, whereas a protein can be represented as a sequence of 20 letters representing 20 different amino acids. To identify similar regions between two sequences, a sequence alignment procedure is applied, in which gaps are inserted and the sequences are shifted accordingly. The following shows a pairwise alignment of two DNA sequences (top and bottom):

```

ACTT—GACCCTATTA—ACTTGCATGCTCTC—ATCAAAA
CCTTTGACCTTAATAACA—CATCCTCTCGCATCGAAA

```

The algorithms to obtain the optimal pairwise alignment between two sequences have been well studied in computer science, known as *string*

pattern matching algorithms.⁹ The early approaches to pairwise sequence alignment problem aimed at aligning two *entire* sequences, now referred to as the *global* alignment problem. A dynamic programming solution to this problem was proposed by Needleman and Wuncsh.¹⁰ However, in many biological applications, global sequence alignment fails to reveal the similarity between two given biological sequences, because the alignment score of the entire sequence is lower than the alignment score between their two subsequences. Smith and Waterman made a small but critical modification of the original dynamic programming algorithm that can solve the *local* alignment problem.¹¹

Dynamic programming algorithms for pairwise sequence alignments are *exact*, i.e. they are guaranteed to report the optimal alignment with a given scoring scheme. Although the optimal alignment is not necessarily the *correct* alignment in a biological sense, we hope to obtain an alignment reflecting the evolutionary process by which these two DNA (or protein) sequences evolved from their common ancestor. Scoring schemes used in sequence alignment usually award identical symbols, and penalize substitutions and gaps based on different evolution models. As a result, scoring schemes affect the resulting alignment in ways as important as the alignment algorithms.¹²

The most important application of pairwise sequence alignment is to find similar sequences of a newly sequenced gene (or protein) in a collection of previously known genes (or proteins), i.e. *gene (or protein) databases*. Thanks to the genome sequencing projects, the size of sequence databases (e.g. Genbank) increases dramatically in the past few years, now achieving 10^{11} bases. Hence, searching such huge databases using dynamic programming algorithm, which takes a quadratic amount of time in relation to the size of query and match sequences, is still too slow. It is not until the invention of the rapid database searching programs (e.g. FASTA¹³ and BLAST¹⁴) that sequence similarity comparison became a popular exercise in molecular biology. Nevertheless, the suggestion of *k-tuple filtering* to speed up sequence comparison, which is essential for FASTA and BLAST, goes back to earlier time.¹⁵ This is not the only example of the foresight of bioinformatics researchers on the increasing algorithmic needs for data analysis in genomics. Even long before the first complete genome was sequenced, computer scientists started thinking of algorithms for comparing

megabase-long genome sequences.¹⁶ Many novel programs developed to align genomic sequences adopt the same *seed-and-extend* strategy, which first identify near-exact matches, then filter them based on various criteria into a reliable subset, called *anchors*, and finally chain them into long pairwise alignments by filling in the gaps between anchors using classical global and local pairwise alignment algorithms.¹⁷ Novel seeding and filtering techniques, such as maximal unique matches (MUMs),¹⁸ maximal exact matches (MEMs),¹⁹ and gapped seeds,²⁰ were applied to acquire accurate genome alignments more rapidly and memory-efficiently. With more and more complete genomes, especially mammalian genomes like mouse genome,²¹ was sequenced, large scale genome sequence alignment programs become essential tools¹⁷ for studying the function and evolution of genes in genomics.

Unlike pairwise alignment, the exact multiple alignment algorithm for large amount of sequences is not feasible.²¹ A straight forward heuristic is known as *progressive alignment*, initially proposed by Feng and Doolittle.²² They aligned the most similar pair of sequences and merged them together to create a new pseudo-sequence called *sequence profile*, thus reduced the problem of aligning the original N sequences into the problem of aligning $N - 1$ sequences, following the concept of “once a gap, always a gap”. After iterating this procedure, a multiple alignment of all original sequences could be built progressively. Similar strategies were also used by many other multiple alignment programs for protein and DNA sequences (e.g. the commonly used program ClustalW²³). In recent years, progressive multiple alignment was also used in aligning multiple genomic sequences, in which the order of alignment was defined based on the previously known phylogenetic tree of the input genomes.²⁴

To improve the accuracy of protein sequence alignment, alternative heuristics for multiple alignment were developed. The divide-and-conquer method applies an empirical rule to divide long sequences into small segments, then uses a dynamic programming algorithm to acquire their multiple alignment and finally combines these small sections of alignments into a long one.²⁵ Another heuristic for multiple alignment is recently implemented in T-Coffee,²⁶ which attempts to use empirical rules to combine the library of every optimal pairwise alignments between input sequences into a multiple alignment.

Despite the long history of research, sequence alignment problem remains one of the hottest topics in bioinformatics. The future developments of sequence alignment algorithms will still be focused on two directions: the alignment efficiency, especially for many long genomic sequences; and the alignment accuracy, especially for protein sequences with low similarities.

2.2. Genome sequencing and fragment assembly

Modern DNA sequencing machine based on Sanger's principle²⁷ can determine the sequence of a short DNA fragment, typically 500–800 base pairs (bps) long. To sequence a long DNA fragment, biologists usually use a shotgun approach: first break the DNA molecule into short overlapping fragments, then sequence each fragment separately until the enough number of fragments are sequenced (typically 10 times of the target DNA size), and finally assemble these fragment sequences into the complete target DNA sequence on computer.

