Introduction to molecular biology and bioinformatics-methods for functional genomics

Taken from: Hvidsten T. R. 2004. Predicting function of genes and proteins from sequence, structure and expression data. Acta Universitatis Upsaliensis. *Comprehensive Summaries of Uppsala Dissertations from the Faculty of Science and Technology 999*. 63 pp. Uppsala. ISBN 91-554-6014-3.

1.1 Introduction to molecular biology

Living organisms are governed by a set of inherited instructions encoded by the four letter alphabet A, G, C and T. The "letters" take physical shape in terms of four different *nucleotides* constituting the basic repeating unit of *deoxyribonucleic acid* (DNA) molecules. Each nucleotide consists of a 5-carbon sugar with a nitrogen base covalently attached[1] to carbon atom 1' and a phosphate group covalently attached to carbon atom 3' or 5'. A DNA molecule is a repeating chain of nucleotides where each phosphate group links carbon atom 3' of the sugar in one nucleotide to carbon atom 5' of the sugar in the neighboring nucleotide. There are four types of nitrogen bases determining the four different nucleotides in DNA (adenine (A), guanine (G), cytosine (C) and thymine (T)), and hence each DNA molecule represents a unique sequence of these four chemical "letters". DNA molecules are furthermore structurally organized in duplexes consisting of two helical DNA molecules coiled around a common axis forming a *double helix*. The two strands of the double helix have opposite directions for linking 3' carbon atoms to 5' carbon atoms (i.e. they are anti-parallel) and are held together by hydrogen bonds[2] between opposite bases in the two strands. An important property of the double helix is that hydrogen bonds only occur between two specific pairs of bases. A only binds to T and C only to G. This means that the two strands are *complementary* with respect to the sequence they encode, conveniently facilitating important processes such as replication and transcription (see below). In eukaryotes[3], the DNA molecules (the *genome*) are systematically packed into a number of chromosomes residing in the nuclei of each cell (in animal cells a small fraction of the DNA is located

[1] Covalent bonds occur when two atoms share a common pair of electrons and are the type of bindings that hold atoms together in molecules.

[2] Polar molecules may have a weak, negative charge at one region and a weak, positive charge elsewhere. Hence, when such molecules are close, the charged region of one molecule may attract the oppositely charged region of a neighboring molecule. These attractions are called hydrogen bonds.

[3] Eukaryotes refer to animals or plants consisting of cells with a membrane-enclosed nucleus and organelles. Organelles are any structure found in the viscous content of the cell (i.e. the cytoplasm).

in mitochondria[4]). The actual number and content of the chromosomes varies from species to species.

The *central dogma of molecular biology* states that the genetic information hard-wired in the DNA is *transcribed* into portable *messenger ribonucleic acid* (mRNA) molecules that are subsequently *translated* into *proteins* (see Figure 1). Except for uracil (U) replacing thymine (T) in the mRNA sequence, a mRNA molecule is an exact copy of a segment of one DNA strand, and carries the information necessary to synthesize one or a small number of proteins. While the DNA may be viewed as a storage device for genetic instructions, proteins actually execute these instructions as enzymes, receptors, storage proteins, transport proteins, transcription factors, signaling molecules, hormones, etc. Exceptions are some RNAs that are not translated into proteins and that perform functions directly (tRNA, rRNA and snRNA are examples of functional RNAs that will be discussed later)

The RNA-encoding segments of the DNA are called *genes*[5]. Transcription of genes into RNAs is performed by RNA *polymerase* enzymes using one of the DNA strands as a template. The double-stranded DNA is unwound during transcription so that the strand acting as a template for the RNA synthesis can form a hybrid with the new, growing RNA. The transcribed RNA is consequently a single strand sequence complementary to the template strand and identical to the DNA strand not acting as a template (except that U replaces T).

[4] Mitochondria are large organelles responsible for most of the energy production in eukaryotic cells.

[5] In contrast, classical Mendelian genetics refer to a gene as an inheritable trait.

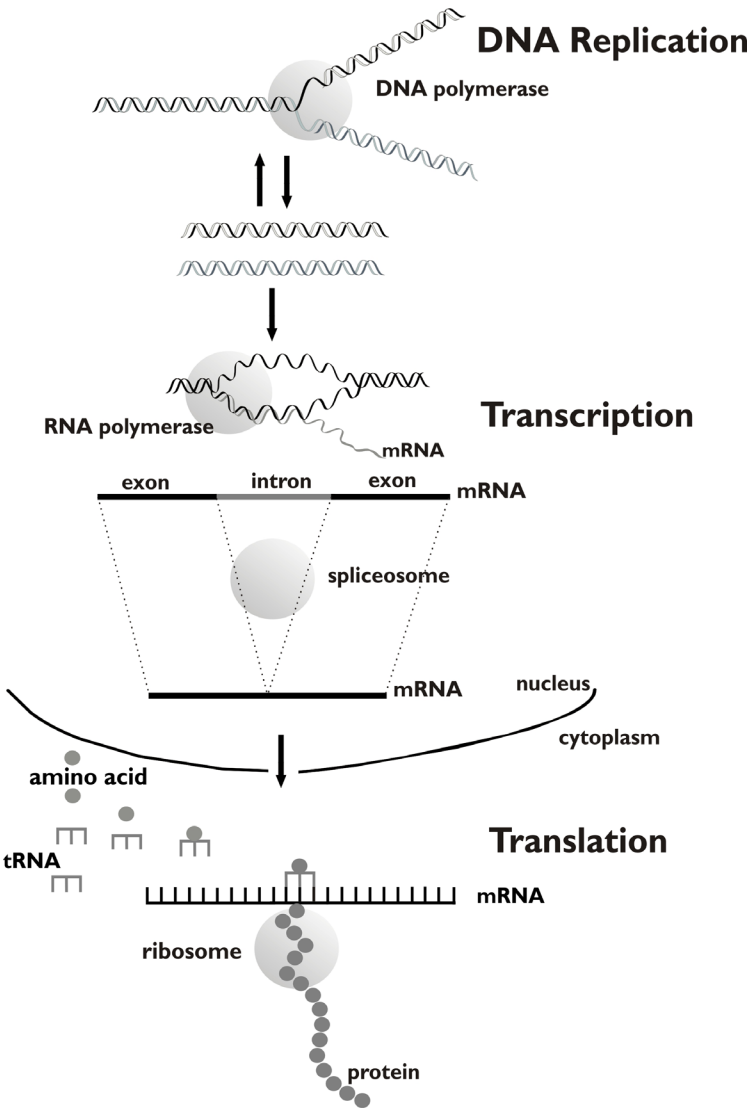

Figure 1. The central dogma of molecular biology (diagrammatic). DNA is transcribed into mRNA that is translated into protein. In addition, DNA is replicated during cell division with the help of DNA polymerase. Transcription is catalyzed by the RNA polymerase. The mRNA is processed by the spliceosome, before translated into a chain of amino acids in the ribosome. tRNA helps the translation by transporting the right amino acids to the right positions as given by the mRNA

Genes are said to be *expressed* in a cell if they are transcribed. The ability to differentially express genes in different cell types, stages of the cell cycle and under various changes in the environment constitute one important level of dynamics in cellular organisms (another level of molecular dynamics is that of

proteins and their interactions with each other and other molecules). A number of factors are important for the differential expression of a particular gene in different cells, including the rate of transcription, the rate of translation and the stability of the protein. However, the most important factor is the initiation of the actual transcription. In eukaryotes, the transcription is not initialized by the RNA polymerase, but by a number of regulatory proteins called *transcription factors* that bind to the DNA and both activate and guide the polymerase. The ability of these transcription factors to selectively recognize specific short sequence elements in DNA is therefore important for the regulation of gene expression (i.e. gene regulation). Many of these regulatory elements or binding sites are in a region called the promoter located upstream of the coding sequence (upstream and downstream refer to the sequences that flank a particular gene at the 5' and 3' ends, respectively).

