Bioinformatics: Organisms from Venus, Technology from Jupiter, Algorithms from Mars [1]

Bart De Moor, Kathleen Marchal, Janick Mathys, Yves Moreau [2]

ESAT-SCD
Katholieke Universiteit Leuven, Kasteelpark Arenberg 10, B-3000 Leuven, Belgium
T: +32-(0)16-321709 F: +32-(0)16-321970 M: +32-(0)475-287052
E: bart.demoor@esat.kuleuven.ac.be
W: www.esat.kuleuven.ac.be/~demoor (personal)
www.esat.kuleuven.ac.be/sista (general research)
www.esat.kuleuven.ac.be/~sistawww/cgi-bin/pub.pl (publication engine)
www.esat.kuleuven.ac.be/~dna/BioI/ (bioinformatics research)

Abstract

In this paper, we discuss datasets that are being generated by microarray technology, which makes it possible to measure in parallel the activity or expression of thousands of genes simultaneously. We discuss the basics of the technology, how to preprocess the data, and how classical and newly developed algorithms can be used to generate insight in the biological processes that have generated the data. Algorithms we discuss are Principal Component Analysis, clustering techniques such as hierarchical clustering and Adaptive Quality Based Clustering and statistical sampling methods, such as Monte Carlo Markov Chains and Gibbs sampling. We illustrate these algorithms with several real-life cases from diagnostics and class discovery in leukemia, functional genomics research on the mitotic cell cycle of yeast, and motif detection in Arabidopsis thaliana using DNA background models. We also discuss some bioinformatics software platforms. In the final part of the manuscript, we present some future perspectives on the development of bioinformatics, including some visionary discussions on technology, algorithms, systems biology and computational biomedicine.

Keywords: Bayesian networks, biclustering, bioinformatics, clustering techniques, computational biology, datamining, DNA chips, dynamical systems, genetic networks, Gibbs sampling, graphical models, information retrieval, metabolome/metabolomics, microarrays, motif detection, ontologies, proteome/proteomics, singular value decomposition, support vector machines, systems biology, text mining, transcriptome/ transciptomics,

[1] Research supported by ***Research Council KUL*****:** GOA-Mefisto 666, IDO (IOTA Oncology, Genetic networks), several PhD/postdoc & fellow grants; ***Flemish Government*****: FWO:** PhD/postdoc grants, projects G.0115.01 (microarrays/oncology), G.0240.99 (multilinear algebra), G.0407.02 (support vector machines), G.0413.03 (inference in bioi), G.0388.03 (microarrays for clinical use), G.0229.03 (ontologies in bioi), research communities (ICCoS, ANMMM); **AWI:** Bil. Int. Collaboration Hungary/ Poland; **IWT:** PhD Grants, STWW-Genprom (gene promotor prediction), GBOU-McKnow (Knowledge management algorithms), GBOU-SQUAD (quorum sensing), GBOU-ANA (biosensors); ***Belgian Federal Government*****:** DWTC (IUAP IV-02 (1996-2001) and IUAP V-22 (2002-2006); **EU: RTD:** CAGE (Compendium of Arabidopsis Gene Expression); ERNSI: European Research Network on System Identification; ***Contract Research/agreements*****:** Data4s, VIB;

[2] BDM is a full professor, KM and YM are postdoc researchers with the Fund of Scientific Research (Flanders) and assistant professor, JM is a postdoc researcher, all at the K.U.Leuven; with the help of and deep gratitude to my interdisciplinary friends, colleagues and PhD students: the control engineers Tijl Debie, Lieven De Lathauwer, Johan Suykens, Gert Thijs, Dries Van Dromme, Tony Van Gestel and Frank De Smet (who is also a medical doctor), the bio-engineers Stein Aerts, Joke Allemeers, Bert Coessens, Steffen Durinck, Kristof Engelen, Pieter Monsieurs, Qizheng Sheng, Ruth Van Hellemont, the AI masters Peter Antal, Patrick Glenisson, Bart Hamers, Frizo Janssens, Kristiaan Pelckmans, Nathalie Pochet, the mathematics master Geert Fannes and the statistician Jos Debrabanter.

Table of contents

1. Introduction

'In vivo veritas'

This year 2003 marks the 50th anniversary of the discovery of the double helix structure of DNA, the basic building structure of all living organisms, by Crick and Watson, in the 1 page landmark paper [Watson, 1953][3]. Since then, over the past 50 years, the evolution of biotechnology has been remarkable and expontential, not in the least because of the recent merger of biology with advanced computation, into what is nowadays called *bioinformatics.* Computer science and mathematical engineering on the one hand, and biology on the other, are, at first blush, an unlikely pairing: abstract, symbolic-numeric and/or mathematical computation, versus wet, evolving living organisms. But in the near future, the relationship between biology and computer science and mathematics will be seen to be as deep and abiding as the relationship between mathematics and physics (see e.g. [Lesk, 2000]). Remarkably enough, in the history of science, there have been many, 'almost accidental', encounters between biology and mathematics: The classic studies of inheritance, reported in an 1866 paper, by Gregor Mendel's [Henig, 2000], were an exercise, not in biology, but in statistical inference. They laid the foundation of contemporary genetics. Other lesser known examples are Shannon, with his 1940 PhD Thesis entitled '*An Algebra for Theoretical Genetics'* or, Alan Turing, who in the fifties described the morphogenesis of embryos in their early stage using convection-diffusion equations.

While the main objective of this paper is to elucidate the way in which biotechnology has boomed because of mathematics and information technology, we, as systems and control engineers, might also learn from biological systems, which themselves are highly evolved, extremely robust and amazingly effective information processing systems. Biological systems continue to produce potent methapors for intelligent systems: Artificial neural networks (modeled on biological neurons) have become highly competitive machine-learning tools. Genetic algorithms mimic the Darwinian optimization program of natural selection ('survival-of-the-fittest'). Artificial immune systems have been devised to detect computer viruses. DNA computing experiments (i.e. calculating by exploiting the complementarity properties of DNA molecules, using chemical concentrations as state variables) have solved NP-hard problems in linear time, and under certain (plausible) conditions can be shown to be Turing complete (see e.g. [Kari, 1997]).

Biology itself is undergoing a revolutionary change that would be impossible without advanced computation. New data generation technologies have brought a 'high-throughput' era to biology. DNA sequencing technologies were the first to produce large amounts of data. In the last 10 years or so, many genomes have been sequenced, such as (non-exhaustively) several tens of viruses [4], unicellular organisms including bacteria (e.g. *Haemophilus influenzae*), yeast (*Saccharomyces cerevisae*), plants such as *Arabidopsis thaliana* (Nature, 14 December 2000), *rice*[5], the nematode worm *Caenorhabditis elegans*[6], the fruitfly *Drosophila melanogaster* [Science, March 24, 2000], the mouse *Mus musculus* (only the second mammalian sequenced to date, see Nature, 420, December 5, 2002) . Most spectacular of all is of course the Humane Genome Project, in which two teams managed to sequence the complete

[3] In Nature of January 23, vol.421, 2003, there is a special section (pp.395-453) commemorating this 50th anniversary. It includes some very interesting state-of-the-art survey papers as well as some historical reprints.

[4] The genetic code of the SARS (Severe Acute Respiratory Syndrome) virus for instance, was cracked in a record amount of 3 weeks in April 2003, and can be found at http://www.bcgsc.ca/bioinfo/SARS (a slightly different version, because based on another sample, can be found at www.cdc.gov). The virus has about 29 700 nucleotides. The knowledge of its genetic code may help in investigating which proteins it can generate and might also lead to refined diagnostic tests.

[5] The genome maps of two subspecies of rice were published in Science, April 5, 2002. They pave the way for breakthroughs in framing humankind's most important food staples, for instance by developing better strains of rice and benefits for other crops, including wheat, corn, oats, sorghum and barley (70 % of world's agricultural acres are planted in rice, wheat and corn!). The sequencing was achieved in just 74 days (working around the clock), by a method called whole-genome shotgun technique, in which scientists break up the genome, sequence the overlapping pieces simultaneously, and then use advanced computing to arrange the segments as they exist on the chromosomes. It is estimated that rice contains between 32 000 and 50 000 genes, and it is expected that each rice gene creates only one protein, whereas a single human gene usually spawns several.

[6] This little worm, which only has 959 cells, is an example of what is called in biotechnology, a 'model organism', as its genes can be used to analyse corresponding genes and their functionalities in humans. A deeper understanding of the genetic processes that govern its organ development and cell death earned three scientists the 2002 Nobel prize in Medicine. Mice are also often used as model organisms, e.g. by knocking out certain genes and then observe the induced developments or by inserting genes so that they can develop certain human diseases (such as Alzheimer's) so that the effect of drugs can be tested.

human genome ([Lander, 2001], [Venter, 2001] [7]. Typically, the number of genes in each of these genomes at this moment is unknown, albeit that estimates are available (e.g. 31000 to 35000 for humans). The number of genes analysed to date ranges from a few hundred for bacteria to tens of thousands for mammalian species. The number of products encoded by these genes (e.g. proteins) is much higher.

Bioinformatics originates in the collision of two historical trends: The exponential increase of computing power (as expressed by Moore's law), and the exponential increase of biological and biomolecular data. Indeed, following the sequencing methodologies, many more high-throughput data acquisition methods have been and are being developed, including DNA microarrays (see Section 3 of this paper), protein measuring devices based on 2D-electrophoresis and other technologies, identifying the compounds present in a mixture of biological molecules using mass spectroscopy, determining the 3D structure of proteins (using x-ray crystallography and nuclear magnetic resonance (NMR) spectroscopy), etc… As a result, genome sequence information is doubling in size every 18 months (which coincidentally happens to be the time constant in Moore's law too). Some experts predict a production of 100 GB of biological information worldwide *per day* !

These biological data have certain key features (we cite here from [Altman, 2001]):

Biodata feature 1: Biological data is normally collected with a relatively low signal-to-noise ratio. This creates a need for robust analysis methods.

Biodata feature 2: Biology's theoretical basis is still in its infancy, so few 'first principle' approaches have any chance of working yet. This creates a need for statistical and probabilistic models.

Biodata feature 3: Despite the wealth of biological data, biology is still relatively knowledge rich and data poor. We know more about biology in a qualitative sense than a quantitative one. This creates a need for complex knowledge representations.

Biodata feature 4: Biology (and its associated data sources) operate at multiple scales that are tightly linked. This creates a need for cross-scale data integration methods.

Biodata feature 5: Biological research efforts are distributed, and the associated databases focus on particular types of data. This creates a need for data integration methods.

Biodata feature 6: Biologists think graphically about their work. This creates a need for user interfaces and graphical metaphors for communicating information.

Table 1: Biological data have some characteristic features that have to be taken into account when developing algorithms and software tools.

In the different subsections of this paper, we will refer to specific *biodata features* that are relevant for the problem discussed in that particular subsection.

The increasing availability of these data, many – if not all - of which can be found in databases on the web, has started to attract a lot of system theory and control engineers, statisticians and mathematicias to biology. It is however predicted that an alarming shortness of bioinformaticians will occur in the near future, indicating the need for training programs and representing intruiging challenges for experts in systems theory and identification, dynamical systems and control theory, who are looking for nice applications (and development of new theories). People working in 'our' community, or in more general terms, in mathematical engineering, increasingly get involved in bioinformatics, as is witnessed by some special issues of journals in 'our' domain, like

- The special issue of ERCIM News (European Consortium for Informatics and Mathematics, www.ercim.org) of October 2000;
- The November 2000 issue of IEEE Spectrum on 'Gene sequencing's industrial revolution';
- The December 2000 issue of the Proceedings of the IEEE on 'Genomic Engineering: Moving beyond DNA Sequence to Function;
- An intriguing article on 'Genomic signal processing' in IEEE Signal Processing Magazine [Anastassiou, 2001];

[7] A supercomputer was used, consisting of a cluster of 800 processors with 70 terabytes of storage.