The first fragment assembly program was developed in the same year when the DNA sequencing method was published,²⁸ which used a greedy algorithm to merge the fragment sequences with strong overlaps. Most later developed assembly programs followed a similar three step procedure²⁹:

- *Overlap*: Identifying overlaps between fragments;
- *Layout*: Determining the order of fragments;
- *Consensus*: Deriving the complete DNA sequence.

Due to the potential sequencing errors from DNA sequencing machine (typically 1%), in the popular assembly programs like Phrap,³⁰ the overlaps and layout of the fragment are determined based on not only the sequences of the fragments, but also on the reliability of each nucleotide output from the sequencing (*base calling*).

Conventional assembly programs were very successful in sequencing DNA molecules of medium size (about 200 000 bps). However, it encountered a new challenge when moving toward assembling shotgun fragments of whole genomes. Depending on the genome complexity, various portions of a genome may be present in more than one copy, referred to as *repeats*. It turns out that about 25% of human genome are repetitive sequences. To address this issue, *double-barreled sequencing* was suggested, in which two fragments are sequenced from a same relative long DNA clone and paired

together in assembly.³¹ Many algorithms were developed since then for repeat resolution combining double-barreled data and advanced sequence analysis techniques.³²⁻³⁴ The resulting new assembly programs were successfully used to assemble many large genomes, such as human and mouse genomes.

Although the conventional sequencing technology has accomplished great success in genomics, it remains an expensive experiment. New technologies, pyrosequencing, following the same principle of “sequence by synthesize”,³⁵ were developed towards more affordable experiment for sequencing a higher diversity of genomes. These experiments, however, produced DNA fragments much shorter than the conventional technologies, typically from 20 to 100 bps. It raises new challenges for fragment assembly since the fragment length limits the size of repeats that can be resolved.³⁶ As a result, the *de novo* sequencing of even a small bacterium genome using the new technology may result in many gaps (caused by repeats in the genome).³⁷ The development of new computational methods and tools to overcome this difficulty will be an active research topic in bioinformatics.

2.3. Gene annotation

The first type of analysis that a biologist would want to carry out after a new genome is sequenced is to find genes (often referred to as protein coding regions) within it. The first approach in detecting protein coding regions is to recognize *Open Reading Frames* (ORFs), i.e. a long (typically ≥ 50) sequence of codons (triplets of nucleotides) starting from a Start Codon and ending with a Stop Codon. In addition to its length, protein coding regions have other statistical properties different from the non-coding regions. One of them that is commonly used in current gene finding programs is *codon usage*, which describes the frequencies of 64 possible codons in coding and non-coding regions. High order Markov models are often built using species specific parameters for coding and non-coding regions, respectively.^{38,39} Discriminative approaches can be applied to estimate the conditional probabilities of a given DNA sequence to be within coding or non-coding regions.⁴⁰

The discovery of split genes created another complication for gene annotation in eukaryotic genomes. The coding sequence of a single gene is not

continuous in the genome, forming a number (up to thousands) of segments whose transcripts are joined together in cells through a process called *splicing*. It turns out that there are some sequence signals (“splicing signal”) embedded in the junction between the coding (“exons”) and the non-coding segments (“introns”) that is used to guide the gene splicing in cells. Successful eukaryotic gene annotation tools like Genscan⁴¹ attempt to combine the properties of the coding regions and splicing signals using more complex statistical models (e.g. hidden Markov models, HMMs) to annotate the gene structure.⁴²

Other than the *ab initio* methods mentioned above, gene similarity search can also be used for gene structure prediction. A spliced alignment algorithm, the modified version of the conventional dynamic programming algorithm for pairwise sequence alignment, is proposed to find an assembly of putative exons that is closest to a related protein, thus deriving the gene structure from genomic sequences.⁴³ A useful extension of spliced alignment algorithm is based on the comparison between genome sequences and *Expressed Sequence Tags* (ESTs), rapidly sequenced fragments from message RNAs.⁴⁴ Current gene annotation programs usually integrate both the statistical and similarity searching approaches when annotating genes from a newly sequenced genome, and then provide an option to include putative ESTs from the same organism.⁴⁵

With increasingly closely related genomes being sequenced, a novel approach to gene annotation emerges, based on the comparison of syntenic regions across multiple genome. The concept of this approach is that the coding regions are in general more conserved than non-coding regions in evolution, owing to the selection pressure. Furthermore, the level of conservation in the coding regions is different from one reading frame to another, since the mutations at the third position of synonymous codons do not change the coding amino acids, thus are under lower selection pressure compared to the 1st and 2nd positions. Gene annotation systems now allow the use of more than one genomes for gene structure prediction.^{46,47} Results have shown that the incorporation of multiple genomes across a variety of evolutionary distance can significantly improve gene annotation.⁴⁸ Gene annotation will remain an important research topic in bioinformatics and its accuracy will be continuously pushed to the limit by newly developed methods as well as the accumulated genomic sequences.

2.4. RNA folding

Unlike DNAs, RNAs usually function as single strand molecules. The nucleotides of a single RNA molecule can pair with each other (through hydrogen bonds) and form a stable *secondary structure*. Figure 1 shows the common nomenclature for loops in RNA secondary structures. The stable secondary structure of an RNA molecule is thought to be the one with the lowest free energy, and the problem of finding this stable structure computationally is called RNA folding problem.