Most eukaryotic RNA transcripts go through a number of preprocessing steps including the removal of certain segments within the gene and the merging of the remaining segments (RNA *splicing*). This is due to the internal structure of the genes which consists of coding segments called *exons* separated by non-coding regions called *introns*. Although both segments are transcribed, the introns are later removed by a large complex (the *spliceosome*) consisting of five types of small nuclear RNAs (snRNAs) and proteins. Newer studies show that exons in complex organisms such as humans are spliced in different ways, forming different splicing variants and hence different protein products from the same gene [27].

The synthesis of proteins from mRNA takes place in *ribosomes* that function as structural frameworks for translation. Ribosomes are large RNA-protein complexes consisting of a number of ribosomal RNAs (rRNAs) and proteins. The basic building blocks for proteins are *amino acids*. There are 20 amino acids, all consisting of a α-carbon atom (C_α) bound to an amino (NH_2) group, a carboxyl (COOH) group, a hydrogen (H) atom and one variable group determining the 20 different amino acids (the *side chains*). Proteins are simply linear, unbranched chains of amino acids where the amino group of one amino acid forms a peptide bond[6] with the carboxyl group of the neighboring amino acid. The repeating chain without the variable side chains is called the main chain or the *backbone* of the protein molecule. Proteins are coded directly in the mRNA sequence in terms of successive groups of three nucleotides (*codons*). Since there are four different bases in RNA (and DNA) and three base positions in a codon, there are $4^3=64$ possible combinations for coding 20

[6] A peptide bond is a special chemical linage connecting amino acids into linear chains. It is formed by a condensation reaction between the amino group of one amino acid and the carboxyl group of the neighboring amino acid.

amino acids. Hence, each amino acid is specified on average by about three different codons (the genetic code is said to be degenerate). mRNAs are translated into an amino acid chain with the help of transport RNAs (tRNAs). There is one tRNA per amino acid, capable of binding and transporting this specific amino acid. Each tRNA also includes a specific sequence (*anticodon*) that recognizes the relevant codon in the mRNA sequence so that the correct amino acid can be inserted into the growing amino acid chain.

An important principle in molecular biology is that the amino acid sequence of a protein determines its three-dimensional shape (i.e. its *structure*) and furthermore that the structure of a protein determines its function. Since the amino acid sequence is encoded in the DNA, it follows that the mechanisms of evolution (i.e. mutation and crossover) contribute almost directly in changing protein function. To accommodate different three-dimensional conformations, the 20 amino acids vary in shape, charge, hydrophobicity and reactivity. For example, the hydrophobic amino acids tend to be buried inside the protein (where they are protected from the water surrounding the protein), while the hydrophilic amino acids tend to be at the surface of the protein.

Protein structure is more complex than the double helix of DNA (see Figure 2 for an example), and may be organized into four levels. The amino acid sequence itself is referred to as the *primary structure*. When stable, the protein main chain folds into either an α *helix* (i.e. a spiral structure), a β *sheet* (i.e. a planar structure of more than one β strand) or a *coil* (i.e. a random structure) (see Figure 2a). These confirmations constitute the *secondary structure* of proteins. Furthermore, the secondary structure elements (sheets and helices) tend to form simple motifs connected by short U-shaped turns or *loops* often located at the protein surface (e.g. the common hairpin β motif consisting of two neighboring β strands joined by a loop). Several motifs form compact globular domains referred to as the *tertiary structure* of proteins. While secondary structure is stabilized by hydrogen bonds between certain side chains, tertiary structure is mainly stabilized by hydrophobic interactions. Finally, some proteins consist of several amino acid chains (also called subunits) and their arrangements are referred to as the quaternary protein structure. As we have already seen with the spliceosome and the ribosome, proteins often function in large complexes involving several proteins and possibly other macromolecules.

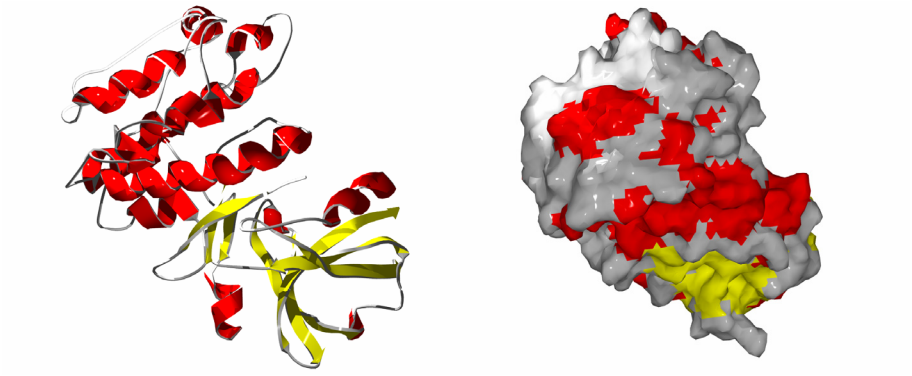

Figure 2. An example protein structure. Helices are colored red, sheets yellow and coils grey. a) shows a cartoon of the protein backbone, while b) shows the protein as a solid molecule. The pictures were generated using Swiss-PdbViewer [20].

1.2 Functional genomics

Biology has traditionally focused on classifying living systems (hierarchically) into increasingly smaller parts, and on studying these parts separately. This reductionistic research approach has culminated in molecular biology, where single molecules in terms of genes and gene products have been studied independently. This way of doing research has of course not been a result of biologists failing to realize the value of understanding the holistic molecular operation of biological systems, but rather a product of the sheer complexity of these systems and the lack of appropriate technology to probe them. With the publishing of the first complete genome sequence in 1995 (the bacteria *Haemophilus influenzae Rd* [18]), the premises have changed. A number of genome sequencing projects are now providing researchers with the basic instructions for the operation of entire organisms at an ever increasing speed (see http://www.genomesonline.org/, [7]). However, although DNA sequence data to some degree has facilitated a transition from molecular *genetics* (i.e. the study of single genes) to *genomics* (i.e. the study of all genes in a genome), genomics is more likely to complement rather than replace traditional use of genetics in understanding the detailed functioning of individual macromolecules [23]. Genomics has also undergone a change from the mapping and sequencing of genomes to the more complex task of determining gene function (i.e. the function of the functional RNAs or proteins coded by the genes) and understanding gene regulation at a genome-wide scale. This part of genomics has been coined *functional genomics*, and has spanned a whole generation of technologies and databases to provide data and support for the statistical and computational analysis making this research possible.