- the special issue of the ‘IEEE Transactions on Intelligent Systems’ on ‘Intelligent Systems in Biology’ of March/April 2002;
- the special issue on ‘Bioinformatics’ of the IEEE Journal ‘Computer’ of July 2002 (Vol.35, no.7);
- the two special issues on ‘Bioinformatics’ of the IEEE Proceedings, of November 2002 (Vol.90, no.11) and December 2002 (Vol.90, no.12);
- the special issue on Genomic Signal Processing of the journal Signal Processing, April 2003;

Of course, it is merely impossible in this survey paper to provide the reader with a complete, let alone exhaustive overview of what bio-informatics is about. But these forementioned issues are a good start to get acquainted with the challenges. Early books on bioinformatics include [Bishop, 1997] [Baldi, 1998]. More recent ones include [Mount, 2001] [Ewens, 2001]. Nice reading is also provided by collections of papers like for instance the ones on Functional Genomics (Nature Insight, Nature, Vol.405, no.6788, June 15 2000, pp.819-865). Besides these books and articles, in the list of references at the end, we have included two types of papers: Some key *scientific* references, needed when discussing some of our results on the one hand (they will be referred to in the body of the text), and some more *popular* accounts about the state-of-the-art in genomics, biology and bioinformatics on the other hand, such as [Davies, 2001] [Ezzel, 2002] [Friend, 2002] [Henig, 2000] [Lesk, 2000] [Ridley, 1999] [Sykes, 2002] [8].

This paper presents a view on bioinformatics that is (quite understandably and hopefully forgivable) subjective and heavily biased by our own research, the details of which can be found in papers listed in the references, most of which can be downloaded from our website mentioned above. Two of our own survey papers are [Moreau, 2002a] [Moreau, 2002b].

This paper is organized as follows:

- In Section 2, we review some necessary basic biological facts that are needed for a further understanding of this paper (It is of course infeasible to do justice to the current state of knowledge in biology within the context of this paper and we realize that this Section, to a trained biologist or physician, is hopelessly naïve);
- In Section 3, we discuss microarray technology, which allows to unravel many genetic mechanisms by observing thousands of gene expression levels at once, and will play a very important role in both scientific and clinical applications in the near future. We will also elaborate on the indispensable sequence of steps required to store and preprocess microarray data;
- In Section 4, we will concentrate on the state-of-the-art in bioinformatics algorithms, giving a (biased) survey of basic tools from linear algebra and statistics, advanced clustering methods and statistical Gibbs sampling based algorithms, including a recently developed biclustering algorithm;
- In Section 5, we will present several cases that demonstrate the central role mathematical engineering methodologies play in modern biotechnology, ranging from
    - Performance assessment of clinical classiciation and prediction methodologies in diagnosis, prognosis and therapy response of leukemia (Subsection 5.1 and 5.2);
    - Discovery of relevant genes or groups of genes in yeast cell cycles (Subsection 5.3.);
    - Motif detection in Arababidopsis thaliana (Subsection 5.4.);
- Finally, in Section 6, we present some visionary views on future developments in technology (we survey several ‘-omes’ and ‘-omics’), algorithms (Bayesian networks, support vector machines), systems biology and computational biomedicine.

The type of bioinformatics we will mainly be dealing with in this paper is largely concentrated on *transcriptomics*, analyzing data that originate from so-called microarrays, that allow to obtain gene expression levels from thousands of genes simultaneously. Of course, the whole field of bioinformatics will grow much larger than transcriptomics alone, and we will elaborate on these future perspectives in Section 6. We will show how the road towards a deeper understanding of biological processes of life, disease and death, lies wide open. Brand new hardware and information technologies have raised grand expectations for biology and medicine, where they will be instrumental in unraveling the molecular and cellular mechanisms of acquired or inherited diseases, lead to the development of new diagnostic methods, prevention methodologies, therapeutics or successful treatment.

[8] That bioinformatics is a rapidly evolving discipline, is also witnessed by some (only indicative) statistics of the reference list of this paper, 9.5 % of which is from 2003, 34.5 % from 2002, 24 % from 2001, 7 % from 2000 and another 25 % from 1999 and earlier.

2. Organisms from Venus: (Some) biology

Biodata feature 3

It has not escaped our notice that the specific pairing we have postulated immediately suggests a possible copying mechanism for the genetic material
Crick & Watson in their 1953 Nature paper

In this Section, we briefly discuss DNA, which contains the code and structure of living organisms, RNA, which acts a messenger and proteins, which can be seen as effectors. Obviously, we can only provide a very crude introduction to biology, and as a compensation, we refer to some very nice books, such as [Griffiths, 1996] [Kreuzer, 1996] [Griffiths, 1999] [Brown, 2002] [Karp, 2002] for more detailed, extensive and didactical expositions.

The human body is made up of trillions of cells, the nucleus of which contains an identical complement of chromosomes. Each chromosome is one long DNA molecule, and genes are functional regions of this DNA [9]. DNA stands for DeoxyriboNucleic Acid. The genome of every living organism, its genetic sequence, consists of genetic building blocks, called nucleotide bases, that make up its DNA. There are 4 of them, called A (adenine), C (cytosine), T (thymine) and G (guanine). The humane genome contains approximately 3 billion DNA bases, and an (almost complete) draft of its sequence is now available [Lander, 2002] [Venter, 2002]. The geometric structure of DNA is the famous double helix discovered by Crick and Watson [Watson, 1953] [10]: The structure looks like a spiral staircase, in which the backbone of each strand is a repeating phosphate-dexoyribose sugar polymer and the stairs are formed by (always) complementary pairs of A-T and C-G. Genes are segments of DNA that encode the structure of some cellular product, but also bear control buttons that determine when, where and how much of that product is synthesized. Most genes encode for proteins through the intermediate action of messenger RNA (RiboNucleic Acid, mRNA). As the genome of many organisms has been sequenced, estimates of the number of genes become more and more reliable. Some examples are: Bacteriophage lambda (genome size 5.0E+04 base pairs, 60 genes), Escherichia coli (4.6E+06 bp, 4290 genes), Yeast (12.0E+06 bp, 6144 genes), Drosophila melanogaster (1.0E+08 bp, 13338 genes), Caenorhabditis elegans (1.0E+08 bp, 18266 genes), Arabidopsis thaliana (2.3E+08 bp, 25000 genes), Homo sapiens (3.0E+09 bp, 32000 genes).

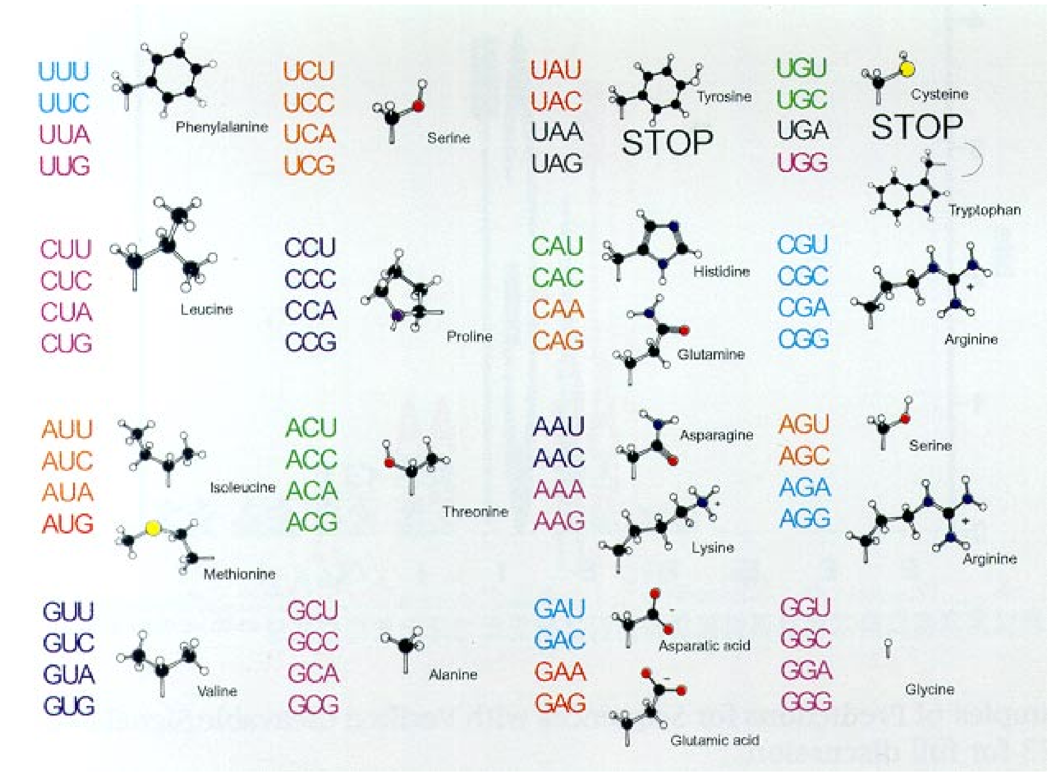
RNA has a number of biological functions (informational RNAs, Functional RNAs (Transfer RNA, Ribosomal RNA,…),..), but one of its primary function is to be the working copy of the gene (a copy made directly from the DNA) that is then used to synthesize proteins. The first step in the way genes encode for proteins is to copy (*transcribe, transcription*) the information encoded in the DNA of the gene as a related, but single-stranded molecule called messenger RNA (mRNA) (In RNA, the 'T' is replaced by Uracil, denoted by 'U'). The gene and the genomic region surrounding it consist of a transcribed sequence, which is converted into an mRNA transcript, and of various untranscribed sequences, called untranslated regions (UTRs). These UTRs play a major role in the regulation of expression. Notably, the promoter region in front of the transcribed sequence contains the binding sites for the transcription factor proteins that start up transcription. The transcription process is initiated by the binding of several transcription factors to regulatory sites in the DNA, usually located in the promoter region of the gene. The transcription factor proteins bind each other to form a complex that associates with an enzyme called RNA polymerase. This association enables the binding of RNA polymerase to a specific site in the promoter. Together, the complex of transcription factors and the RNA polymerase unravel the DNA and separate both strands. Subsequently, the polymerase proceeds down on one strand while it builds up a strand of mRNA complementary to the DNA, until it reaches the terminator sequence. In this way, an mRNA is produced that is complementary to the transcribed part of the gene. Then, the mRNA transcript detaches from the RNA polymerase and the polymerase breaks its contact with the DNA. In a later stage, the mRNA is processed, transported out of the

[9] At the time of writing of this article, April 2003, a team of 90 scientists from 10 countries has just completely finished chromosome 7 of the human genome, which contains 158 million nucleotides (see Science, April 11, 2003 and www.chr7.org) . 1455 genes have been identified, some of which are responsible for genetic diseases such as mucoviscidose, leukemia and autism. So far, only chromosomes 14 (Nature Feb. 6, 2003), 20, 21 and 22 had been fully completed ('fully' means that 99.99 % of the 'letters' are correct).

[10] Event though throughout history, DNA research has resulted in 9 Nobel Prizes, as of today there is still a lot of research activity on the properties of DNA (see the special issue of the New Scientist of March 15, 2003: DNA: The next fifty years).

nucleus, and translated into a protein. Moreover, the region upstream of the transcription start contains many binding sites for transcription factors that act as enhancers and repressors of gene expression (although some transcription factors can bind outside this region). Transcription factors are proteins that bind to regulatory sequences on eukaryotic chromosomes thereby modifying the rate of transcription of a gene. Some transcription factors bind directly to specific sequences in the DNA (promoters, enhancers, and repressors), others bind to each other. Most of them bind both to the DNA as well as to other transcription factors. It should be noted that the transcription rate can be positively or negatively affected by the action of transcription factors. When the transcription factor significantly decreases the transcription of a gene, it is called a repressor. If, on the other hand, the expression of a gene is upregulated, biologists speak of an enhancer.

The expressed mRNA is brought to the ribosome, which can be considered as a protein factory. In a ribosome, the genetic code is used to read off the sequence of amino acids that will create a protein. This step is called *translation*. A protein is a polymer composed of monomers called amino acids. The amino acid sequence is determined by the nucleotide sequence of the gene that encodes for it. As a ribosome moves along the mRNA, it reads three nucleotides at a time, called a triplet codon. Since there are 4 different nucleotides (A-C-U-G), there are 4x4x4=64 different possible codons. There are 20 amino acids, each of them encoded by a codon. This genetic code, which is summarized in Table 1, is used by virtually all organisms on the planet.