RNA secondary structures can be represented by a list of base-pairs in a RNA sequence. An approximate solution to RNA folding problem is to find a secondary structure of a given RNA sequence with the maximal number of base-pairs using a cubic dynamic programming algorithm.^{49,50} More realistic thermodynamic models of RNA folding take into consideration of free energy of loops in addition to base-pairs, and were implemented in commonly used programs such as MFOLD⁵¹ and ViennaRNA.⁵²

Recently, a surprisingly large number of functional RNA molecules encoded by non-coding RNA genes have been found by large scale experimental screening methods. It shows that RNAs play a more important role in cells than biologists initially imagined.⁵³ As a result, computational identification of non-coding RNA genes has become a very important problem. Non-coding RNA genes encode functional RNAs instead of proteins. Hence, they have different statistical properties from the protein coding genes, and the computational methods described above for protein coding gene annotation cannot be applied directly to this problem. It has been shown that the folding energy alone is insufficient to distinguish non-coding RNA

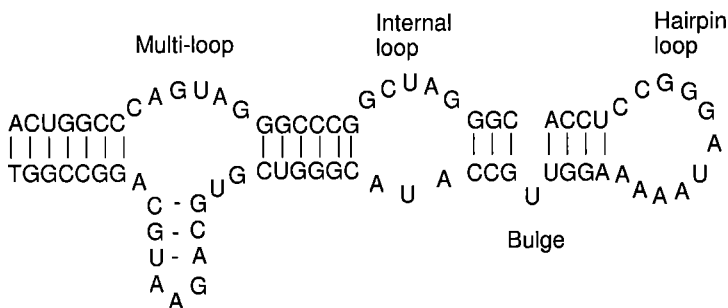


Fig. 1. A schematic illustration of an RNA secondary structure and its loop components.

sequences from the other genome sequences.⁵⁴ On the other hand, automated methods similar to the one used in the first approach to determining the theoretical secondary structure of tRNAs⁸ are developed.⁵⁵ When comparing non-coding RNA genes in different species, it is often found that some substitutions occurred at two sides of a base pair such that the base pair retains. The substitutions are referred to as *compensatory mutations*. Since the structures of non-coding RNAs are important for their functions, many more compensatory mutations can be observed in the aligned non-coding RNA genes than other aligned genomic sequences. This property has been implemented in a few non-coding RNA gene finding programs, using two⁵⁶ or more⁵⁷ aligned RNA sequences from different species. The most significant progress by applying these methods is the discovery of a new class of RNA regulatory elements, *riboswitches*.⁵⁸ As a part of an mRNA molecule, riboswitches can directly bind a small target molecule, and regulate (activate or repress) the gene's activity. Nearly all riboswitch elements were found through the computational analysis of multi-aligned mRNAs that are presumably co-regulated. The conserved secondary structure among these mRNAs then can be identified based on the compensatory mutations in the alignment.⁵⁹ The discovery of riboswitch demonstrates the power of bioinformatics methods in identifying novel molecular elements in biology.

2.5. Motif finding

A sequence motif in a nucleic acid or a protein is referred to a conserved sequence pattern that is determined or conjectured to have a biological function. It was noticed long ago that proteins sharing similar functions may not share sequence similarity along their entire sequences, but only one or a few segments of them, which are often sufficient for proteins carrying out their biological functions.⁶⁰ Similarly, there are also essential sequence patterns in DNAs. A simple example is the palindromic site of the restriction enzyme that activates the DNA cleavage. More complex DNA sequence motifs are those binding sites of Transcriptional Factors (TFs). These short DNA segments (typically 5 to 20 nucleotides long) can bind to TFs and regulate the gene expression. Since the interactions between the binding sites and TFs are often complex, they cannot be represented by a simple DNA sequence (*word*), but a pattern (*motif*).⁶¹

The motif finding problem, i.e. finding the most conserved sequence motif among a set of given DNA (or protein) sequences, has been studied as extensively as the sequence alignment problem in bioinformatics. There are several various formulation of this problem that differ in the rigorous definition of a sequence motif (e.g. *consensus*, *sequence profile* or a set of *words*). The most successful algorithms for solving these problems, however, are not combinatorial algorithms like dynamic programming for solving sequence alignment problem, but probabilistic methods. Gibbs sampling is a procedure to iteratively improve the identified motif, starting from an arbitrarily chosen one.⁶² Another popular motif finding program MEME adopts the Expectation-Maximization algorithm to achieve the same goal.⁶³

In spite of successful applications of these methods, there is still room for further improvement of motif finding, in particular those *weak* motifs that carry on subtle signals over the random noises.⁶⁴ Obviously, an exhaustive searching for all potential motifs will guarantee detection of the motif if there is one in the input sequences. However, there is a tradeoff between the sensitivity and computer time. Advanced probabilistic methods can detect weak motifs within a reasonable computer time,⁶⁵ whereas sophisticated data structure can further speed up the searching process.⁶⁶

2.6. Protein structure prediction

All classical topics we discussed so far are about analyzing the sequences of biomolecules. There is a second source of bioinformatics research coming from the modern biochemistry. During the 1960s, soon after Sanger designed the experimental method for determining the sequence of a protein, Anfinsen concluded from his protein refolding experiments that the native structure of a protein can be determined from its sequence. Anfinsen's theory set one of the most important and difficult goals in bioinformatics, known as the *protein structure prediction problem*.

The early approaches to protein structure prediction were based on the free energy optimization of protein structure. The free energy was evaluated using molecular force fields that describe the physical interactions between atoms, and two types of optimization methods, molecular dynamics and Brownian dynamics, were generally used.