Obviously, although sequence data provide us with the static map of an organism, it seems impossible to reach the goals of functional genomics using this information alone. Additional information, however, may be acquired using the opportunities that sequence data gives to identify genes and gene products and hence obtain data on the actual dynamic expression of the DNA code. One such example is technology for the genome-wide measurement of mRNA levels, providing valuable information on which genes are expressed, and thereby which gene products are active, in a potentially large range of biological contexts. Another example is methods for obtaining structural data. As stated earlier, it is a fundamental biological principle that protein sequence determines structure and that protein structure determines function. However, solving the structure of a protein is a time-consuming task and the amount of structural information therefore lags far behind the vast amount of sequence information. Structural genomics, however, promises to close this gap by combining a systematic approach to solving protein structure experimentally with computational methods for protein structure prediction. The next section will give an introduction to bioinformatics and the use of computers in aiding functional genomics. However, this section will be complemented with a short introduction to DNA sequencing, the high-throughput gene expression measurement technology of microarrays and the two most important experimental methods for solving protein structure. The two latter methods provide additional, partly sequence-derived, data for functional genomics.

1.2.1 DNA sequencing

There are several different techniques for determining the nucleotide sequence of DNA segments (i.e. DNA *sequencing*). In one of the most used approaches, DNA polymerase (which in the organism amongst other functions performs replication) is allowed to copy single stranded DNA segments using both altered nucleotides (*dideoxynucleotides*) and ordinary nucleotides. The alteration of the four dideoxynucleotides, corresponding to the four ordinary nucleotides, has the effect that when added by the polymerase to the growing chain, no further nucleotides can be added to the 3' end afterwards and hence the strand is terminated. Consequently, many fragments of different lengths are created, all with a dideoxynucleotide at the 3' end. A gel solution containing the copied fragments may now be charged by a voltage so that the DNA fragments, which are slightly negative, start traveling towards the positive end of the gel. The speed at which the fragments travel depends on their length and the fragments may therefore be ordered accordingly. The four different dideoxynucleotides are labeled with four different *fluorochromes* that emit four different colors of light when absorbing radiation of specific wavelengths. The dideoxynucleotides at the 3' end may therefore be scanned with a laser and determined from the

resulting image. Furthermore, the nucleotides in each position in the original DNA segment may now also be determined given fragments of all possible lengths. For example, the nucleotide in position 7 in the original segment is determined by the color of the light emitted by the dideoxynucleotide at the 3' end of fragments of length 7.

Whole genomes (or long DNA segments) may be sequenced by first dividing them into many overlapping fragments, then sequencing each of the fragments separately and finally assembling the genome sequence with the help of the overlaps. In addition to DNA, proteins may also be sequenced directly using methods such as Edman degradation.

1.2.2 Microarray technology

The complementary nature of the DNA double helix is of great importance to replication and transcription, and may also be utilized for the large-scale measurement of mRNA levels in cells. Two complementary nucleic acid molecules (i.e. strands) will combine under the right conditions to form double stranded helices. In a reaction vessel this is referred to as *hybridization*. Hence, it is possible to use identified DNA strands (*probes*) to query complex populations of unidentified, complementary strands (*targets*) by checking for hybridization. Microarrays are glass slides or wafers populated with large numbers of strands derived from identified genes. By applying a target sample of unidentified mRNA to the array, the expression level of each gene probe may be quantified from the extent of hybridization between the probes and the targets. Since one slide may contain probes from thousands of genes, one microarray experiment may determine the genome-wide expression state of a cell sample. Furthermore, systematic series of microarray experiments may reveal the specific changes in cellular gene expression associated with different physiological or pathophysiological[7] responses.

The most common microarray technology is that of *DNA microarrays* [15, 36]. DNA microarrays are glass slides with DNA probes robotically printed in spots. Each spot contains probes from the same gene. The target mRNA is reverse transcribed into the more stable cDNA (*complementary* DNA) and is therefore complementary to the original mRNA. The target mRNA comes from two different samples (often called the *test sample* and the *reference sample*) and is separately labeled with the two different fluorescent dyes Cy5 and Cy3. Cy5/Cy3 are chemical groups that emit red/green light when absorbing radiation of particular wavelengths. The two target samples are in solution and

[7] Physiology is the study of life at the organism level in healthy states, while pathophysiology is the study of disease states.

are simultaneously applied to the slide. The microarray is then scanned with a laser, and the two resulting images are analyzed using image analysis software. The intensity of the red and green light from each spot is assumed to be proportional to the amount of hybridized target cDNA labeled with Cy5 and Cy3, respectively. The expression level of each gene is presented as the ratio between the intensity of the red light and the green light, and hence reflects the expression level in the test sample relative to the expression level in the reference.

The most used technology besides that of DNA microarrays is the so-called *GeneChips* manufactured by Affymetrix [19]. This technology uses photo-lithographic techniques from the semiconductor industry to synthesize *oligonucleotides* on glass wafers. These oligonucleotide probes are in general much shorter than DNA probes (20-25 bases compared to 100-2000 bases) and hence less specific to one particular gene. However, oligonucleotides are more sensitive since such short probe strands only form stable double stranded DNA with target strands that match perfectly. Hence, oligonucleotides are more versatile and may be used for example to screen for DNA variations between individuals. Unlike DNA microarrays, oligonucleotide microarrays measure the absolute mRNA level and hence only need one sample. Another advantage is that probes may be synthesized directly from sequence databases, and do not need to be produced in advance. However, the oligonucleotide microarrays are considerably more expensive to produce than DNA microarrays.

A microarray study comprises a number of steps in addition to what has been described here. Obtaining the actual mRNA measurement is preceded by the experimental design (e.g. [14]) and followed by filtering and normalization of the data (e.g. [33]) and computational data analysis (e.g. [1, 32, 38]).

1.2.3 Crystallography and nuclear magnetic resonance

Protein structure is physically determined by *x-ray crystallography* [39] or *nuclear magnetic resonance* (NMR) [43]. Although other methods may give different and complementary information about the structure of proteins, including the primary and quaternary structure, crystallography or NMR are needed to obtain the secondary and tertiary structure since this requires determining the arrangement of atoms within proteins.

The x-ray crystallography method depends on placing a repeating array of many identical molecules (a *crystal*) in an x-ray beam and observing the *diffraction pattern*. The x-ray beam interacts with the electrons of all atoms in the crystal. These interactions scatter x-rays in all direction and only those positively interfering with each other give rise to diffracted beams that may be seen as spots in the

diffraction pattern. To calculate the positions of each atom, the amplitude, wavelength and phase of the diffracted beams are needed. The amplitude is proportional to the intensity of the spot and the wavelength is set by the x-ray source, however, the phase is lost in the diffraction pattern. The so-called *phase problem* is a major problem of crystallography and may be solved by comparing the diffraction data from the original crystal with data from crystals modified with the addition of heavy atoms. An *electron-density map* is then calculated for the repeating molecule in the crystal and interpreted as a structural model. The quality of the model mainly depends on the errors in the phases and the resolution of the diffraction pattern, which in turn depend on the crystal quality. The model is subjected to a computational process where the atoms in the model are shifted about to optimize the fit between the model and the experimental data.