© Alberts B. et al. Essential cell biology. Garland Pub. 1997

Figure 1. In Eukaryotes, AUG is generally the first codon (start). Some codons in the Table do not specify an amino acid at all, such as UAA, UAG and UGA. They are called stop or termination codons, that can be regarded as punctuation marks ending the message encoded in the mRNA. Often, potential genes are identified by looking for open reading frames (ORFs), which are DNA sequences that start with an initiation codon ATG and end in one of the three stop codons.

Each string of amino acids then folds into a 3D protein structure. The determination of the exact 3D structure of a protein starting from its 1D amino acid sequence, which is basic for understanding its functionality, is still quite a challenge (protein folding, see also [Altman, 2001] for computational challenges). Protein architecture is the key to gene function. The specific amino acid sequence determines the general shape, binding properties and reactivity of proteins, that can be responsible for enzymatic catalysis, structural support, motion, signal transduction of physical signals (e.g. light), cell-to-cell communication, and many other functions. Proteins are the most important determinants of the properties of cells and organisms. Any failure in the genetic procedure above, can result in genetic deficiencies and diseases [11].

3. Technology from Jupiter: Transcriptomics and micro-arrays

Biodate feature 1

Microarray technology is one of the most promising technologies recently developed in molecular biology [Schena, 1995] [DeRisi, 1997] [Lander 1999]. Microarrays make it possible to measure in parallel the activity or expression (transcription or amount of mRNA produced for a specific gene) of thousands of genes in a certain tissue (e.g. in a tumour), measurements that can be repeated under several different conditions (e.g. normal versus malignant tissues, tumours that are or are not sensitive to chemotherapy, tumours with or without metastatic potential, etc…) or over several tens to hundreds of patients. The hardware will be explained in Subsection 3.1., while in Subsection 3.2, we briefly discuss the required Laboratory Information Management Systems set up and the necessary preprocessing of the data is the subject of Subsection 3.3.

3.1. Microarrays

Microarrays in essence exploit the complementarity of DNA as evidenced in the double helix structure. As we have seen in Section 2, cells produce the proteins they need to function properly by (1) transcribing the corresponding genes from DNA into messenger RNA (mRNA) transcripts and (2) translating the mRNA molecules into proteins. Microarrays obtain a snapshot of the activity of a cell by deriving a measurement from the number of copies of each type of mRNA molecule (which also gives an indirect and imperfect picture of the protein activity). The key to this measurement is the double-helix hybridization properties of DNA (and RNA): When a single strand of DNA is brought in contact with a complementary DNA sequence, it will anneal to this complementary sequence to form double-stranded DNA. For the four DNA bases, Adenine is complementary to Cytosine and Guanine is complementary to Thymine. Because both strands have opposite orientations, the complementary sequence is produced by complementing the bases of the reference sequence starting from the end of this sequence and proceeding further upstream. Hybridization will therefore allow a DNA probe to recognize a copy of its complementary sequence obtained from a biological sample. An array consists of a reproducible pattern of different DNA probes attached to a solid support, as a lawn of single-stranded DNA molecules that are tethered to a wafer often not bigger than a thumbprint. After RNA extraction from a biological sample, fluorescently labeled complementary DNA (cDNA) or cRNA is prepared. This fluorescent sample is then hybridized to the DNA present on the array. Each kind of probe – be it a gene or a shorter sequence of genetic code – sits in an assigned spot within a checkerboardlike grid on the chip. The DNA or RNA molecules that get poured over the array carry a fluorescent tag that can be detected by a scanner. Thanks to the fluorescence, hybridization intensities (which are related to the number of copies of each RNA species present in the sample) can be measured by a laser scanner and converted to a quantitative readout.

DNA microarrays are used for genotype applications (in which the DNA on a chip is compared to the DNA in a tissue sample to determine which genes are in the sample or to decipher the order of the code letters in as yet unsequenced strings of DNA). But increasingly more often these days, the microarrays

[11] As an example, consider Huntington's disease, cause by the gene huntingtin, that lies at the tip of chromosome 4 and was identified in 1993 [Cattaneo, 2002] and which is one of a number of inherited neurodegenerative disorders characterized by the presence of CAG repeat coding for an expanded polyglutamine domain. Normally the gene contains between 9 and 35 repeats of the DNA sequence CAG, that encodes for the amino acid glutamine. But in families with Huntington's, the gene usually has between 40 to 60 repeats. When transcribed into messenger RNA, which directs the cell's protein-making machinery (transfer RNA and ribosomes), mutant huntingtin contains a large polyglutamine region, that probably causes the disease by either disabling huntingtin protein or by allowing to stick to and inactivate normal huntingtin protein or other proteins, or by a combination of these mechanisms. The abnormal proteins form insoluble protein aggregates, accompanied by neural dysfunction and cell loss. Such protein aggregates appear to be toxic to brain cells.

are used to assess the activitity level, called ‘expression’, of genes. A gene is said to be expressed when it is transcribed into messenger RNA (mRNA) and translated into protein. Generally, the more copies of mRNA a cell makes, the more copies of protein it will make, so, in this sense, the quantities of the various mRNAs in a sample can indirectly indicate the types and amounts of proteins present. Proteins in a certain sense are more interesting than DNA, because they control and carry out most activities in our bodies’ cells and tissues. Through ‘guilt-by-association’, genes of unknown functions can be assigned a putative function by linking them to genes with similar patterns of expression whose function is already known. In many organisms, genes for which nothing is known about the function, still represent 30% of the genome and for many more genes the information available is fragmentary at best. Microarrays therefore provide a powerful approach to this extremely pressing question. Further, they are invaluable for unraveling the networks of regulation that control the dynamic behavior of genes. Understanding the network of interaction between genes is the central goal of genomics and we will come back to all of this further down in this paper. A picture of the robotic set up of typical microarray hardware, and an example of a resulting image, is shown in Figure 2.

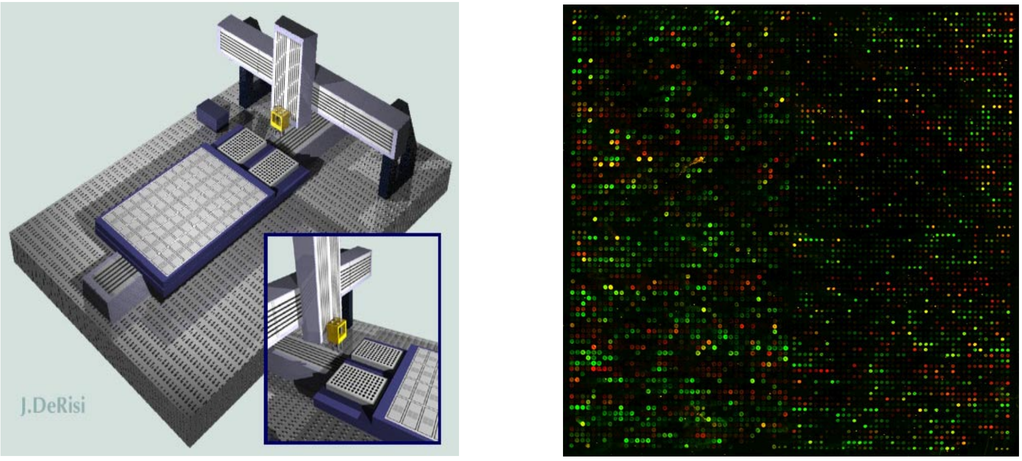


Figure 2: Left: Microarray spotter (from [DeRisi, 1007]) ; Right: Typical microarray image

An array consists of a reproducible pattern of different DNAs (primarily PCR products or oligonucleotides – also called probes) attached to a solid support. Fluorescently labeled cDNA, prepared from mRNA, is hybridized to the complementary DNA present on the array. cDNA-DNA hybridization intensities are measured by a laser scanner and converted to a quantitative read out. This data can be further analyzed by data-mining techniques (see below).

Two basic types of arrays are currently available:

1). Spotted arrays (Duggan et al., 1999) or cDNA-microarrays are small glass slides on which pre-synthesized single stranded DNA or double-stranded DNA is spotted. These DNA fragments are usually several hundred base pairs in length and are derived from ESTs (Expressed Sequence Tag, which are subsequences from an mRNA transcript that uniquely identify this transcript) or known coding sequences from the organism studied. Usually each spot represents one single ORF (Open Reading Frame) or gene. A pair of cDNA samples is independently copied from the corresponding mRNA populations (usually derived from a reference and a test sample (e.g., normal versus malignant tissue)) with reverse transcriptase and labeled using distinct fluorochromes (green and red). These labeled cDNA samples are subsequently pooled and hybridized to the array. Relative amounts of a particular gene transcript in the two samples are determined by measuring the signal intensities detected for both fluorochromes and calculating the ratios (here, only relative expression levels are obtained). A cDNA microarray is therefore a differential technique, which intrinsically (at least partially) normalizes for noise and background. An overview of the procedure that can be followed with spotted arrays is given in Figure 3.

2). GeneChip oligonucleotide arrays (Affymetrix, Inc., Santa Clara, CA) (Lipshutz et al., 1999) are high-density arrays of oligonucleotides synthesized *in situ* using light-directed chemistry (photolithographic technology similar to chip technology) consisting of thousands of different oligomer probes (25-mers). Each gene is represented by 15-20 different oligonucleotides, serving as unique sequence-specific detectors. In addition mismatch control oligonucleotides (identical to the perfect match probes except for a single base-pair mismatch) are added. These control probes allow for estimation of cross-hybridization and significantly reduce the number of false positives. With this technology, absolute expression levels are obtained (no ratios).

For a popular account on microarrays, we refer to [Friend, 2002] or also to the animated technology demonstration that can be found at http://www.bio.davidson.edu/courses/genomics/chip/chip.html. Some publicly available microarray datasets include one on leukemia ([Golub, 1999], also discussed in Section 5 below), breast cancer [Van ‘t Veer, 2000], Colon Tumor [Alon, 1999], mitotic cell cycle in yeast [Cho, 1998] (also discussed below). DNA microarrays, first introduced commercially in 1996, are now mainstays of scientific research, drug discovery, medical diagnosis and prognosis, etc. There are several companies producing these arrays and the whole sector is in a permanently ongoing technological evolution. More and more research institutions and companies have their own microarray facilities (see e.g. the one of the Flemish Biotech Institute at www.vib.be/maf).

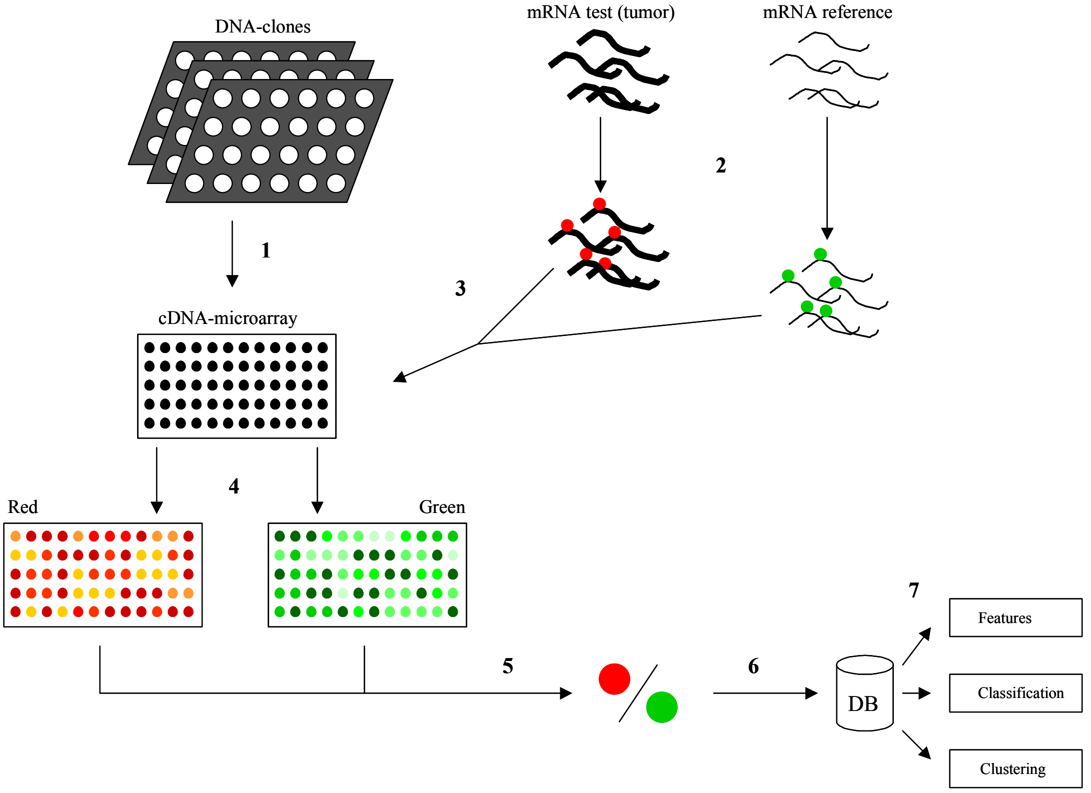


Figure 3: Schematic overview of an experiment with a cDNA-microarray. (1) Spotting of the pre-synthesized DNA-probes (derived from the genes to be studied) on the glass slide. These probes are the purified products from PCR-amplification of the associated DNA-clones. (2) Labeling (via reverse transcriptase) of the total mRNA of the test sample (tumor - red) and reference sample (green). (3) Pooling of the two samples and hybridization (4) Read-out of the red and green intensities separately (measure for the hybridization by the test and reference sample) in each probe. (5) Calculation of the relative expression levels (intensity in the red channel / intensity in the green channel). (6) Storage of results in a database. (7) Data mining and algorithms.