Due to the huge search space for potential protein conformations, pure theoretical methods for protein structure prediction are not very successful

in practice. Biochemists started to look for different approaches. A new type of protein structure prediction methods, referred to as *protein comparative modeling*, were developed based on the same concept that proteins with similar sequences often share structures. Browne and co-workers modeled the structure of α -lactalbumin using the known lysozyme structure as a template, which is the first successful example of comparative modeling.⁶⁷ Since then, several generations of comparative modeling tools were developed and many protein structures were modeled.⁶⁸ The accuracy of comparative protein modeling programs depends on two factors: the identification of an appropriate (known) structure as template, and the alignment between the template and the protein to be modeled (target). When no close homolog exists for modeling, sophisticated methods to address these problems are needed to improve the quality of comparative modeling. *Threading* methods, which attempt to align the target protein sequence with template protein structure(s), can sometimes detect protein similarity beyond their sequence homology.⁶⁹ Other methods for achieving the same goal utilize multiple sequence alignment from the same protein families in template detection as well as template-target alignment to improve their sensitivity.⁷⁰

A significant progress in this area is the recent development of *segment assembly* method for *ab initio* protein structure prediction. The ROSETTA program,⁷¹ which pioneered this strategy to model protein structures by assembling predicted local structural segments, based on the assumption that short sequence segments in proteins almost determine their local structures, and the search space for the global protein structure can be narrowed to the arrangement of these structural segments. ROSETTA and several other programs using similar strategy have performed very well in a series of independent and blind tests, thus pushing forward the practical applications of protein structure prediction in molecular biology.⁷²

3. Emerging Topics from New Genome Technologies

With the advancement of genome technologies, many new research topics in bioinformatics have emerged. Some of them relate to data analysis for specific experimental platforms, whereas others relate to integrating data generated using distinct techniques.

3.1. Comparative genomics: beyond genome comparison

In theory, the full sequence of a genome consists of the most heritable information of an organism. However, the sequence itself is not directly linked to the observable *phenotypes*, which are of the ultimate interests for life and medical scientists and will be the focus of analysis of the available genome sequences. Comparative genomics aims to discover the functional units by comparing multiple genomic sequences,⁷³ based on the principle that the functional units encoded in the genome, e.g. proteins, RNAs and regulatory elements, are conserved across species.⁷²

The fundamental question that comparative genomics has to answer is how to discriminate conserved (and functional) sequence units from the rest of genomic sequences that are under neutral divergence. Depending on different phylogenetic distances between genomes, functional units within different biological systems can be discovered. The first few sequenced eukaryotic genomes, including the yeast, worm and fly genomes, are greater than 1 billion years apart. The comparison of these genomes can reveal a common set of proteins that are responsible for the basic biological functions.⁷⁴ Many genes involving in a large number of pathways can be commonly found in the worm and fly, but not in yeast, reflecting the higher cellular organization complexity of multi-cellular organisms. Despite the conservation of genes with essential function, other functional units, like non-coding RNA genes or the gene regulatory elements, are not anticipated to be conserved over such large evolution distances. In order to study those elements, multiple genomes at moderate evolution distances, e.g. 100 million years apart, should be used. Successful examples of such analysis include the comparison of human and mouse genomes,²¹ two worm genomes,⁷⁵ and multiple yeast genomes.^{76,77} Different biological questions can be addressed when comparing genomes that are very closely related. The comparison of human and chimpanzee genomes (5 million years apart) can reveal the key functional units that are responsible the phenotype difference between similar species.⁷⁸

Genome alignment is the central computational technique used in comparative genomics. The recently developed methods for these problems are scalable to genome scale analysis. With the help of the power of supercomputers, the whole genome alignments now can be built soon after the availability of genome sequences, and made accessible through several

Table 1. Some integrated comparative genomics platforms.

Platforms	Genomes covered	URL	Reference
EnteriX	Prokaryotes	http://globin.cse.psu.edu/enterix/	79
PLATCOM	Prokaryotes	http://platcom.informatics.indiana.edu/	80
Ensembl	Eukaryotes	http://www.ensembl.org/	81
UCSC	Eukaryotes	http://genome.ucsc.edu	82

integrated comparative genomics platforms, for prokaryotic and eukaryotic genomes (Table 1).

The interplay between evolutionary analysis and gene functional prediction is one of the themes in comparative genomics. On one hand, the functions of genes can be predicted through the comparative analysis of their occurrences across multiple genomes. Rigorous evolutionary analysis can distinguish *orthologous genes* from *paralogous genes* in large duplicated gene families. Orthologous genes often carry out the same biological function, thus can be used to improve the straightforward homolog-based gene function annotation.⁸³ Other information derived from comparative genomics, such as gene context⁸⁴ gene fusion,⁸⁴ and phylogenetic profile,⁸⁵ are useful in predicting functions of genes without function-known homologues. On the other hand, the genome-scale annotation of gene functions will provide a complete evolutionary scenario of the transfer and innovation of functions.⁸⁶

Genomes evolve through not only point mutations in individual genes, but also the chromosomal *rearrangement* of gene contents and orders.⁸⁷ Genome duplications, including whole genome duplications and segmental duplications, are known to be important for evolution, in particular innovations of gene function. Comparative genomics can provide solid evidence to trace back those hypothetical events in history.⁸⁸ Chromosomal inversion, fusion/fission are frequently observed rearrangement events. Bioinformatics methods have been developed to elucidate them based on different mathematical models in the context of comparative genomics.⁸⁹

3.2. Pathway reconstruction

Although comparative genomics approaches succeed in predicting the functions of many genes, they fail to annotate 20%–60% genes' function in most

genomes, creating the well known *hypothetical proteins problem*.⁸⁴ Some of the hypothetical proteins are parts of key pathways, and hence, identification of the *missing* genes becomes an important problem for reconstructing the whole pathways in particular genomes. Combining evidences from multiple comparative genomics techniques, it is possible to infer the connection between a function unknown gene and certain cellular processes, thus to suggest putative missing genes for incomplete pathways. Application of this approach has produced valuable pathway reconstructions for newly sequence genomes.⁹⁰ Although these predictions are upon further experimental validation, they provide useful information for metabolic analysis and engineering of bacteria, and development of new medicine.⁹¹ The pathways reconstructed from genomic sequences have been integrated into pathway databases, e.g. Kyoto Encyclopedia of Genes and Genomes (KEGG),⁹² which are accessible through web-based searching.