NMR measures the magnetic momentum or *spin* of certain atomic nuclei. Since the spin of atoms is affected by their bonds to other atoms, this method may obtain a list of distance constraints between the atoms of the molecule. A structural model of the protein molecule may then be calculated using these constraints. The main advantage of NMR over crystallography is that the proteins are in solution and do not need to be crystallized. The problems related to obtaining good crystals are the main restriction on the rate at which crystallography produces structural models. The main disadvantage of NMR is that the method cannot currently be applied to large protein molecules and that it requires the protein to have high solubility.

1.3 Bioinformatics

The development of genomics and high-throughput experimental technologies created the need for computers to store and analyze large amounts of data. As was the case for genomics, *bioinformatics* developed from being a discipline mainly associated with sequence databases and sequence analysis to a computational science using biological data to do e.g. functional genomics. Although different definitions and views of bioinformatics exist, most researchers now use bioinformatics as a generic term for both the storage and maintenance of biological data and the use of computational data analysis methods and algorithms in functional genomics-related studies [25]. Bioinformatics thus involves a number of scientific fields including mathematics, statistics, informatics, physics, chemistry, biology and medicine. It is the definition of bioinformatics as data analysis for functional genomics that will be emphasized in this study.

One commonly used methodology in bioinformatics and functional genomics is that of *machine learning*. Machine learning addresses the problem of using

computers to *learn* general concepts from observations and knowledge, and has traditionally been developed in two different schools. Statisticians develop learning methods based on the mathematical frameworks of probability theory and statistics (see e.g. [22, 24]). Computer scientists often develop methods based on models of intelligent systems (e.g. methods inspired by biology such as genetic algorithms and neural networks, or methods based on logic such as rule learning, see the section on machine learning below) [29]. The differences are primarily due to the fact that statisticians have mostly been interested in pure data analysis, while computer scientist have also been interested in building intelligent systems (e.g. robots with *artificial intelligence* [34]). However, these different views are somewhat converging, forming hybrids using elements from both statistics and computer science (e.g. *pattern recognition* [41]).

Induction refers to generalizing from observations to broad concepts and differs from *deduction* that refers to using general concepts (or theories) to infer specific hypotheses. In molecular biology, induction is particularly relevant since the general theories have not yet been worked out. For example, we know that a relationship exists between sequence and structure, but this relationship is not well understood in terms of theories that may be used to deduce good structural models for a particular protein sequence. However, we do have examples of this relationship in terms of protein structures that are experimentally solved. And machine learning methods are designed to induce models based on *examples*, partially describing the assumed underlying functional relationship between, in this case, sequence and structure. The most common application of such models is that of prediction. However, given a model that can reliably predict protein structure from sequence (in particular for unseen proteins, i.e. proteins that were not available when the model was induced), this model obviously includes general concepts that may also be used to understand the relationship. And this understanding may in time lead to general theories. Consequently, machine learning may be used both for *predictive* and for *descriptive* purposes. In molecular biology, and in particular in functional genomics, we will se that a number of problems may be addressed using the concepts of examples and machine learning. And successful application of such methods could lead to situations where biological experiments are used to obtain information on a (representative) set of cases, models are automatically induced from these examples and finally used to fill in the missing knowledge for the remaining cases. This is the philosophy of structural genomics mentioned earlier: to solve the structure of at least one protein from each structural class (e.g. fold, see the section on databases and annotations below) experimentally and to predict the structure of the remaining proteins using sequence similarity to proteins with solved structures.

One of the major obstacles for effective use of machine learning in functional genomics has been the lack of structure in the existing biological knowledge in terms of computer readable databases and annotations. Text mining and automatic inference from free text has therefore been one major part of bioinformatics and will continue to be so (for an overview see [37]). In what follows, a short introduction will be given to relevant databases and annotation efforts. This will be followed by an introduction to the most popular machine learning methods used for utilizing these resources in functional genomics.

1.3.1 Databases and annotations

The Internet provides the infrastructure for accessing and sharing biological information, and has been decisive in the development of functional genomics and bioinformatics. In general, we will divide biological information into measured, unprocessed *data* such as sequences and expressions, and human-processed *knowledge* such as gene function. Data are normally stored in publicly accessible databases, while most biological knowledge is available in terms of published articles. PubMed (http://www.ncbi.nlm.nih.gov/PubMed) is the main electronic free-text database providing access to all biomedical literature in MEDLINE[8]. However, although PubMed in principle includes all available biological knowledge, this knowledge is not easily accessible at the large scale required by functional genomics studies. A biologist may read all articles relevant to one particular gene, but the task of extracting all relevant knowledge on all characterized genes for a genome-wide study is overwhelming. Additionally, this knowledge needs to be structured in a computer readable fashion so that, for example, expression data may be automatically correlated with gene function for a large number of genes. Hence, genomic studies have pushed the formalization of biological knowledge in terms of structured vocabularies that may be used for annotating the databases. A short overview of the most important and relevant databases and annotation efforts will be given next.

The International Nucleotide Sequence Database Collaboration (INSD) consists of DNA Databank of Japan (Japan, http://www.ddbj.nig.ac.jp/, [40]), GenBank (USA, http://www.ncbi.nih.gov/Genbank/, [5]) and EMBL [9] Nucleotide Sequence Database (Europe, http://www.ebi.ac.uk/embl/, [26]). These databases store and maintain all publicly available DNA sequences according to a commonly agreed-upon standard. In addition to sequences of

[8] MEDLINE is the literature database maintained by the National Library of Medicine (NLM) covering the fields of medicine, nursing, dentistry, veterinary medicine, the health care system and the preclinical sciences. It contains abstracts, MeSH terms and other publication details. MeSH is a controlled hierarchical vocabulary used to index the articles.

[9] European Molecular Biology Laboratory (EMBL).

characterized genes, the nucleotide sequence databases include a large number of so-called expressed sequence tags (EST) [8]. ESTs are short sub-sequences of expressed DNA and are synthesized using mRNA as a template (hence the name). Many of these ESTs are not linked to any characterized genes, and are used both for gene discovery and for designing probes for microarray experiments. Since ESTs are short sub-sequences, even non-overlapping ESTs may come from the same gene. UniGene (http://www.ncbi.nlm.nih.gov/UniGene, [42]) is an experimental system attempting to bring some order to the gene/EST sequence data by automatically clustering GenBank sequences into non-redundant sets that correspond to single genes.

Swiss-Prot (http://www.ebi.ac.uk/swissprot/, [4]) is a protein sequence database that together with the TrEMBL supplement (Translated EMBL Nucleotide Sequence Data Library, [4]) contains translated protein sequences for all DNA sequences in the nucleotide sequence databases. In addition, Swiss-Prot provides extensive annotation and cross-references to other databases. Both these databases are now integrated in UniProt (Universal Protein Resource, http://www.ebi.ac.uk/uniprot/, [2]).