The technology of microarrays is changing so rapidly and has become so important that dedicated organizations have been created to coordinate its development, such as the Microarray Gene Expression Data Society (MGED, see www.mged.org, and their international meeting (see e.g. http://tagc.univ-mrs.fr/mged6/). Also, in order to standardize the way microarray experiments should be performed, some *universal* rules were defined on how to annotate every microarray based experiment to allow unambiguous interpretation of its results (MIAME: Minimum Information About a Microarray Experiment, [Brazma, 2001]). MIAME-compliant microarray experiment annotation can be done by simply following the MIAME checklist and web-based forms., and has been adopted by many scientific journals (including e.g. Nature[12], the Lancet,…) as a requirement for microarray based publications.

3.2. MAF-LIMS: Microarray-facility Laboratory Information Systems

Biodata feature 5

The production of a microarray is a complex procedure that is inevitably error prone. This necessitates the backtracking on several experimental settings or hypotheses. A recent study for example report error rates over 35% of the data points due to lack of consistency checks and flaws in annotation [Knight, 2001]. Although it is impossible to render a system foolproof, guaranteeing an acceptable quality level and reproducibility is possible through meticulous recording of the various steps in the data generation process using a Laboratory Information Management Systems (LIMS). The use of standards to this end can enhance and prolong the life cycle of a microarray experiment.

The following steps involved in the production of a cDNA microarray illustrate the many points, which are noise-sensitive or error-prone.

- Purchase or generation of proprietary 'clone' library – a collection of genetic fragments ideally representing a set of genes of a given organism. These clones are multiplied, stored and ordered into so called well-plates, usually per 96 or 384.
- Contamination (leaking of DNA fragments to neighboring wells) and equipment constraints usually require a reordering of the clones usually performed by pipetting robots.
- In the final preparation another robot 'prints' the genetic material to the glass slide so that each spot represents a single open reading frame or a gene. The biochemistry of this printing (or 'spotting') is complex and the configurations that are possible, are numerous.

These procedures lead to a spotted glass slide or microarray, which can be used to conduct an experiment. To successfully backtrack on any errors that might have occurred during this labor-intensive production process, an automated administration of each action on the workbench is necessary. LIMS serve this goal and we developed a first version for the Microarray Facility of our Biotechnology Institute (www.vib.be), visualized in Figure 4. The hybridization(s) of test and reference sample on one or several microarrays and the consecutive measurement of the relative abundance of the screened gene transcripts represent the core of a single microarray experiment. Therefore it is connected in a modular way to the LIMS information.

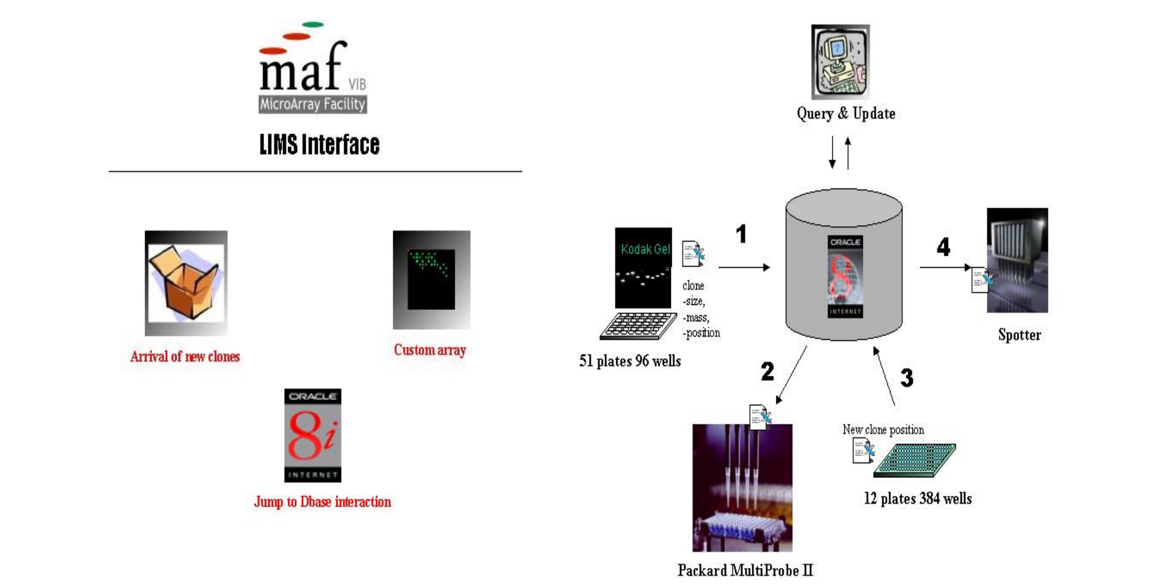


Figure 4: Screenshots of our Laboratory Information Management System: the various steps involved in a hybridization experiment are visualized for optimal tracking.

[12] See Nature, 26 September 2002, Vol.419, p.323.

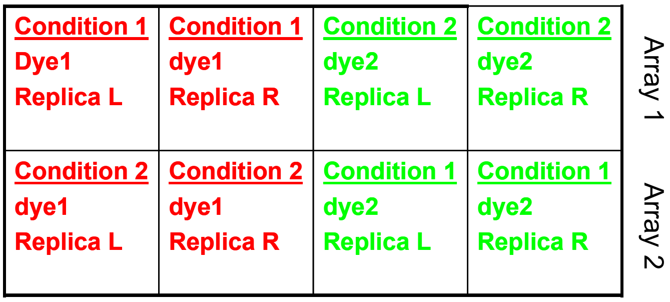
3.3. Preprocessing data

Biodata feature 1& 4

Recalling the several ‘biodata features’ enumerated in Section 1, it should come as no surprise that, with the current state of technology, observations from biological experiments are extremely noisy, with possibly outliers and missing values. Preprocessing methods are definitely needed to derive, for each gene, the intrinsic expression level as caused by the condition tested. Many methods to preprocess the data have been proposed in the literature. A thorough understanding of how different preprocessing methods transform the data and the search for a set of preprocessing techniques that make the data compatible with further analysis, is a crucial aspect of microarray analysis. We present two different approaches, known as the ratio technique, which is the ‘traditional’ one and the ANOVA technique, which is the more ‘modern’ one, to perform normalization and to detect differentially expressed genes. More details and references can be found in [Quackenbush, 2001] [Yang, 2002] and our own work [Marchal, 2002] [Moreau, 2002a] [Moreau, 2002b]. Our experience and expertise in preprocessing microarray data has been made publicly available at http://www.esat/kuleuven.ac.be/maran, which later on was also integrated in INCLUSive (which is a web portal and service registry for microarray and regulatory sequence analysis, see Section 5).

3.3.1. Sources of noise

To understand the necessity and importance of preprocessing, we need to have a clear picture of the raw data generated by a microarray experiment. As a detailed example, we describe a simple black-white experiment based on a Latin square design in Figure 5.



Condition 1 Dye1 Replica L	Condition 1 dye1 Replica R	Condition 2 dye2 Replica L	Condition 2 dye2 Replica R	Array 1
Condition 2 dye1 Replica L	Condition 2 dye1 Replica R	Condition 1 dye2 Replica L	Condition 1 dye2 Replica R	Array 2

Figure 5. Schematic representation of a Latin Square design; In such design, expression in two distinct conditions is compared (test and reference condition). On the first array, the test sample is labeled with Cy5 (red dye) while the corresponding reference is labeled with Cy3 (green dye). For each gene, two replicate spots are available on each array (referred to as left and right spot). In addition a color-flip experiment is performed: the same test and reference conditions are measured once more in duplicate on a different array where dyes have been swapped. Such a design results in four measurements per gene for each condition tested, though not directly comparable with each other.

A first set of effects that prohibits direct comparison between measurements are the “condition and dye effects”. These effects reflect differences in mRNA isolation and labeling efficiency respectively between samples of the conditions tested. For genes equally expressed in both the reference and the induced sample the ratio of test/ref is expected to be 1. Condition and dye effects result in a deviation of such ratios from 1. The mathematical transformation that tries to compensate for these effects is called *normalization*. A second source of variation is related to the imperfections of the spotting device used to produce the array. Small variations in pin geometry, target volume and target fixation cause spot dependent variations in the amount of cDNA present on the array. The observed signal intensity does not only reflect differences in mRNA population present in the sample but also the amount of spotted cDNA. Depending on the influence of the spot effects, direct comparison of the absolute expression levels may be unreliable. This problem can be alleviated by comparison of the relative expression levels (ratio of the test and reference intensities) instead of the absolute levels. Indeed reference and test have been measured on the same spot and by dividing the measured intensities, spot effects drop out. When performing multiple experiments (i.e., more arrays), arrays are not necessarily treated simultaneously. Differences in hybridization efficiency can result in global differences in intensities between slides, making measurements derived from different slides mutually incomparable.

This effect is generally called the array effect. All these effects occur simultaneously and prohibit direct comparison of expression levels.

3.3.2. Log transformation of the raw data

A log transformation of the data is the initial step in the preprocessing data analysis flow. Its necessity is explained in Figure 6 [Baldi, 2001] [Kerr , 2000]. Especially when dealing with expression ratios (coming from two-channel cDNA microarray experiments, using a test and reference sample), this transformation is suited since expression ratios are not symmetrical. Upregulated genes have expression ratios between 1 and infinity, while downregulated genes have expression ratios squashed between 1 and 0. Taking the logarithms of these expression ratios results in symmetry between expression values of up- and downregulated genes.

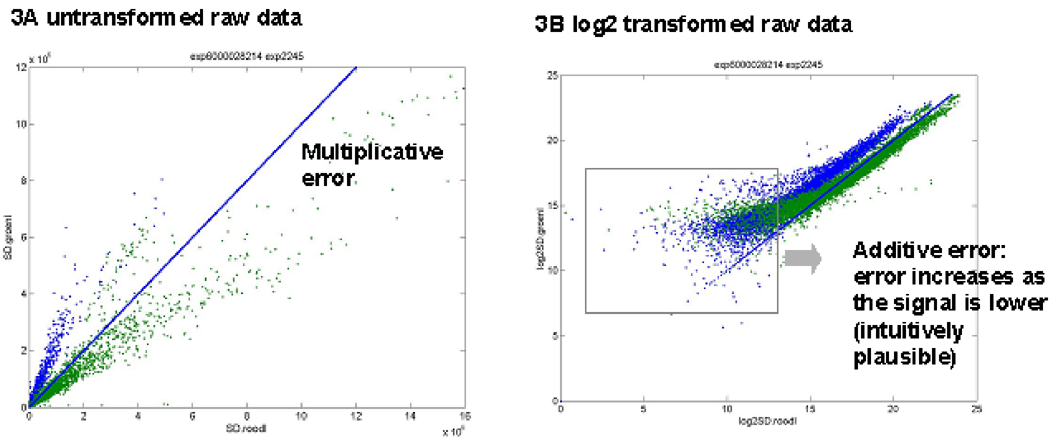


Figure 6. In the left Figure, replicate measurements (normal and color flip) of different genes are plotted against each other. The x-axis is the intensity measured in the red channel., the y-axis in the green channel. When considering untransformed raw data (background corrected intensity values), the increase of the residuals with increasing signal intensities clearly reflects the multiplicative effects. The increase of the measurement error with increasing signal intensities as present in the untransformed data is counterintuitive since high expression levels are generally considered more reliable than low levels. It is well known that multiplicative errors decrease the efficiency of most statistical tests. Therefore, it is important to get rid of multiplicative errors by log-transforming the data. In the Figure on the right, we show the influence of a log2 transformation on the multiplicative and additive errors; *x*- axis: log2 of intensity measured in red channel, y-axis: log2 of intensity measured in green channel.