3.3. Microarray analysis

The development of DNA *microarray* technique is a key technology that facilitates the genome wide analysis of gene expression levels.⁹³ Experimentally, a microarray is a tiny square array, on which thousands of *probes*, each corresponding to a specific gene of interest, are synthesized or placed at a high density. The mRNAs extracted from a sample are labeled with a fluorescent dye and hybridized to the microarray. The expression level of corresponding genes can be measured using the amount of mRNAs that stick to spot of probes.

According to the design of probes, DNA microarray can be classified into two broad categories: the oligonucleotide arrays and cDNA arrays. The oligonucleotide array technology (*GeneChip*) developed by Affymetrix (<http://www.affimetrix.com>) uses *situ* synthesized oligonucleotides as probes, whereas cDNA array technique places cDNA clones on the array as probes.⁹⁴ There are several levels of bioinformatics analysis for microarray experiments. On the bottom level, statistical methods are needed to analyze the scanned image from a microarray experiment to extract fluorescent intensities.⁹⁴ The resulting data needs to be further normalized within a single array to remove the background noise, and across multiple arrays to remove array specific biases.⁹⁶ After these steps, the expression level of each analyzed gene can be obtained and used for the next level

analysis. The medium level data analysis involves hypothesis tests (for two sample comparison) or multi-variable variation analysis (for multiple sample comparison) in an attempt to detect differential gene expression. Finally the high level analysis aims at studying gene functions using gene expression levels from multiple samples and experiments, and also integrating other data resources.

Gene expression profiling is a straightforward application of microarray technique.⁹⁵ The result of a gene expression profiling experiment can be represented by a high dimensional matrix, in which each row represents an analyzed gene, and each column represents an individual microarray experiment, e.g. an environmental condition or a tissue sample. Since genes that are similarly expressed may be functionally related, various clustering methods have been used to recognize groups of genes sharing similar gene expression patterns, which can then be used ultimately to build a global gene regulatory network.⁹⁷ Gene expression profiling can also be used for biomarker discovery and disease diagnosis.⁹⁶ Conceptually, some genes may be differentially expressed in disease tissues and normal tissues, and can be used as biomarkers for early disease diagnosis. The biomarkers can also be used for a detailed classification of diseases that show no clear distinct phenotypes.

In addition to gene expression analysis, microarray techniques are also applied to other problems. Genome tiling arrays⁹⁸ utilize probes spanning the entire genome, thus can be used to detect genome variations, such as *single nucleotide polymorphisms (SNPs)*⁹⁹ and *copy number polymorphisms (CNPs)*,¹⁰⁰ by hybridizing to chromosomal DNAs, and to discover the transcription of new genes and alternative splicing¹⁰¹ by hybridizing to mRNAs. Novel bioinformatics methods are required to analyze the data generated from these experiments.

3.4. Proteomics

While the genome encodes the entire genetic information of a living organism, it is proteins that carry out biological processes. Proteins are *synthesized* using amino acids molecules following the direction encoded in DNA or RNA. Proteomics aims to identify the whole set of proteins inside a cell (*proteome*) and to study their dynamic changes across different physiological conditions. In recent years, because of its high sensitivity, mass

spectroscopy (MS) has become an essential analytical technology in proteomics. In a typical proteomics project, proteins are first separated by liquid chromatography (LC) or electrophoresis, then digested into peptides by proteases (e.g. trypsin) and finally analyzed by tandem mass spectroscopy (MS/MS).¹⁰² In MS/MS instruments, the covalent bonds of peptides are broken at different energy levels and the masses of the resulting fragment ions are measured by MS, which provide valuable information for determining the covalent structures of peptides.

Many bioinformatics methods have been developed to interpret the peptide MS/MS spectra automatically. These methods are often classified into two types according to the methodology they adopt: database searching methods and *de novo* sequencing methods.¹⁰³ For examples, Sequest¹⁰⁴ and Mascot¹⁰⁵ are two most frequently used peptide database searching tools; algorithms have also been designed for *de novo* peptide sequencing.

Quantifying proteins in a complex proteome sample (or comparing protein abundances across different samples), is another focus in the field of proteomics, sometimes referred to as quantitative proteomics. Several labeling techniques applied to various MS instruments including isotopic coded affinity tag (ICAT), mass-coded abundance tagging (MCAT) stable isotopic labeling, and global internal standard technology (GIST).¹⁰⁶ On the other hand, label-free protein quantification approaches attempt to quantify protein abundances directly from high-throughput proteomics analysis. Different measures that can be derived from proteomics experiments and presumably correlated to protein abundance were proposed for different MS instruments. For instance, the integration of extracted ion chromatogram (XIC) peaks is thought to be a good measure for LC/MS experiments¹⁰⁷ and sophisticated data analysis tools have been proposed to improve its accuracy.¹⁰⁸

Proteins undergo different types of modifications after they are translated from mRNA. Many of these post-translational modifications (PTMs) have important biological functions, e.g. phosphorylations in signal transduction. MS-based proteomics approaches have been applied to large scale analysis of site-specific modifications.¹⁰⁹ Although several algorithms have been developed to analyze these data, it remains a challenge in bioinformatics to automatically identify these sites from proteomics data.¹¹⁰

Table 2. Online resources for curated protein-protein interactions.