The Protein Data Bank (PDB, http://www.rcsb.org/pdb/, [6]) is the major database for protein structures and provides 3D coordinates for all publicly known structures. The Macromolecular Structure Database (MSD, http://www.ebi.ac.uk/msd/, [9]) includes all proteins in PDB and provide extensive annotations and cross-references to other databases such as Swiss-Prot. In addition, two major classification trees exist for protein structures. SCOP (Structural Classification of Proteins, http://scop.berkeley.edu/, [30]) classify protein domains from PDB proteins into three major levels of increasing specificity:

- Fold: Domains are classified to the same fold if their main secondary structure elements have the same relative orientation and connectivity (Protein structure topology may be defined in terms of orientation and connectivity. Orientation refers to the direction of the structural elements in space, while connectivity refers to the order of these elements along the main chain, i.e. how they are connected by the main chain).
- Superfamily: Domains are classified to the same superfamily if their sequence identity is low, but structural and functional features indicate that a common evolutionary origin is probable.

- Family: Domains classified to the same family have a clear evolutionary relationship, and normally have sequence identity greater than 30% or, in some cases where sequence identity is lower, common structural or functional features that provide definitive evidence of an evolutionary relationship.

ASTRAL (http://astral.berkeley.edu/, [13]) provides non-redundant sets of SCOP protein domains and PDB coordinates for these domains. CATH (Class, Architecture, Topology and Homologous superfamily, http://www.biochem.ucl.ac.uk/bsm/cath/, [31]) is the other major classification tree for protein domains providing a similar classification tree to that of SCOP.

Gene expression data are now also published in databases. MIAME (Minimum Information About a Microarray Experiment, [10]) is a standard specifying the information that should be published together with a microarray experiment to facilitate correct interpretation and reproducibility. A number of public databases storing gene expression data are using the MIAME standard, including ArrayExpress (http://www.ebi.ac.uk/arrayexpress/, [11]) and GEO (Gene Expression Omnibus, http://www.ncbi.nlm.nih.gov/geo/, [16]).

The main goal of functional genomics is the genome-wide determination of gene function. Gene Ontology (GO, http://www.geneontology.org, [3]) provides an organism-independent controlled vocabulary for describing the cellular roles of genes and gene products to this end. The ontology is divided into three parts:

- Molecular function: task performed by an individual gene product.
- Biological process: broad biological goal accomplished by an ordered assembly of molecular functions.
- Cellular component: subcellular location where a gene product is active.

Each of the three parts of GO is a *directed acyclic graph* (DAG)[10] where nodes are so called GO terms describing a particular aspect of a molecular function, biological process or cellular component and edges are either is-a or part-of[11]

[10] A graph is defined by a finite set of nodes connected by edges. A directed acyclic graph is a graph where the edges only have one direction (often symbolized by arrows) and where there is no path (i.e. set of connected nodes) starting and ending at the same node.

[11] The is-a relationship between two terms (or nodes) means that one term (the child) is a subclass of the other term (the parent) (e.g. mitotic cell cycle is-a cell cycle). The part-of relationship means that whenever the child exists, it is as part of the parent (but not necessarily the other way around) (e.g. cell cycle is part-of cell proliferation (i.e. cell growth through cell division)).

relations connecting two nodes. GO consequently describes cellular roles at different levels of generality and offers a powerful vocabulary for annotating gene products. An *annotation* in this context is simply an association between a gene or gene product and a GO term. An annotated gene should be associated with at least one GO term from each of the three GO parts (very often biologists find that several terms from each part are needed in order to describe the role of a gene product). Obviously, annotations will reflect the knowledge biologists possess about a certain gene product and may therefore vary in terms of how general they are. The GO graph, however, describes the relationship between different GO terms and therefore provides a way of comparing the annotations of two different gene products. The GO homepage provides annotations for a number of organisms made available by different collaborating groups. The MSD database (see earlier in this section) provides GO annotations for all characterized protein structures in PDB. Finally, there exist several other controlled vocabularies for functional annotation, most notably the MIPS[12] functional catalogue (http://mips.gsf.de/projects/funcat, [28]).

1.3.2 Machine learning

Machine learning deals with the problem of using computers to learn general concepts from *training sets*. A training set consists of a finite number of observations labeled or annotated with class knowledge and is assumed to constitute a partial description of an underlying functional relationship between the observations and the classes. In general, the labels may be continuous values or even more complex structures. However, in this section we will deal with the so called *classification* problem in which the training observations are assumed to belong to a finite set of classes and we want to learn a model or *classifier* capable of assigning an observation to one of these classes. Moreover, we will in general assume two classes, since problems with more than two classes easily may be reduced to a set of two-class problems.

Most machine learning methods represent the observations in terms of *features*. Each observation is a set of measurements, one for each such feature, collectively constituting a *feature vector*. Each observation may alternatively be viewed as a point in the multidimensional space spanned by the features (i.e. the *feature space*). Of course, not all classification problems are easily represented in this way, and choosing the right features is a very important issue specific to each classification problem.

[12] Munich Information center for Protein Sequences (MIPS)

The machine learning methods mainly differ in how they represent the induced model. A number of different designs exist with different advantages and disadvantages. A short overview will be given in the next paragraphs, emphasizing methods that are commonly used in relevant functional genomics studies (see e.g. [24, 29, 41] or specified references for further reading).

Clustering methods

Methods for discovering natural, underlying classes from a set of observations are called *clustering* or *unsupervised learning*. These methods are used when no class knowledge is available. Consequently, methods utilizing labeled training sets are called *supervised learning* reflecting the conceptual idea that a supervisor provides the labels to the learning system.

Clustering methods are divided into *iterative* methods and *hierarchical* methods. The *k-means* algorithm is the most used iterative approach. It starts with a set of *k* randomly chosen clusters of observations and iteratively (a) calculates the center of each cluster (i.e. the *centroid*), (b) assigns each observation to the cluster defined by the closest centroid and (c) returns to (a) until no more observations change clusters. The centroid of a cluster and the closeness of two observations may easily be calculated in the feature space by using e.g. the notion of distance. The *k*-means algorithm is fast and uses little memory, but depends on the initial number and configuration of clusters. A well known related method is that of *self-organizing maps*.

The most popular hierarchical clustering method is *agglomerative* hierarchical clustering. It starts with the observations as single clusters and subsequently merges the two most similar clusters until all observations reside within one big cluster. The distance between two clusters may easily be calculated as the average distance between all pairs of observations in the two clusters (*average linkage*) or the longest/shortest distance between two observations in the two clusters (*complete/single linkage*). The result of the algorithm is a tree of clusters (*dendrogram*) illuminating the similarity structures in the data set. Since the method needs to compute and store the distance between all clusters, it is much slower and uses much more memory than for example the *k*-means algorithm.

Bayes classification rule

The *Bayes classification rule* states that an observation should be assigned to the class with the highest probability given the probability distribution of feature vectors in each class. It may be proven that this rule results in an optimal *error rate* for classification (i.e. fraction of training observations classified to the wrong class). However, the true probability distribution is normally not known and hence needs to be estimated. The difficulty of estimating the distributions

from the training data is why other methods exist and often perform better on real world problems.