3.3.3. Filtering data

As a next step in the preprocessing flow, filtering is used to remove unreliable measurements or zero values from the data set. Such filtering procedures often depend on the choice of an arbitrary threshold (e.g., all genes of which the measured expression value does not exceed twice the expression level of the background are discarded). Since in our data sets the green and red channel display different sensitivities in the low expression level range, the choice of such threshold is prone to mistakes. Therefore, if possible we try to avoid filtering procedures based on an arbitrary threshold and try to retain all genes for further analysis [Kadota, 2001]. The use of robust statistical tests allows for the discrimination between statistically over- and underexpressed genes at a later stage of the analysis. Zero values result in undefined values (e.g., when dividing by zero values or taking the log of a zero value) and therefore are automatically discarded for further analysis. However, in the light of a black & white experiment, zero values might be of major biological significance. Indeed consistent zero values correspond to genes switched off in one condition, but on in the other condition, might be very significant. Therefore if a least one measurement for a gene contains a zero value in a particular condition, all measurements of that gene are treated separately.

3.3.4. Ratio approach

Normalization is a mandatory step to remove consistent condition and dye effects. Although the use of spikes (control spots, external control) and housekeeping genes (genes expected not to alter their expression level under the conditions tested) has been described, global normalization is considered as most reliable. Global normalization assumes that only a small fraction of the total number of genes on the array alters its expression level and that symmetry exists in the number of genes that is upregulated versus downregulated. Under this assumption the average intensity of the Cy3 channel should be equal to the average Cy5 channel. Based on this hypothesis, the average ratio log2 (red/green) should be equal to 0. Regardless of the procedure used, all normalized log-ratios therefore will be centered on zero. *Linear normalization* assumes a linear relationship between red (*R*) and green (*G*) intensities. A common choice for c = $\log_2 k$ is the mean or the median of the log intensity ratios for a given gene set. Alternatively the constant normalization factor can be determined by linear regression of the red signal versus the green signal. The coefficient as identified by regression determines the rescaling factor that should be used to either divide or multiply the red signal to obtain an average signal of 0 (in log scale). As can be derived from Figure 7, the assumption of a constant rescaling factor for all intensities is an oversimplification. Indeed dye and condition effects seem to be dependent on the measured intensity. Such intensity dependent patterns are better visualized using a plot of *M* versus *A* (see Figure 7 for definitions of *M* and *A*). The relationship between the dyes is linear only in a certain range. However, when measured intensities are extreme (either high or low) nonlinear effects occur. To take into account these nonlinear effects during normalization, we prefer to use a robust scatter plot smoother Lowess that performs locally linear fits and allows calculating the normalized absolute values log2(*R*) and log2(*G*).

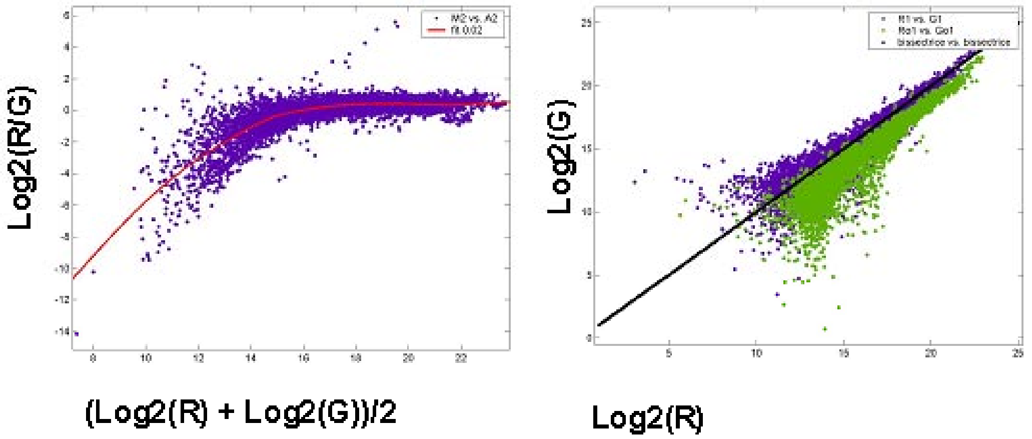


Figure 7. (A) Representation of the log intensity ratio *M* = log2 *R*/*G* versus the mean log intensity *A* = (log2(*R*)+log2(*G*))/2. At low average intensities on average the ratio becomes negative indicating that the green dye is consistently more intense than the red dye. Either the sensitivity of the red signal is lower than the one of the green signal or the basal noise level on the green signal is more pronounced. To compensate for nonlinear dye effects a Lowess fit with *f* value of 0.2 was used (solid line). (B) *x* axis log2(*R*); *y* axis: log2(*G*) Represents a plot of the log intensity of the *R* versus *G* prior to normalization by Lowess (green dots) and after Lowess normalization (blue dots). (New values of the log intensities of red and green were calculated based on the Lowess fitted values *M* and *A*.)

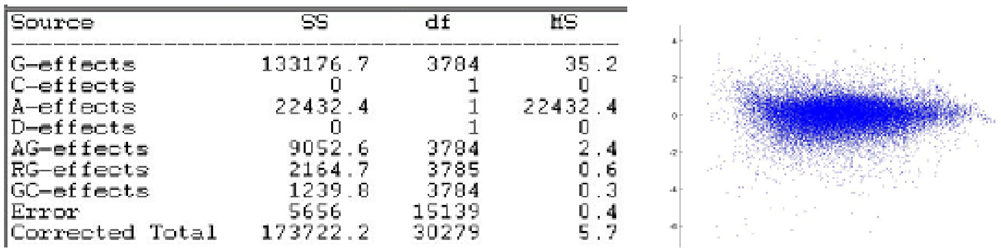
*A*fter these transformations, we can use the preprocessed values (corrected for array, spot, dye, and condition effects) to identify which genes show a differential level of expression between the two conditions by using a test statistic (T-test, paired T-test, Bayesian T-test). The drawback of most of these classical test statistics is the need for a high number of replicates. Since microarray experiments are expensive and labor intensive, the limited number of replicates usually available decreases the reliability of using classical T-tests.

3.3.5. Analysis of variance

An alternative approach that avoids the use of ratios and the need for a high number of replicates is based on analysis of variance (ANOVA), in which the measurement of the expression level of each gene is modeled as a linear combination of the major sources of variation that we have been describing. Several major effects representing the condition, dye and array effects, and combinations (2-, 3-, and 4-level combinations) of these main effects are taken into account in the models. Not all of the combined effects, however, have a physical meaning and only those considered to be important are retained in the models. Reliable use of an ANOVA model therefore requires a good insight into the process to be modeled. The model that best corresponds to physical reality is preferred. The most important combined factor in all models is the *GC* effect, the factor of interest. The *GC* effect reflects the expression of a gene depending on the condition (i.e., condition specific expression). The following equation represents the ANOVA model of microarray data taking into account the gene effect (*G*), condition effect (*C*), the dye effect (*D*), array effect (*A*), replicate effects (*R*), spot effects (*AG*):

$$I_{ijklm} = \mu + G_i + C_j + A_k + D_l + R_{m(i)} + (AG)_{ki} + (GC)_{ij} + \varepsilon_{ijklm}$$

The effect of interest (*GC*), R is 'nested' within G (indicated by brackets in the subscripts). One of the basic requirements of ANOVA is that all dependencies need to be linear. This should be reflected by a normal distribution of the residuals of the fit with zero mean and equal variance. These requirements imply that mathematical transformations (log transformation described above) are mandatory to compensate for multiplicative effects. The prerequisite of normally distributed residuals however, is not too stringent; a proper ANOVA analysis can be done when residuals are independent and identically distributed (*i.i.d.*), but not necessarily normal, with zero mean and constant variance. One of the major advantages of the ANOVA approach is that it allows gaining more information from the data than by looking at each gene separately (e.g., the array effect is similar to all genes on an array). Since all measurements are combined to allow statistical inference, the need for a high number of replicates is less pronounced. Figure 8 shows the results of an ANOVA model on Lowess normalized data (see [Yang, 2002]) that takes into account the 4 main factors (see above) and the factor of interest. It compensates for array, dye and condition effects and spot effects. To model the spots, a relationship between spots on the same array and a relationship between all left and right spots is assumed.



```
Source                    SS         df         MS
------------------------------------------------------
G-effects            133176.7      3784       35.2
C-effects                   0         1          0
A-effects             22432.4         1    22432.4
D-effects                   0         1          0
AG-effects             9052.6      3784        2.4
RG-effects             2164.7      3785        0.6
GC-effects             1239.8      3784        0.3
Error                  5656       15139        0.4
Corrected Total      173722.2     30279        5.7
```

Figure 8. Result showing the ANOVA table and corresponding residual plot of the ANOVA model on the log transformed Lowess normalized data.

An important problem with current ANOVA models for microarrays is that nonlinearities can be observed in the residual plots, which means that the basic assumptions of the model are not entirely fulfilled. Finding a proper data transformation that would render the residuals linear with respect to the estimated intensity values is the current focus of our research. A possibility here is the use of LS-SVMs (see Section 6.2. below).

Obviously, microarray experiments are not always simple black & white experiments, like the one explained above. Very often more complex designs are used to investigate biological processes of interest. This is certainly true when more than two conditions need to be compared (e.g., a time course experiment, a comparison between different mutant strains, and so on). The methodologies just described can be extended without much trouble to handle these more complex experiments. It is clear that microarray technology is very powerful but data generated by these techniques must be handled with caution. At first, there is a need for a consistent recording of the complete production procedure so that mistakes can easily be traced. Secondly, data need to be cleaned prior to further analysis. Once preprocessed, the data can be used for further data exploration and data mining as we will now explain.

4. Algorithms from Mars: How to process microarray data ?

In this Section, we will elaborate on some often used algorithms in microarray data analysis, including Principal Component Analysis (or the Singular Value Decomposition) and some classical and newly developed clustering algorithms, including a recently developed and very promising method of biclustering, We will also discuss statistical sampling methods (such as an extended version of Gibss sampling).

4.1. Basic tools from linear algebra and statistics

The basic linear algebra tools used in bio-informatics are linear regression (e.g. in the ANOVA model) and principal component analysis (PCA) for feature extraction and dimensionality reduction. An example of this will be shown in Subsection 5.1, where we will use PCA to design a diagnosis test in leukemia and in Subsection 5.2, where we show how PCA can lead to the discovery of new classes. The SVD of an uncentered expression matrix was also proposed in [Alter, 2000] [Nielsen, 2002] to define the notion of 'eigengenes' and 'eigenarrays'.

4.2. Clustering techniques

Biodata feature 2 and 6

Starting from the preprocessed microarray data, a first major computational task is to cluster genes into biologically meaningful groups according to their pattern of expression. Such groups of related genes are much more tractable for study by biologists than the full data itself. As explained in the previous section, we can measure the expression levels of thousands of genes simultaneously. These expression levels can be determined for samples taken at different time points during a certain biological process (e.g., different phases of the cycle of cell division) or for samples taken under different conditions (e.g., cells originating from tumor samples with a different histopathological diagnosis). For each gene, the arrangement of these measurements into a (row) vector leads to what is generally called an expression profile. These expression profiles or vectors can be regarded as data points in a high-dimensional space. Because relatedness in biological function often implies similarity in expression behavior (and vice versa) and because several genes might be involved in the process under study, it will be possible to identify subgroups or clusters of genes that will have similar expression profiles (i.e., according to a certain distance function, the associated expression vectors are sufficiently close to one another). Genes with similar expression profiles are said to be coexpressed. Conversely, coexpression of genes can thus be an important observation to infer the biological role of these genes. For example, coexpression of a gene of unknown biological function with a cluster containing genes with known (or partially known) function can give an indication of the role of the unknown gene [13].