Database	URL	Reference
Database of Interacting Proteins (DIP)	http://dip.doe-mbi.ucla.edu/	115
The Biomolecular Interaction Network Database (BIND)	http://bind.ca	116
Munich Information Center for Protein Sequences (MIPS)	http://mips.gsf.de/	117

3.5. Protein-protein interaction

Proteins carry out their functions by cooperating with each other as well as other types of biomolecules. Recently, high throughput technologies have been developed to determine the interaction partners of proteins at genome scale.¹¹¹ *In vitro* methods like two hybrid technique¹¹² can determine a pair of proteins that can putatively interact with each other. MS-based methods can identify components of an *in vivo* trapped protein complex.¹¹³ These data are being maintained as protein interaction databases (see Table 2). The availability of the interaction map on the whole proteome has inspired new computational methods to study protein functions and biological processes on a system level.¹¹⁴

4. Conclusion

Bioinformatics is still a young discipline and forming its core research topics. Nevertheless, its data-centric nature and the challenge of analyzing massive high dimensional data have drawn a lot of attention from computer scientists. Bioinformatics has been one of the major resources of new problems for computer science and it will remain in this way in the future.

Acknowledgement

Sun Kim was partially supported by CAREER Award DBI-0237901 from National Science Foundations (USA).

References

1. International Human Genome Sequencing Consortium, *Nature* **409**, 860 (2001).
2. G. McVean, C. C. Spencer and R. Chaix, *PLoS Genetics* **1**, e54 (2005).

3. P. O. Brown and D. Botstein, *Nature Genetics* **21**, 33 (1999).
4. C. L. de Hoog and M. Mann, *Annu. Rev. Genomics Hum. Genet.* **5**, 267 (2004).
5. G. Gibson and S. Muse, *A Primer of Genome Science*, Sinauer Associates, USA, (2004).
6. P. J. McLaughlin, L. T. Hunt and M. O. Dayhoff, *Journal of Human Evolution* **1**, 565 (1972).
7. K. D. Gibson and H. A. Scheraga *Proc. Nat. Acad. Sci., USA* **58**, 420 (1967).
8. M. Levitt, *Nature* **224**, 759 (1969).
9. D. Gusfield, *Algorithms on Strings, Trees, and Sequences: Computer Science and Computational Biology*, Cambridge University Press, England (1997).
10. S. B. Needleman and C. D. Wunsch, *Journal of Molecular Biology* **48**, 443 (1970).
11. T. F. Smith and M. S. Waterman, *Journal of Molecular Biology* **48**, 443 (1970).
12. S. Henikoff, *Curr. Opin. Struct. Biol.* **6**, 353 (1996).
13. D. J. Lipman and W. R. Pearson, *Science* **227**, 1435 (1985).
14. S. Altschul, W. Gish, W. Miller, E. Myers and J. Lipman, *Journal of Molecular Biology* **215**, 403 (1990).
15. J. Dumas and J. Ninio, *Nucleic Acids Research* **10**, 197 (1982).
16. W. Miller, *Bioinformatics* **17**, 391 (2001).
17. S. Batzoglou, *Brief in Bioinformatics* **6**, 6 (2005).
18. J. H. Choi, H. G. Cho and S. Kim, *Comput. Biol. Chem.* **29**, 244 (2005).
19. A. L. Delcher, A. Phillippy, J. Carlton and S. L. Salzberg, *Nucleic Acids Research* **30**, 2478 (2002).
20. B. Ma, J. Tromp and M. Li, *Bioinformatics* **18**, 440 (2002).
21. L. Wang and T. Jiang, *Journal of Computational Biology* **1**, 337 (1994).
22. D. Feng and R. Doolittle, *Journal of Molecular Evolution* **25**, 351 (1987).
23. J. D. Thompson, D. G. Higgins and T. J. Gibson, *Nucleic Acids Research* **22**, 4673 (1994).
24. C. N. Dewey and L. Pachter, *Hum. Mol. Genet.* **15**, R51 (2006).
25. J. Stoye, *Gene* **211**, GC45 (1998).
26. C. Notredame, D. G. Higgins and J. Heringa, *Journal of Molecular Biology* **302**, 205 (2000).
27. F. Sanger, S. Nilken and A. R. Coulson, *Proc. Nat. Acad. Sci., USA* **74**, 5463 (1977).
28. R. Staden, *Nucleic Acids Research* **4**, 4037 (1977).
29. H. Peltola, H. Soderlund and E. Ukkonen, *Nucleic Acids Research*, **12**, 307 (1984).
30. P. Green, *Documentation for Phrap* (1994).
31. J. C. Roach, C. Boysen, K. Wang and L. Hood, *Genomics*, **26**, 345 (1995).
32. G. Myers, *IEEE Computing in Science and Engineering* **1**, 33 (1999).
33. P. A. Pevzner, H. Tang and M. S. Waterman, *Proc. Nat. Acad. Sci., USA* **98**, 9748 (2001).
34. M. Pop, S. L. Salzberg and M. Shumway, *IEEE Computer* **35**, 47 (2002).
35. M. Ronaghi, *Genome Research* **11**, 3 (2001).
36. M. Chaisson, P. A. Pevzner and P. H. Tang, *Bioinformatics* **20**, 2067 (2004).
37. M. Margulies, *et al.*, *Nature* **437**, 376 (2005).
38. R. Staden and A. D. McLachlan, *Nucleic Acids Research* **10**, 141 (1982).
39. J. W. Fickett, *Nucleic Acids Research* **10**, 5318 (1982).
40. M. Borodovsky and J. McIninch, *Computers and Chemistry* **17**, 123 (1993).