There are two basic concepts for estimating probability distributions from data; *parametric* and *non-parametric* methods. A parametric method assumes a distribution structure (e.g. the normal distribution) and calculates its parameters from the data (e.g. average and variance for the one-dimensional normal distribution). A non-parametric method is based on constructing *histograms* from the data using for example Parzen windows or k nearest neighbor density estimation, or simulation methods such as Monte Carlo simulation or bootstrapping. In the one dimensional case, a histogram is constructed by dividing the observations into bins and using the fraction of observations from each bin as probability estimates. In the multidimensional case, however, bins are replaced by hypercubes (e.g. *Parzen windows*). If N observations are needed from each bin to get good probability estimates in the one dimensional case, N^n observations are needed in the n-dimensional case. The dramatic increase in the number of observations needed to get good estimates is often referred to as the "*curse of dimensionality*".

Linear classifiers

Linear classifiers use a line (in two dimensions) or a hyperplane (in multiple dimensions) to separate two classes of observations in feature space. These methods generally consist of a cost function (e.g. error rate) and an optimization algorithm which iteratively changes the parameters defining the hyperplane so that the cost function is minimized over the training set.

If linear classifiers do not yield good results, the problem might be that the classes are not linearly separable. *Artificial neural networks* (ANNs) are one popular method for nonlinear problems and are based on networks of so-called *perceptrons*. A perceptron is a simple computational unit that multiplies each input value with a weight and sums up the products. In principle, the output from the perceptron is 0 if the sum is less than a particular threshold and 1 otherwise. ANNs consist of layers of perceptrons, where the output of each perceptron in one layer is connected to the input of each perceptron in the next layer. The first layer (i.e. the *input layer*) consists of the same number of perceptrons as the number of features and the last layer (i.e. the *output layer*) consists, in the case of two classes, of one perceptron. The network is trained by iteratively inputting the feature vectors to the first layer, calculating the output of each perceptron until the last perceptron, comparing the output value with the true class label and updating the weights for each perceptron by propagating the error backwards in the network (the *backpropagation algorithm*). The training stops when the network is no longer improving its classification.

Another popular method for nonlinear problems is (nonlinear) *support vector machines* (SVMs). The SVMs first map the observations in the feature space into another space using a *kernel function*. A maximally separating hyperplane is then constructed based on the observations closest to the region that separates the two classes (the *support vectors*). The performance of SVMs greatly relies on the choice of kernel function and to what degree the kernel function is able to map the original classification problem into a linearly separable one.

Context-dependent classifiers

A classifier is *context dependent* if the classification does not only depend on the feature vector of one observation, but also on the feature vectors of the other observations and on the dependencies between the classes. The task then becomes to simultaneously assign a class sequence to a sequence of observations. This corresponds to the problem of optimally aligning two sequences and therefore often occurs in DNA and amino acid sequence analysis. One of the most common approaches to this problem is to assume that the class of one observation only depends on the class of the previous observation. This model is called a (first-order) *Markov model* and may be utilized to find the optimal class sequence with a reasonable amount of computation (using e.g. dynamic programming).

k-nearest neighbor classifiers

k-nearest neighbor approaches are based on classifying observations according to the class labels of the k closest training observations in the feature space. This is probably the simplest and most intuitive approach among all supervised methods, and is therefore commonly used.

Decision trees and rule-based classifiers

Decision trees and *rule-based* classifiers work on discrete (i.e. categorical) values or by dividing the feature space into boxes (two dimensions) or hypercubes (multiple dimensions), and by combining these into complex decision surfaces (i.e. surfaces in the feature space separating the classes).

Decision trees classify observations by sorting them down a tree from the root node to the leaf nodes, where the leaf nodes actually provide the classification. Each node corresponds to a feature and redirects the observations to different child nodes depending on their values for that feature. The tree is constructed top-down by iteratively selecting the most class-separating feature as a node.

A related approach is that of learning a set of IF-THEN rules. Note that a decision tree may be represented as a set of rules by translating each path in the tree (from root to leaf) into a rule.

Feature selection

Feature selection refers to the problem of selecting the most important features so as to reduce their number and at the same time retaining class separability allowing classification. There are a number of reasons for doing feature selection. The obvious reason relates to reducing the computational cost of inducing classifiers. However, more important is the fact that the number of features translates directly into the number of *classifier parameters* (e.g. the number of perceptron/weights in an artificial neural network). And there is a fundamental principle in machine learning stating that the higher the ratio between the numbers of training examples and the numbers of classifier parameters, the better the induced classifier will perform on unseen observations (e.g. more observations per dimension/feature gives better estimates of the probability distribution and hence better performance using Bayes classification rule).

There are two broad approaches to feature selection. *Filter* methods select features according to some evaluation criterion (e.g. correlation between the feature and the class knowledge) and then induce a classifier based on these features. *Wrapper* methods use the classifier itself as the evaluation criterion, and select the features that result in the best classification performance.

Feature generation/extraction refers to constructing new features based on different combinations of the old features. One example is rotating the feature space to possibly obtain better class separation (e.g. using principle component analysis).

Bootstrapping, bagging and boosting

Bootstrapping [17] is a general re-sampling method that allows statistical inference about a summary statistic (e.g. sample mean) from a data set without knowing the sample distribution. The idea is to randomly draw with replacement a large number of new data sets from the original data set and to calculate the summary statistic from each such bootstrap sample. This provides several values for the summary statistic which may be used to infer for example its variance or confidence interval.

Bagging [12] and *boosting* [35] are general methods for improving the classification performance of any supervised method. Bagging (bootstrap aggregation) uses bootstrapping to sample a large number of training sets from the original set of examples. A model is induced from each such bootstrap sample and combined (aggregated) during classification to obtain what is often a better classification performance. Boosting is a similar method in which a weight is associated with each training example. Models are iteratively induced from the training set according to these weights and used to re-classify the

examples. The weights are subsequently updated to put more emphasis on incorrectly classified examples. If the applied learning method cannot utilize the weights directly, bootstrap training sets may be constructed according to the weights (i.e. each example is drawn with a probability corresponding to the weight).

Genetic algorithms

Genetic algorithms are used to solve search problems where solutions can be coded as strings of 0's and 1's. An initial population of solutions is generated randomly and the best solutions, according to some fitness function, are iteratively chosen to breed new generations of solutions using genetic operators such as mutation and crossover. Supervised learning involves a number of search problems that may easily be approached with genetic algorithms. One example is feature selection, where each solution may be interpreted as a mask for including or excluding features.

Time complexity

The *big O* notation is used to describe the worst case running time of an algorithm as a function of its input size n. For example, the agglomerative hierarchical clustering algorithm using single linkage has a time complexity of $O(n^2)$ (i.e. it computes the "all-against-all" distance between observations in feature space). Hence, if 100 observations take 10 seconds to cluster, then 10000 observations (which is a typical number of genes in a microarray experiment) take 27.8 hours.