Besides functional relationship, clustering is also a first step preceding further analysis, which includes motif finding, functional annotation, genetic network inference, and class discovery in the microarray data. Moreover, clustering often is an interactive process, where the biologist or medical doctor has to validate or further refine the results and combine the clusters with prior biological or medical knowledge. Full automation of the clustering process is here still far away. Classical 'general-purpose' clustering techniques (developed 'outside' biological research) such as hierarchical clustering, K-means, self-organizing maps, model-based clustering (i.e. based on a mixture of probability distributions) can be applied here (see e.g. [Duda, 2001] [Moreau, 2002] [De Smet, 2002] for a survey). In this paper, we will only briefly discuss two methods: Hierarchical clustering, which is one of the *de facto* standards in bioinformatics, and AQBC (Adaptive Quality Based Clustering). In Table 2 we give a survey of some publically available clustering algorithms.

[13] An example of such a study is [Dabrowski, 2002] where we preformed mRNA expression profiling (6 time points) of mouse primary hippocampal neurons undergoing differentiation in vitro. We have shown that 2319 genes significantly change expression during neuronal differentiation, and the patterns allow to distinguish between several stages of neurite outgrowth. Cluster analysis reveals that a high level of expression of genes involved in the synthesis of DNA and protein, precedes upregulation of genes involved in protein transport, energy generation and synaptic functions. Some 419 genes were found to be likely to belong to an intrinsically driven core of the neuronal differentiation program.

	URL
Cluster	http://rana.lbl.gov/EisenSoftware.htm
J-Express	http://www.molmine.com
Expr. Profiler	http://ep.ebi.ac.uk/
SOTA	http://bioinfo.cnio.es/sotarray
MCLUST	http://www.stat.washington.edu/fraley/mclust
AQBC	http://www.esat.kuleuven.ac.be/~dna/BioI/Software.html

Table 2: Websites with some clustering algorithms

4.2.1. Hierarchical clustering

Biodata feature 6

Hierarchical clustering (see e.g. [Duda, 2001] [Quackenbush, 2001]) is the *de facto* standard in clustering gene expression data. It has the advantage that the results can be nicely visualized (see Figure 9 for an example). Two approaches are possible: a top-down approach (divisive clustering) and a bottom-up approach (agglomerative clustering). As the latter one is most commonly used, we explain it in Figure 9.

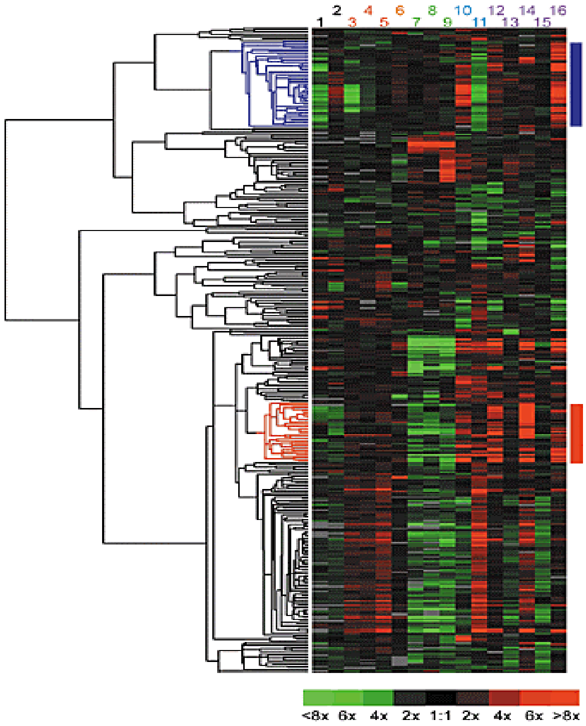
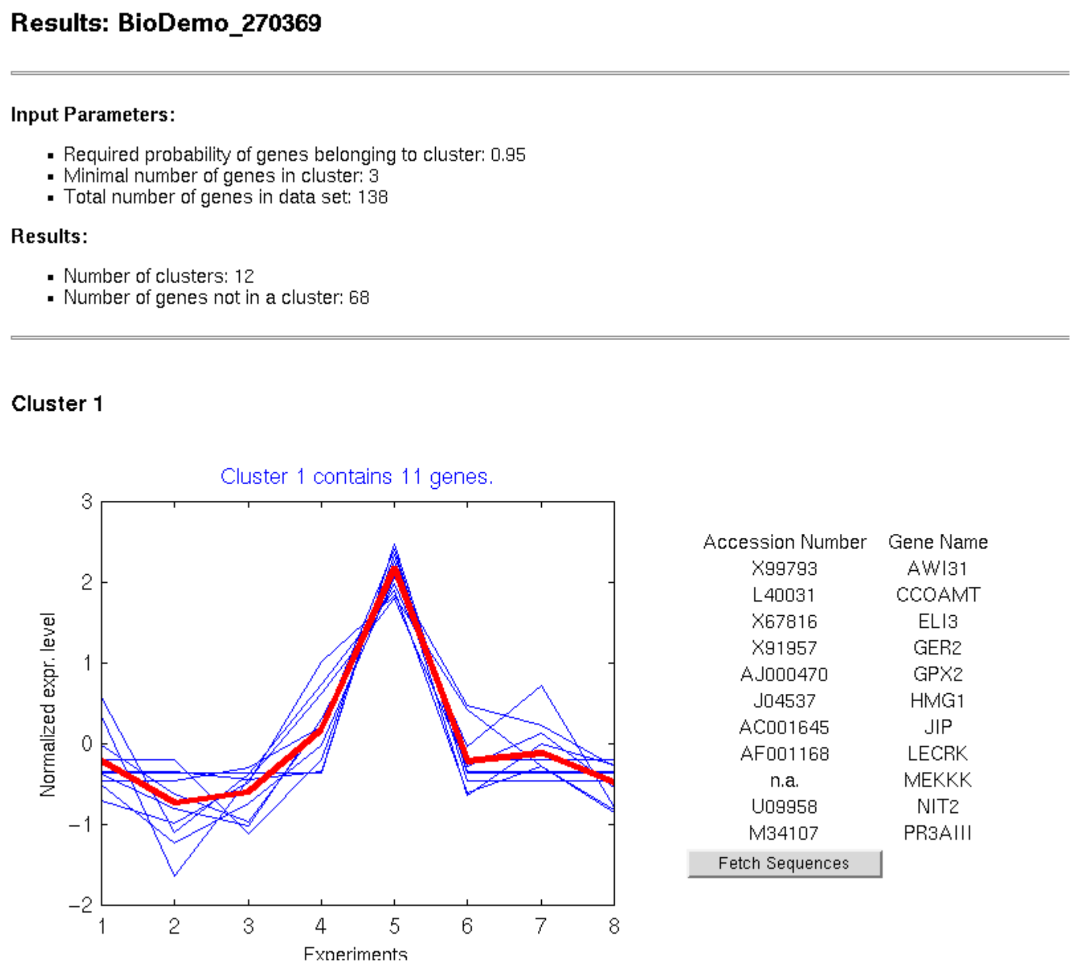


Figure 9. Typical result of hierarchical clustering. The rows of a gene expression matrix are the gene expression profiles, that are vectors the components of which are the intensities as measured over several time points, conditions or patients. One column of such a matrix could be obtained by vectorizing one microarray experiment (the microarray matrix from Figure 2). The end result of hierarchical clustering is a permutation of the rows of the gene expression matrix (the matrix as visualized is also called a 'heat map'). First, each gene expression profile is assigned to a single cluster. The distance (measured in some way, see below) between every couple of clusters is calculated according to a certain distance measure (this results in a pairwise distance matrix). Iteratively (and starting from all singletons as clusters), the two closest clusters are merged and the distance matrix is updated to take this cluster merging into account. This process gives rise to a tree structure where the height of the branches is proportional to the pairwise distance between the clusters. Merging stops if only one cluster is left. Clusters are formed by cutting the tree at a certain level or height. Note that this level corresponds to a certain pairwise distance which in its turn is rather arbitrary (it is difficult to predict which level will give the best biological results). Finally, note that the memory complexity of hierarchical clustering is quadratic in the number of gene expression profiles. This can be a problem when considering the current size of bioinformatics data sets.

4.2.2. Adaptive quality based clustering (AQBC)

One problem with classical clustering algorithms is that typically they require the pre-specification of one or more user-defined parameters, that are often hard to estimate by a biologist (e.g. the number of clusters in K-means when clustering gene profiles, almost impossible to 'guess' a priori). Another problem is that many clustering algorithms often force every data point to be in a cluster. It so happens that in every microarray experiment a considerable number of genes does not contribute to the biological process under study, and therefore will lack co-expression with any other gene in the set. When these gene expression profiles are forced to be included in specific clusters, it leads to 'cluster contamination' phenomena, which have to be avoided for obvious reasons. In addition, the specificity of microarray data (such as the high level of noise or the link to extensive biological information) or also the mere number of expression profiles (that might run into the tens of thousands) have created the need for clustering methods specifically tailored to this type of data, in particular also challenges to cope with the required computational complexity. A new clustering method, specifically developed with microarray data in mind, called adaptive quality-based clustering (AQBC-method), was proposed in [DeSmet, 2002], where also a thorough discussion and examples can be found.



Results: BioDemo_270369

Input Parameters:

- Required probability of genes belonging to cluster: 0.95
- Minimal number of genes in cluster: 3
- Total number of genes in data set: 138

Results:

- Number of clusters: 12
- Number of genes not in a cluster: 68

Cluster 1

Accession Number	Gene Name
X99793	AWI31
L40031	CCOAMT
X67816	ELI3
X91957	GER2
AJ000470	GPX2
J04537	HMG1
AC001645	JIP
AF001168	LECRK
n.a.	MEKKK
U09958	NIT2
M34107	PR3AIII

Figure 10. AQBC is an iterative two-step approach. Intitally, all gene expression profiles are normalized to have norm 1, and the 2-norm between two expression profile vectors is used as a distance measure throughout. Using an initial estimate of the quality of the cluster, a cluster center is located in an area where the density of gene expression profiles is locally maximal. The computational complexity of this first step is only linear in the number of expression profiles. In the second step, called the adaptive step, the quality of the cluster, given the cluster center, found in the first step, is re-estimated so that the genes belonging to the cluster are, in a statistical sense, significantly coexpressed (higher coexpression than could be expected by chance according to a significance level S). To this end, a bimodal and one-dimensional probability distribution (the distribution consists of two terms: one for the cluster and one for the rest of the data) is fitted to the data using an Expectation-Maximization algorithm. Note that, the computational complexity of this step is negligible with respect to the computational complexity of the first step. Finally, step one and two are repeated, using the re-estimation of the quality as the initial estimate needed in the first step, until the relative difference between the initial and re-estimated quality is sufficiently small. The cluster is subsequently removed from the data and the whole procedure is restarted. Note that only clusters whose size exceeds a predefined number are presented to the user. The Figure shows a typical output of our website INCLUSive (see below), in which clustering results are summarized, including accession numbers and names of genes in the cluster.

4.3. Statistical sampling theory algorithms

Biodata feature 1 & 2

In this Subsection, we summarize some recent developments in statistical sampling algorithms, such as Markov Chains Monte Carlo methods, and a new bi-clustering algorithm based on Gibbs sampling.