41. C. Burge and S. Karlin, *Journal of Molecular Biology* **268**, 78 (1993).
42. M. R. Brent and R. Guigo, *Curr. Opin. Struct. Biol.* **14**, 264 (2004).
43. M. S. Gelfand, A. A. Mironov and P. A. Pevzner, *Proc. Nat. Acad. Sci., USA* **93**, 9061 (1996).
44. A. Lindlof, *Appl. Bioinformatics* **2**, 123 (2003).
45. M. R. Brent, *Genome Research* **15**, 1777 (2005).
46. I. Korf, P. Flicek, D. Duan and M. R. Brent, *Bioinformatics* **17**, S140 (2001).
47. S. S. Gross and M. R. Brent, *J. Comput. Biol.* **13**, 379 (2006).
48. M. Kellis, N. Patterson, M. Endrizzi, B. Birren and E. S. Lander, *Nature* **423**, 241 (2003).
49. R. Nussinov and A. B. Jacobson, *Proc. Nat. Acad. Sci., USA* **77**, 6309 (1980).
50. M. S. Waterman and T. F. Smith, *Math. Biosci.* **42**, 257 (1978).
51. M. Zuker, *Nucleic Acids Research* **31**, 3406 (2003).
52. I. L. Hofacker, W. Fontana, P. F. Stadler, L. S. Bonhoeffer, M. Tacker and P. Schuster, *Monatshefte für Chemie* **125**, 167 (1994).
53. S. R. Eddy, *Nature Rev. Genet.* **2**, 919 (2001).
54. E. Rivas and S. R. Eddy, *Bioinformatics* **16**, 583 (2000).
55. S. R. Eddy, *Cell* **109**, 137 (2002).
56. E. Rivas and S. R. Eddy, *BMC Bioinformatics* **2**, 8 (2001).
57. S. Washietl, I. L. Hofacker and P. F. Stadler, *Proc. Nat. Acad. Sci., USA* **102**, 2454 (2005).
58. M. S. Gelfand, A. A. Mironov, J. Jomantas, Y. I. Kozlov and D. A. Perumov, *Trends Genet.* **15**, 439 (1999).
59. B. J. Tucker and R. R. Breaker, *Curr. Opin. Struct. Biol.* **15**, 342 (2005).
60. R. F. Doolittle, *Science* **214**, 149 (1981).
61. G. D. Stormo, *Annu. Rev. Biophys. Biophys. Chem.* **17**, 241 (1988).
62. C. E. Lawrence, S. F. Altschul, M. S. Boguski, J. S. Liu, A. F. Neuwald and J. C. Wootton, *Science* **262**, 208 (1993).
63. T. L. Bailey and C. Elkan, *Proc. Int. Conf. Intell. Syst. Mol. Biol.* **2**, 28 (1994).
64. S. H. Sze and P. A. Pevzner, *Proc. Int. Conf. Intell. Syst. Mol. Biol.* **8**, 269 (2000).
65. J. Buhler and M. Tompa, *Journal of Computational Biology* **9**, 225 (2002).
66. E. Eskin and P. A. Pevzner, *Bioinformatics* **18**, S354 (2002).
67. M. A. Marti-Renom, A. C. Stuart, A. Fiser, R. Sanchez, F. Melo and A. Sali, *Annu. Rev. Biophys. Biomol. Struct.* **29**, 291 (2000).
68. W. J. Browne, A. C. North, D. C. Phillips, K. Brew, T. C. Vanaman and R. L. Hill, *Journal of Molecular Biology* **42**, 65 (1969).
69. S. H. Bryant and C. E. Lawrence, *Proteins* **16**, 92 (1993).
70. I. Friedberg, L. Jaroszewski, Y. Ye and A. Godzik, *Curr. Opin. Struct. Biol.* **14**, 307 (2004).
71. K. T. Simons, I. Ruczinski, C. Kooperberg, B. A. Fox, C. Bystroff and D. Baker, *Proteins* **34**, 82 (1999).
72. K. T. Simons, C. Strauss and D. Baker, *Journal of Molecular Biology* **306**, 1191 (2001).
73. R. C. Hardison, *PLoS Biology* **1**, e58 (2003).
74. G. M. Rubin, M. D. Yandell, J. R. Wortman, G. L. Miklos and C. R. Nelson, *Science* **287**, 2204 (2000).