Algorithms that have a worst case running time of $O(n^k)$, where k is a constant, are so-called polynomial-time algorithms. Problems for which no polynomial-time algorithm has yet been discovered are said to belong to the class of *NP-complete* problems (NP stands for non-polynomial). Such problems need to be approached with approximation algorithms that find "good enough" solutions. For example, finding the optimal subset of features (which is the goal of features selection discussed earlier) is NP-complete (i.e. it requires searching trough all 2^n-1 subsets and hence has a time complexity of $O(2^n)$). Feature selection may for example be approached with the wrapper method using a genetic algorithm, or with the filter method using the correlation coefficient between each feature and the class labels. The latter approach of reducing a multi-dimensional problem into considering one dimension at a time (starting with the "best" dimension) is often referred to as a *greedy* approach.

Classifier evaluation

A classifier is best evaluated by applying it to a set of unseen observations (i.e. a *test set*). To obtain good estimates of the true classification performance it is

important to use a test set that is representative for the observations that the classifier is likely to encounter in the future. In practice, it is common to divide the available labeled observations (i.e. examples) randomly into a training set and a test set. The training set is used to induce a classifier and the test set is used for estimating the classification performance. If few observations are available, which is commonly the case, *cross validation* may get the most out of the data in terms of performance estimation. *k*-fold cross validation refers to dividing the examples into *k* equally sized subsets and using one subset for testing and the rest for training. This is done repeatedly so that each subset acts as a test set once and is part of the training set *k-1* times. If *k* equals the number of examples, this method is referred to as *leave-one-out* cross validation. To get good estimates of the classifier performance it is important that information contained in the test set is not used in the training. For example, feature selection should be done after splitting the available examples into training and test sets. Doing feature selection on all available examples implies using the class knowledge contained in future test sets to induce the classifier and hence may lead to optimistic estimates of the true classification performance.

Performance measures and ROC analysis

A number of statistics exist for measuring the performance of a classifier on a test set. *Accuracy* is simply the fraction of test observations classified to the correct class (error rate = 1-accuracy). However, accuracy may provide insufficient information when the classes contain different numbers of examples or when making one type of error is more severe than making another.

Given two classes of positive and negative observations,

- *false positives* (FP) are negative observations classified to the positive class,
- *false negatives* (FN) are positive observations classified to the negative class,
- *true positives* (TP) are correctly classified positive observations and
- *true negatives* (TN) are correctly classified negative observations.

Furthermore, *sensitivity* and *specificity* are the fractions of correctly classified positive and negative observations, respectively (i.e. TP/(TP+FN) and TN/(TN+FP)). Many classification methods do not perform classification directly, but rather output a value representing the certainty that a test observation belongs to the positive class. Hence, we are left with the problem

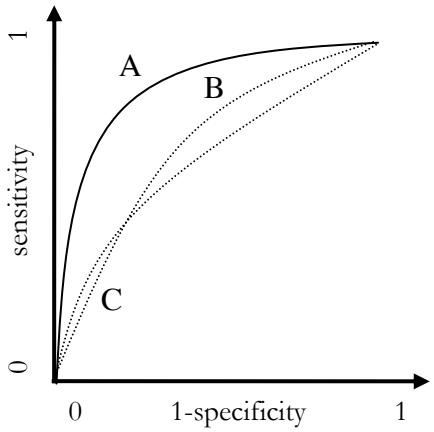

Figure 3. Example ROC curves. Clearly, classifier A performs better than both B and C. However, classifier B only performs better than C on low threshold values, while C performs better than B on high threshold values. Nonetheless, the AUC value of B is larger than that of C.

of choosing a certainty threshold for selecting the positive class as the classification. The *receiver operating characteristic* (ROC) curve may be constructed by plotting sensitivity against specificity for the full range of possible threshold values (see Figure 3). A number of classification applications are associated with different costs for making a false positive classification compared to making a false negative classification. The ROC curve graphically displays the threshold-independent classification performance and provides a vehicle for controlling the number of false positives and false negatives. Increasing the threshold value reduces the number of false positives, but at the same time increases the number of false negatives. The *area under the ROC curve* (AUC, [21]) is often used to measure the threshold independent classification performance using one single number (i.e. AUC equal to 1 signifies a perfect discrimination of the positive and negative examples, while AUC equal to 0.5 signifies no discriminatory capability at all). The standard error of this measure is calculated using the Hanley-McNeil formula [21]. However, one should be aware that two ROC curves obtained using two competing classifiers may intersect and hence indicate that one classifier performs better for one range of threshold values, while the other performs better for another range of threshold values (see Figure 3). This information is of course lost when computing the AUC value.

Overfitting and classifier selection

A classifier is said to *overfit* the training set if there exists another classifier that performs worse on the training set, but better on the test set. A general principle for handling overfitting is related to the principle of *Occam's razor* which states that the simplest model fitting the data should be used. Hence, according to this principle we should for example use the artificial neural network with the fewest perceptrons classifying the training set satisfactorily. This principle also applies to choosing a classification method. One should for example avoid using a nonlinear method on a linearly separable classification

problem. This is of course related to the principle that the ratio between the number of training observations and the number of classifier parameters should be as large as possible (see the discussion in the feature selection paragraph above) More in-depth discussions on issues related to so-called *learning theory* may be found in e.g. [29, 41].

References

1. Altman, R. B. and Raychaudhuri, S. Whole-genome expression analysis: challenges beyond clustering. *Curr Opin Struct Biol.* 11(3): 340-7, 2001.
2. Apweiler, R., Bairoch, A., Wu, C. H., Barker, W. C., Boeckmann, B., Ferro, S., Gasteiger, E., Huang, H., Lopez, R., Magrane, M.*, et al.* UniProt: the Universal Protein knowledgebase. *Nucleic Acids Res.* 32 Database issue: D115-9, 2004.
3. Ashburner, M., Ball, C. A., Blake, J. A., Botstein, D., Butler, H., Cherry, J. M., Davis, A. P., Dolinski, K., Dwight, S. S., Eppig, J. T.*, et al.* Gene ontology: tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet.* 25(1): 25-9, 2000.
4. Bairoch, A. and Apweiler, R. The SWISS-PROT protein sequence database and its supplement TrEMBL in 2000. *Nucleic Acids Res.* 28(1): 45-8, 2000.
5. Benson, D. A., Karsch-Mizrachi, I., Lipman, D. J., Ostell, J. and Wheeler, D. L. GenBank: update. *Nucleic Acids Res.* 32 Database issue: D23-6, 2004.
6. Berman, H. M., Westbrook, J., Feng, Z., Gilliland, G., Bhat, T. N., Weissig, H., Shindyalov, I. N. and Bourne, P. E. The Protein Data Bank. *Nucleic Acids Res.* 28(1): 235-42, 2000.
7. Bernal, A., Ear, U. and Kyrpides, N. Genomes OnLine Database (GOLD): a monitor of genome projects world-wide. *Nucleic Acids Res.* 29(1): 126-7, 2001.
8. Boguski, M. S., Lowe, T. M. and Tolstoshev, C. M. dbEST--database for "expressed sequence tags". *Nat Genet.* 4(4): 332-3, 1993.
9. Boutselakis, H., Dimitropoulos, D., Fillon, J., Golovin, A., Henrick, K., Hussain, A., Ionides, J., John, M., Keller, P. A., Krissinel, E.*, et al.* E-MSD: the European Bioinformatics Institute Macromolecular Structure Database. *Nucleic Acids Res.* 31(1): 458-62, 2003.
10. Brazma, A., Hingamp, P., Quackenbush, J., Sherlock, G., Spellman, P., Stoeckert, C., Aach, J., Ansorge, W., Ball, C. A., Causton, H. C.*, et al.* Minimum information about a microarray experiment (MIAME)-toward standards for microarray data. *Nat Genet.* 29(4): 365-71, 2001.
11. Brazma, A., Parkinson, H., Sarkans, U., Shojatalab, M., Vilo, J., Abeygunawardena, N., Holloway, E., Kapushesky, M., Kemmeren, P., Lara, G. G.*, et al.* ArrayExpress--a public repository for microarray gene expression data at the EBI. *Nucleic Acids Res.* 31(1): 68-71, 2003.
12. Breiman, L. Bagging predictors. *Machine learning.* 24: 123-140, 1996.
13. Chandonia, J. M., Hon, G., Walker, N. S., Lo Conte, L., Koehl, P., Levitt, M. and Brenner, S. E. The ASTRAL Compendium in 2004. *Nucleic Acids Res.* 32 Database issue: D189-92, 2004.