4.3.1. Gibbs sampling, Markov Chains, EM algorithm

Gibbs sampling is a Markov Chain Monte Carlo (MCMC) method for optimization by sampling. The idea behind sampling methods for optimization is the following. In maximum likelihood methods, such as the EM algorithm[14], we choose a set of parameters to describe our data by $\omega^* = \text{argmax}_\omega \, P(D|\omega)$, in which D represents the data. However, the likelihood function P(D|ω) contains much more information about the data than just the point estimate P(D|ω*). In fact, the posterior distribution, obtained from Bayes rule, p(ω|D)=P(D|ω)P(ω)/P(D), provides a more accurate representation of which parameter values are good candidates to describe our data. For example, if p(ω|D) is multimodal, the modes provide very different models that describe the data well. Also, we can construct confidence intervals for the parameters based on this distribution while we do not get this information from an optimal point estimate. Thus it is advantageous to work with the full probability distribution instead of limiting ourselves to a point estimate. In some cases, it is possible to describe the posterior distribution analytically. However, for more complex models such as DNA sequence models, it is impossible to handle the probability distributions analytically. In that case, several methods are available to generate data according to a complex probability distribution. These are methods such as the Metropolis-Hasting algorithm (which is well-known as the foundation of the simulated annealing algorithm for global optimization), the hybrid MCMC method and Gibbs sampling (for references, see [Thijs, 2003]). Gibbs sampling (see e.g. [Casella, 1992]) is one of the best known Markov Chain Monte Carlo (MCMC) methods. Suppose we want to draw samples for the random variables x, y, and z, but that the marginal distributions or the joint distribution are hard to calculate. Suppose also that the conditional distributions p(x|y,z), p(y|x, z), and p(z|x, y) are available. Starting from initial values y(0) and z(0), the Gibbs sampler draws samples for the three variables in the following manner: x(t) ← p(x|y(t), z(t)); y(t+1) ← p(y|x(t),z(t)); z(t+1) ← p(z|x(t),y(t+1)), for t = 0, 1, 2, ... It can be shown that the sequence y(0), z(0), x(0), y(1), z(1), x(1), . . . , y(k), z(k), x(k) constructs a Markov chain and that, as k → ∞, the distribution of the triplet (x(k), y(k), z(k)) converges to the true joint distribution p(x,y,z). Furthermore, the sequence x(0), x(1), . . . , x(k) itself is a Markov chain, and the distribution of x(k) converges to its true marginal distribution p(x) as k → ∞. We refer to [Thijs, 2001] [Thijs, 2002a] [Thijs, 2002b] [Thijs, 2003] for more details and references and to Subsection 5.4 for an application of Gibbs sampling in motif detection.

4.3.2. Bi-clustering

Consider a microarray data set that contains n genes and m conditions and assume that a single bicluster is present in the data (see Figure 11 for an illustrative example). We introduce two vectors g = (g1 g2 . . . gn) and c = (c1 c2 . . . cm), whose elements are Bernoulli random variables indicating respectively whether the i-th gene and the j -th condition belong to the bicluster. Hereafter we refer to these vectors as the label vectors and the Bernoulli random variables that they contain as the labels. Our goal is to draw samples from the joint distribution p(g, c|D) of g and c conditioned on a discretized microarray data set D. In other words, we want to generate samples for every component in g and c from its respective marginal distribution p(gi|D) or p(cj |D). In the manner of Gibbs sampling, this can

[14] Within a maximum likelihood estimation framework, using parametrized probability densities, the Expectation-Maximization (EM) is a two-step iterative procedure for obtaining the maximum likelihood parameter estimates for a model of observed data and missing values (see [Hastie, 2001] or [Duda, 2001] for an exposition of EM). In the expectation step, the expectation of the data and missing values is computed given the current set of model parameters. In the maximization step, the parameters that maximize the likelihood are computed. The algorithm is started with a set of initial parameters and iterates over the two described steps until the parameters have converged. Since EM is a gradient ascent method, EM strongly depends on the initial conditions. Poor initial parameters may lead EM to converge to a local minimum.

be done by sampling iteratively from the full conditional distributions p(gi|gi*, c,D), for i = 1, 2, . . . , n and p(cj |g, cj*,D), for j = 1, 2, . . . , m, where gi* and cj* denote a label vector with all but the i-th gene and j-th condition label fixed. Because of the Bernoulli nature of the labels, these full conditional distributions will be binomial. However, the parameters of the binomial distributions cannot be easily determined without specifying the models of the data. For simplicity, we only consider the case of discretized (e.g. in grey or color scale bins, i.e. non-continuous) microarray data, which corresponds to multinomial distributions for the data. Let us assume that the data under study is preprocessed in such a way that the background data (the part of the data that does not belong to the bicluster) is generated by one single multinomial distribution characterized by l probabilities φ_i , where l is the total number of bins used for discretization. The bicluster that we seek is a subset of the data where the genes behave similarly under each condition. To put this mathematically, we use a multinomial distribution to model the data under every condition in a bicluster, and we also assume that the multinomial distributions for different conditions of a bicluster are mutually independent. Said in other words, a bicluster pattern is an l times w probability matrix, where each column has l components (l is the number of bins used to discretize the data), and w is the total number of conditions in the bicluster (see Figure 11, where l=3 and w=5).
If we would have the background model and the bi-cluster pattern model, we could calculate the likelihood of the complete data (which includes the observed data D and the labels of g and c). After some tedious – but straightforward – calculations (see [Sheng, 2003] for details), one can calculate the conditional distributions p(gi|gi*, c,D), for i = 1, 2, . . . , n and p(cj |g, cj*,D), for j = 1, 2, . . . , m. The resulting expressions can be used in the following bi-clustering algorithm:

1. Initialization: Randomly assign gene labels and condition labels.
2. Fix the labels of the conditions. For every gene i, (i = 1, 2, ..., n), fix the labels for all the other genes, and
    a. Calculate the binomial distribution for the gene;
    b. Draw a label for gene i from the binomial distribution.
3. Fix the labels of the genes. For every condition j, (j = 1, 2, ...,m), fix the labels of all the other conditions, and
    a. Calculate the binomial distribution for the condition;
    b. Draw a label for condition j from the binomial distribution.
4. Go to Step 2 for a predefined number of iterations

The Monte Carlo aspect of the Gibbs sampler, refers to the fact that the final positions of the bicluster are selected as the ones where the relative count of both the gene labels and the condition labels are larger than a certain predefined threshold (by averaging over the iterations after convergence of the Gibbs sampler has been reached, see [Sheng, 2003] for details). In Figure 11, we present an example that clarifies the whole procedure,

There are several approaches to extend the algorithm to enable the detection of multiple biclusters. We choose to mask either the genes or the conditions selected for the found biclusters and perform the algorithm on the rest of the data. By masking, we mean that the gene or condition labels of all the found biclusters are permanently set to zero. In this way, genes or conditions retrieved for previous biclusters will not further be selected as candidate genes or conditions for any future bicluster, while the background model will still be calculated over all the possible positions in the whole data set including the positions of the masked genes or conditions. Note that this choice does allow the unmasked dimension of the biclusters to be selected multiple times. For example, if the genes are masked, a condition can still belong to multiple biclusters. In this way, the algorithm is iterated on a data set until no bicluster can be found for the unmasked part [15].
We will illustrate biclustering with Gibbs sampling in Subsection 5.2, to discover, starting from microarray experiments, groups of patients with different forms of leukemia, and find the genes that characterize each of these different forms of leukemia.

[15] When no bicluster is present in the dataset, the algorithm should detect this. We do this as follows: We check in Step 4 of the algorithm the number of genes or conditions that belong to the bicluster; If one of them is zero, we reinitialize the algorithm and perform Gibbs sampling again. However, if after a predefined number of reinitializations (for example, ten in our implementation) the algorithm still does not succeed to reach convergence, we terminate the algorithm and consider that the data set does not contain a bicluster.

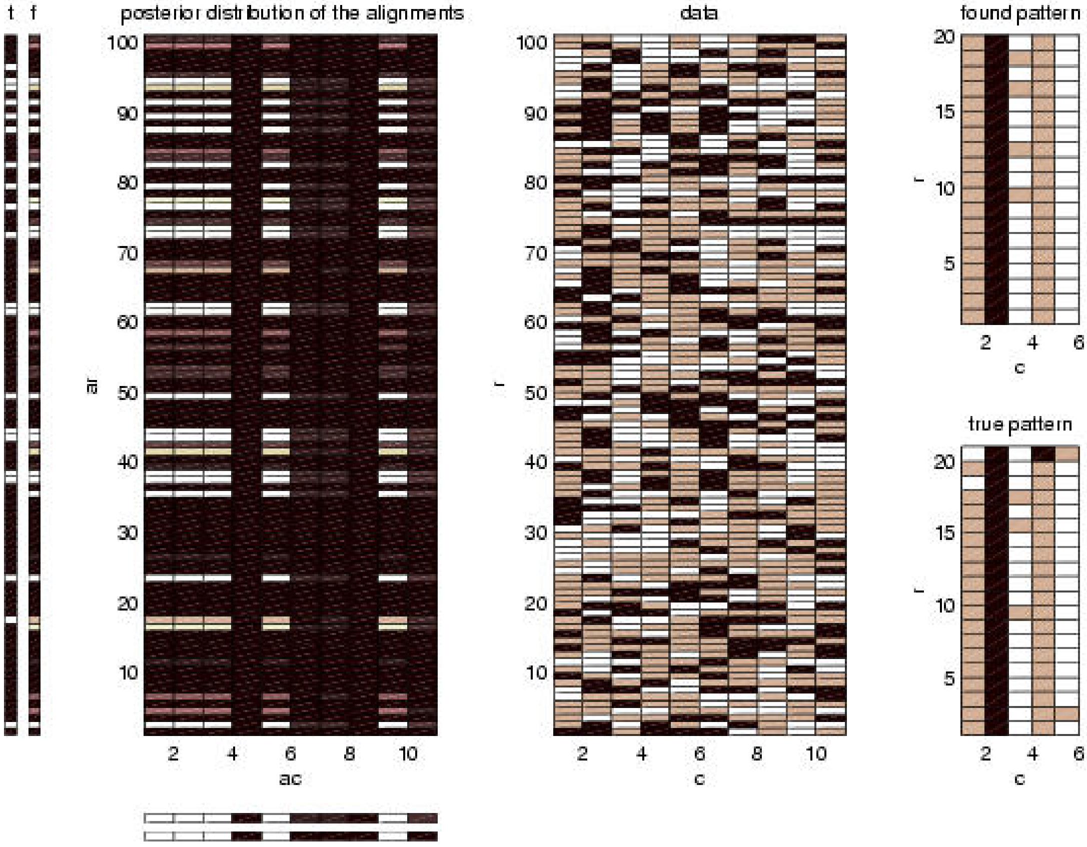


Figure 11. Clarification of the problem that is solved with biclustering. In this example, a pattern of 20 rows and 5 columns (see bottom right) was hidden in a 100 x 10 matrix (middle matrix), discretized using l=3 bins. The pattern was described by 5 sharp independent multinomial distributions, while the background was generated from a uniform multinomial distribution. The biclustering Gibbs sampler [Sheng, 2003] was run for 500 iterations. The left matrix shows the frequency with which every position in the data matrix was sampled as one of the biclustering positions during the sampling procedure. The frequency is reflected by the brightness associated to every position in the plot, where the two extremes, white and black, imply respectively relative frequencies 1 and 0. The vertical and horizontal outer bars that go with this matrix mark the row and column positions of the embedded bicluster (the 20 x 5 true pattern of the bottom right figure) with a white tag (there are 20 white tags on the left most vertical bar and 5 on the lowest horizontal bar). The vertical and horizontal inner bars indicate the expected values of the labels, as estimated from the Gibbs samples. If they are higher than a certain threshold (0.8 in this case), the corresponding matrix element is decided to belong to the pattern. The detected pattern is depicted in the upper right. From these pictures we see that all the columns where the embedded pattern is located, were correctly found, and most of the embedded rows were recovered. In addition, some of the rows that were not designed as the host of the embedded pattern were included in the resulting bicluster, because the patterns in these rows happened to match the one that characterizes the rest of the found bicluster. A more detailed look shows that there was a high variability in the biclusters retrieved at each iteration. However, these biclusters overlapped with each other most frequently at the positions of our final decision, which is illustrated in the left matrix. This is a typical characteristic of Gibbs sampling, which presents targets in terms of distributions rather than deterministic values. In this way, Gibbs sampling also avoids the problem of local maxima that often hinders Expectation-Maximization.