75. L. D. Stein, Z. Bao, D. Blasiar, T. Blumenthal, M. R. Brent *et al.*, *PLoS Biology* **1**, e44 (2003).
76. P. F. Cliften, L. W. Hillier, L. Fulton, T. Graves, T. Miner *et al.*, *Genome Research* **11**, 1175 (2001).
77. M. Kellis, N. Patterson, M. Endrizzi, B. Birren and E. S. Lander, *Nature* **423**, 241 (2003).
78. The chimpanzee sequencing and analysis consortium, *Nature* **437**, 69 (2005).
79. L. Florea, C. Riemer, S. Schwartz, Z. Zhang, N. Stojanovic *et al.*, *Nucleic Acids Research* **28**, 3486 (2000).
80. K. Choi, Y. Ma, J. H. Choi and S. Kim, *Bioinformatics* **21**, 2514 (2005).
81. W. J. Kent, C. W. Sugnet, T. S. Furey, K. M. Roskin, T. H. Pringle *et al.*, *Genome Research* **12**, 996 (2002).
82. E. V. Koonin, *Annu. Rev. Genet.* **39**, 309 (2005).
83. A. Osterman and R. Overbeek, Missing genes in metabolic pathways: a comparative genomics approach, *Curr. Opin. Chem. Biol.* **7**, 238 (2003).
84. E. M. Marcotte, M. Pellegrini, H. L. Ng, D. W. Rice, T. O. Yeates and D. Eisenberg, *Science* **285**, 5428 (1999).
85. M. Pellegrini, E. M. Marcotte, M. J. Thompson, D. Eisenberg, and T. O. Yeates, *Proc. Nat. Acad. Sci., USA* **96**, 4285 (1999).
86. L. Aravind, L. M. Iyer and E. V. Koonin, *Curr. Opin. Struct. Biol.* **16** (2006).
87. E. E. Eichler and D. Sankoff, *Science* **301**, 5634 (2006).
88. M. Kellis, B. W. Birren and E. S. Lander, *Nature* **428**, 617 (2004).
89. G. Bourque, G. Tesler and P. A. Pevzner, *Genome Research* **16**, 311 (2006).
90. T. Dandekar and R. Sauerborn, *Pharmacogenomics* **3**, 245 (2002).
91. M. W. Covert, C. H. Schilling, I. Famili, J. S. Edwards, I. I. Goryanin, E. Selkov and B. O. Palsson, *Trends Biochem. Sci.* **26**, 179 (2001).
92. M. Kanehisa, S. Goto, S. Kawashima and A. Nakaya, *Nucleic Acids Res.* **30**, 42 (2002).
93. V. G. Cheung, M. Morley, F. Aguilar, A. Massimi, R. Kucherlapati and G. Childs, *Nature Genetics* **21**, 15 (1999).
94. D. D. Bowtell, *Nature Genetics* **21**, 25 (1999).
95. P. C. Boutros and A. B. Okey, Unsupervised pattern recognition: an introduction to the whys and wherefores of clustering microarray data, *Brief Bioinformics* **6**, 331 (2005).
96. J. S. Verducci, V. F. Melfi, S. Lin, Z. Wang, S. Roy and C. K. Sen, *Physiol. Genomics* **25**, 355 (2006).
97. X. Wu and T. G. Dewey, *Methods Mol. Biol.* **316**, 35 (2006).
98. P. Kapranov, V. I. Sementchenko and T. R. Gingeras, *Brief Funct. Genomic Proteomic* **2**, 47 (2003).
99. A. E. Oostlander, G. A. Meijer and B. Ylstra, *Clin. Genet.* **66**, 488 (2004).
100. B. Ylstra, P. van den Ijssel, B. Carvalho, R. H. Brakenhoff and G. A. Meijer, *Nucleic Acids Res.* **34**, 445 (2006).
101. T. E. Royce, J. S. Rozowsky, P. Bertone, M. Samanta, V. Stolc, S. Weissman, M. Snyder and M. Gerstein, *Trends Genet.* **21**, 466 (2005).
102. J. R. Yates III, *Annu. Rev. Biophys. Biomol. Struct.* **33**, 297 (2004).
103. R. S. Johnson, M. T. Davis, J. A. Taylor and S. D. Patterson, *Methods* **35**, 223 (2005).
104. J. R. Yates, J. K. Eng, A. L. McCormack and D. Schieltz, *Anal. Chem.* **67**, 1426 (1995).

105. D. N. Perkins, D. J. Pappin, D. M. Creasy and J. S. Cottrell, *Electrophoresis* **20**, 3551 (1999).
106. X. Zhang, W. Hines, J. Adamec, J. M. Asara, S. Naylor and F. E. Regnier, *J. Am. Soc. Mass Spectrom.* **16**, 1181 (2005).
107. R. E. Higgs, M. D. Knierman, V. Gelfanova, J. P. Butler and J. E. Hale, *J. Proteome Res.* **4**, 1442 (2005).
108. K. C. Leptos, D. A. Sarracino, J. D. Jaffe, B. Krastins and G. M. Church, *Proteomics*, **157**, 1770 (2006).
109. S. A. Carr, R. S. Annan and J. Huddleston, *Methods Enzymol.* **405**, 82 (2005).
110. D. Tsur, S. Tanner, E. Zandi, V. Bafna and P. A. Pevzner, *Nat. Biotechnol.* **23**, 1562 (2005).
111. J. Piehler, *Curr. Opin. Struct. Biol.* **15**, 4 (2005).
112. J. Miller and I. Stagljar, *Methods Mol. Biol.* **261**, 247 (2004).
113. S. Kaveti and J. R. Engen, *Methods Mol. Biol.* **316**, 179 (2006).
114. M. E. Cusick, N. Klitgord, M. Vidal and D. E. Hill, *Hum. Mol. Genet.* **14**, R171 (2005).
115. L. Salwinski, C. S. Miller, A. J. Smith, F. K. Pettit, J. U. Bowie and D. Eisenberg, *Nucleic Acids Research* **32**, D449 (2004).
116. C. Alfarano, C. E. Andrade, K. Anthony *et al.*, *Nucleic Acids Research* **33**, D418 (2005).
117. H. W. Mewes, D. Frishman, K. F. X. Mayer, M. Munsterkotter, O. Noubibou, P. Pagel and T. Rattei, *Nucleic Acids Research* **34**, D169 (2006).