14. Churchill, G. A. Fundamentals of experimental design for cDNA microarrays. *Nat Genet*. 32 Suppl: 490-5, 2002.
15. Duggan, D. J., Bittner, M., Chen, Y., Meltzer, P. and Trent, J. M. Expression profiling using cDNA microarrays. *Nat Genet*. 21(1 Suppl): 10-4, 1999.
16. Edgar, R., Domrachev, M. and Lash, A. E. Gene Expression Omnibus: NCBI gene expression and hybridization array data repository. *Nucleic Acids Res*. 30(1): 207-10, 2002.
17. Efron, B. and Tibshirani, R. J. *An introduction to the Bootstrap*. Chapman & Hall, London, 1993.
18. Fleischmann, R. D., Adams, M. D., White, O., Clayton, R. A., Kirkness, E. F., Kerlavage, A. R., Bult, C. J., Tomb, J. F., Dougherty, B. A., Merrick, J. M*., et al.* Whole-genome random sequencing and assembly of Haemophilus influenzae Rd. *Science*. 269(5223): 496-512, 1995.
19. Fodor, S. P., Read, J. L., Pirrung, M. C., Stryer, L., Lu, A. T. and Solas, D. Light-directed, spatially addressable parallel chemical synthesis. *Science*. 251(4995): 767-73, 1991.
20. Guex, N. and Peitsch, M. C. SWISS-MODEL and the Swiss-PdbViewer: an environment for comparative protein modeling. *Electrophoresis*. 18(15): 2714-23, 1997.
21. Hanley, J. A. and McNeil, B. J. The meaning and use of the area under a receiver operating characteristic (ROC) curve. *Radiology*. 143: 29-36, 1982.
22. Hastie, T., Tibshirani, R. J. and Friedman, J. *The Elements of Statistical Learning*. Springer, New York, 2001.
23. Hieter, P. and Boguski, M. Functional genomics: it's all how you read it. *Science*. 278(5338): 601-2, 1997.
24. Johnson, R. A. and Wichern, D. W. *Applied multivariate statistical analysis*. Prentice Hall, Upper Saddle River, N.J., 2002.
25. Kanehisa, M. and Bork, P. Bioinformatics in the post-sequence era. *Nat Genet*. 33 Suppl: 305-10, 2003.
26. Kulikova, T., Aldebert, P., Althorpe, N., Baker, W., Bates, K., Browne, P., van den Broek, A., Cochrane, G., Duggan, K., Eberhardt, R*., et al.* The EMBL Nucleotide Sequence Database. *Nucleic Acids Res*. 32 Database issue: D27-30, 2004.
27. Lander, E. S., Linton, L. M., Birren, B., Nusbaum, C., Zody, M. C., Baldwin, J., Devon, K., Dewar, K., Doyle, M., FitzHugh, W*., et al.* Initial sequencing and analysis of the human genome. *Nature*. 409(6822): 860-921, 2001.
28. Mewes, H. W., Frishman, D., Guldener, U., Mannhaupt, G., Mayer, K., Mokrejs, M., Morgenstern, B., Munsterkotter, M., Rudd, S. and Weil, B. MIPS: a database for genomes and protein sequences. *Nucleic Acids Res*. 30(1): 31-4, 2002.
29. Mitchell, T. M. *Machine Learning*. McGraw-Hill, New York, 1997.

30. Murzin, A. G., Brenner, S. E., Hubbard, T. and Chothia, C. SCOP: a structural classification of proteins database for the investigation of sequences and structures. *J Mol Biol.* 247(4): 536-40, 1995.
31. Orengo, C. A., Michie, A. D., Jones, S., Jones, D. T., Swindells, M. B. and Thornton, J. M. CATH--a hierarchic classification of protein domain structures. *Structure.* 5(8): 1093-108, 1997.
32. Quackenbush, J. Computational analysis of microarray data. *Nat Rev Genet.* 2(6): 418-27, 2001.
33. Quackenbush, J. Microarray data normalization and transformation. *Nat Genet.* 32 Suppl: 496-501, 2002.
34. Russell, S. and Norvig, P. *Artificial Intelligence.* Prentice-Hall, New Jersey, 1995.
35. Schapire, R. E. The strength of weak learnability. *Machine learning.* 5: 197-227, 1990.
36. Schena, M., Shalon, D., Davis, R. W. and Brown, P. O. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science.* 270(5235): 467-70, 1995.
37. Shatkay, H. and Feldman, R. Mining the biomedical literature in the genomic era: an overview. *J Comput Biol.* 10(6): 821-55, 2003.
38. Sherlock, G. Analysis of large-scale gene expression data. *Curr Opin Immunol.* 12(2): 201-5, 2000.
39. Smyth, M. S. and Martin, J. H. x ray crystallography. *Mol Pathol.* 53(1): 8-14, 2000.
40. Tateno, Y., Imanishi, T., Miyazaki, S., Fukami-Kobayashi, K., Saitou, N., Sugawara, H. and Gojobori, T. DNA Data Bank of Japan (DDBJ) for genome scale research in life science. *Nucleic Acids Res.* 30(1): 27-30, 2002.
41. Theodoridis, S. and Koutroumbas, K. *Pattern recognition.* Academic Press, Amsterdam ; Boston, 2003.
42. Wheeler, D. L., Church, D. M., Edgar, R., Federhen, S., Helmberg, W., Madden, T. L., Pontius, J. U., Schuler, G. D., Schriml, L. M., Sequeira, E.*, et al.* Database resources of the National Center for Biotechnology Information: update. *Nucleic Acids Res.* 32 Database issue: D35-40, 2004.
43. Wuthrich, K. Determination of three-dimensional protein structures in solution by nuclear magnetic resonance: an overview. *Methods Enzymol.* 177: 125-31, 1989.