5. Cases

Come forth into the light of things
Let nature be your teacher
Wordsworth

Having discussed some basic algorithms, we are now ready to discuss some (real-life) cases in which we have applied these numerical methodologies to learn something about the biology behind the data. First, in Subsection 5.1. we will show how PCA and hierarchical clustering can be used as a diagnostics tool starting from microarray data, to distinguish between two types of leukemia. Next, in Subsection 5.2., we show how our biclustering method manages to discover 3 different types of leukemia in another microarray data set, which nicely confirms some recent findings in the literature. Next, in Subsection 5.3, we illustrate how ABCQ can be used in knowledge discovery about the mitotic cell cycle in yeast. In Subsection 5.4., we illustrate the use of Gibbs sampling in motif detection, starting from microarray data, which is illustrated by a motif finding application in Arabidopsis thaliana.

5.1. Leukemia diagnostics with PCA and hierarchical clustering

Microarrays have revealed themselves as an important tool for disease management. Indeed, the global expression profile as measured by a microarray of a patient suffering from a specific disease can be used as a molecular fingerprint for that disease. The identification of such disease-specific fingerprints (e.g., expression profiles of tumor cell lines) can help in improving the diagnosis, staging and prognosis of a genetic disease. Although the exact mechanisms of carcinogenesis are usually unknown, it is generally assumed that most cancers originate from genetic disorders. Distinct processes such as contact with carcinogens, viral infections, irradiation can induce mutations in the human genome. This can transform a normal cell into a tumor cell, induce its (uncontrolled) proliferation and finally lead to invasion and metastasis. Mutations leading to cancer can either occur in proto-oncogenes (genes involved in controlled cell proliferation and cell division), in tumor suppressor genes (encoding for inhibitors of uncontrolled cell proliferation), in genes linked with apoptosis (programmed cell death), genes linked with invasion and metastasis, and so on. These mutations can also induce changes in the expression levels of other genes (genes without mutations, but their expression levels are directly or indirectly controlled by the genes in which the mutations occur). It will be the collection of these disturbed expression levels that will determine the phenotype of the tumor. Using microarrays to measure all these expression levels will therefore be of great benefit to know, to determine and to understand the real (clinical) behavior of the tumor cells (with respect to prognosis, therapy response, extend of tumor spread, …). Each microarray experiment determines the expression level of all the genes (for which there are probes present on the array) of a specific cell. This gives, for each experiment, a vector with thousands of components (1 component for each probe present on the array). The determination of these expression levels can be repeated for tumor cells with different properties or from different classes, for example tumors with a different histopathological diagnosis. with a different therapy response, with a different prognosis (e.g., patients that will or will not develop distant metastases). One such a vector would for instance be the *vectorized matrix* that represents the microarray image of Figure 2 (right). The vectors generated by these experiments can be arranged in an expression matrix (the rows contain the expression levels of a specific gene and the columns contain the expression levels of a specific patient, sample or tumor), an example of which is shown in Figure 9 (after hierarchical clustering) or in Figure 15 (using only a grey scale with 3 bins). This expression matrix can now be used for further data analysis (feature selection, class prediction and class discovery).

As an example, we will use a data set containing 72 patients with acute lymphoblastic leukemia (ALL, which we will call class 1) or acute myeloid leukemia (AML, called here class 2). Peripheral blood or bone marrow samples of these patients were analyzed with an Affymetrix oligonucleotide array (with about 7000 probes) (Golub et al., 1999). These measurements (7129 x 72 matrix) are publicly available (http://www-genome.wi.mit.edu/cgibin/cancer/publications/pub_paper.cgi?mode=view&paper_id=43). The 72 patients were divided in two groups:

1. The training set: containing the first 38 patients (27 with ALL and 11 with AML).
2. The test set: containing the remaining 34 patients (20 with ALL and 14 with AML).

We will refer to the entire data set (no distinction between training and test set) as the leukemia data set. The four steps we are going to consider in the analysis of these data are feature selection using PCA, class prediction, class discovery and hierarchical clustering. Using PCA, we will select a limited number of features (a single feature can mathematically be described as a certain function of all the gene expression levels of one patient, resulting in exactly one number for each patient) that contain as much information as possible about a certain class distinction and subsequently use this features to do class prediction or discover. This selection also amounts to a reduction in dimensionality. PCA finds linear combinations of gene expression levels in such a way that these combinations have maximal spread (or standard deviation) for a certain collection of microarray experiments. In fact, PCA searches for the combinations that are most informative. These (linear) combinations are called the Principal Components for this collection of experiments and they can be found by calculating the eigenvectors of Σ (= A.A'/(n-1), covariance matrix of A – note that in this formula A has to be centralized, i.e., the mean column vector of A has to lie in the origin), where A is the expression matrix (m x n matrix – collection of n microarray experiments where m gene expression levels were measured). The eigenvectors or Principal Components with the largest eigenvalues also correspond to the linear combinations with the largest spread in the collection represented by A. For a certain experiment, the linear combinations (or features) themselves can be calculated by projecting the expression vector (for that experiment) onto the Principal Components. Often, we will only use the Principal Components with the largest eigenvalues. So let

- E being the m x 1 matrix expression vector for a certain microarray experiment (where also m gene expression levels were measured);
- P being the m x p matrix, the columns of which are the p Principal Components associated with the p largest eigenvalues for A;
- F the p x 1 vector given by F=P'.E.

F contains the p features (or linear combinations) for the microarray experiment with expression vector E according to the first p Principal Components of the collection of microarray experiments represented by A (said in other words, P.F=(P.P').E is the orthogonal projection of the expression vector E into the dominant principal subspace of dimension p).

For the leukemia data set (m=7129), we have calculated the two (p=2; reduction of dimensionality from 7129 to 2) first Principal Components of the training set (n=38) and determined the associated features (two for each sample) for the samples in the training AND test set. The results are shown in the Figure 12. Note that for this feature selection method the class distinction does not have to be known in advance, making this procedure suitable to be used in combination with class discovery.

In a clinical environment it is important to be able to do predictions (with respect to for example histopathological diagnosis, prognosis and therapy response), using microarray experiments, for new individual patients. This is the problem of class prediction. Here a prediction must be made for samples or patients for which class membership is not known in advance. Based on a set of features and a training set (experiments done in the past - for which class membership is already known), a model (like a Neural Network, a Support Vector Machine, Linear Discriminant Analysis, …) has to be trained. This model can then be used to classify new patients (prediction of class membership).

For the leukemia data set, we used Linear Discriminant Analysis (Fisher) to construct a linear model after Principal Component Analysis . We used the training set to determine the parameters of this model (training – determination of a line in Figure 12 (b) that optimally separates the patients of the training set in ALL (*) and AML (*) patients). Then, we used this model to classify the patients of the test set, which resulted in 3 misclassifications. This is shown in the Figure 12 (b) as well.

Using the extensive arsenal of clinical and morphological parameters, malignant processes can be divided in different diagnostic categories or entities with similar clinical behavior. In most cases, these categories guide the clinical management. It is however possible that rearranging the diagnostic categories might result in groups of patients with less clinical variability, which would allow us to refine the clinical management.With cluster analysis it is possible to automatically find different classes or clusters in a group of microarray experiments without knowing the properties of these classes in advance. A cluster, in general, will group microarray experiments (or the associated patients) with a certain degree of similarity. The distinct classes or clusters generated by the clustering procedure will

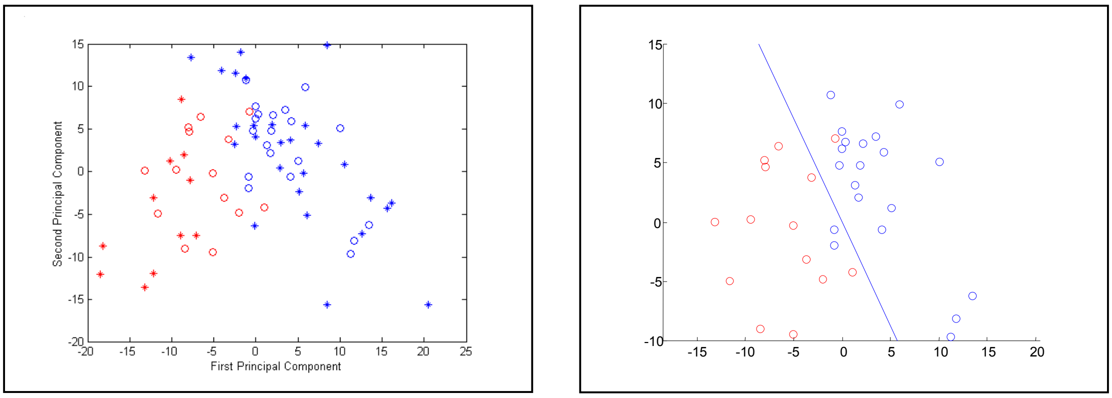


Figure 12 (a) (upper left): Principal Component Analysis for the leukemia data set. First, we calculated the first two Principal Components of the training set. Secondly, we projected every sample from the training AND test set onto these two Principal Components, resulting in two values for each sample (in this figure plotted in the X- and Y-axis). Training set: *=ALL, *=AML; Test set: O=ALL, O=AML.

Figure 12 (b) (upper right): Model for the classification of patients of the test set with acute leukemia after Principal Component Analysis using the first two (p=2) Principal Components of the training set. The parameters of the linear model (blue line) were calculated using the patients of the training set (* and * in Figure 12 (a)). The patients (only the patients of the test set are shown in this Figure) above the line are classified as ALL and below as AML. Note that this results in 3 misclassifications. Test set: O=ALL, O=AML.

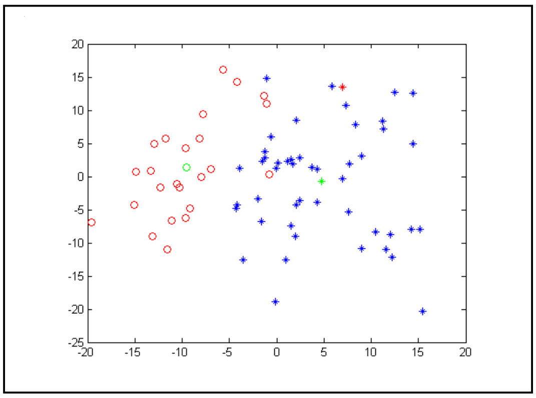


Figure 13: K-means clustering with two clusters for the patients with acute leukemia after Principal Component Analysis according to the first five Principal Components of the entire leukemia data set. Only the first two Principal Components are shown in this Figure. Note the almost perfect correlation between the clusters and the clinical classification (ALL-AML). Cluster 1: *=ALL, *=AML, *=cluster mean; Cluster 2: O=ALL, O=AML, O=cluster mean;

probably - at least partially – match with the existing diagnostic categories used for the current classification tumors. However, it is not excluded that novel, yet unknown diagnostic entities might originate from these analyses which would improve clinical management of cancer.

These techniques can be illustrated using the patients of the entire (here there is NO subdivision between training and test set) leukemia data set. Now imagine that the difference between ALL and AML is not known. In this case we have simply a data set with 72 patients with acute leukemia. We have clustered these 72 experiments or patients using K-means clustering with 2 clusters.

After Principal Component Analysis using the first five Principal Components of the ENTIRE leukemia data set we used the K-means algorithm to find two clusters. The result of this analysis is shown in the Figure 13 (only the first two Principal Components are shown here, but the clustering procedure itself was done in five dimensions). The algorithm has succeeded in finding two clusters. When looking at the first cluster, one can see that all the patients – except one – have ALL. When looking at the second cluster, one can see that all the patients have AML. This means that the procedure was in fact able to redefine the concepts ALL and AML. In this example, nothing new is learned (because ALL and AML were already known), but the result clearly shows the potential of this technique.

We also performed hierarchical clustering on the patients of the entire leukemia data set since this is the method that is commonly used in literature. No Principal Component Analysis is necessary before starting the procedure. This technique iteratively groups the two most similar (according to a certain distance measure like the Euclidean distance or the Pearson correlation coefficient) elements in the data, creating a tree structure. Dependent on the level where the tree is cut, clusters are defined. The result, using the patients of the leukemia data set, is presented in Figure 14. Note that the largest part of the patients with AML is grouped in the right side of the tree structure (see the branch where the arrow is pointing).

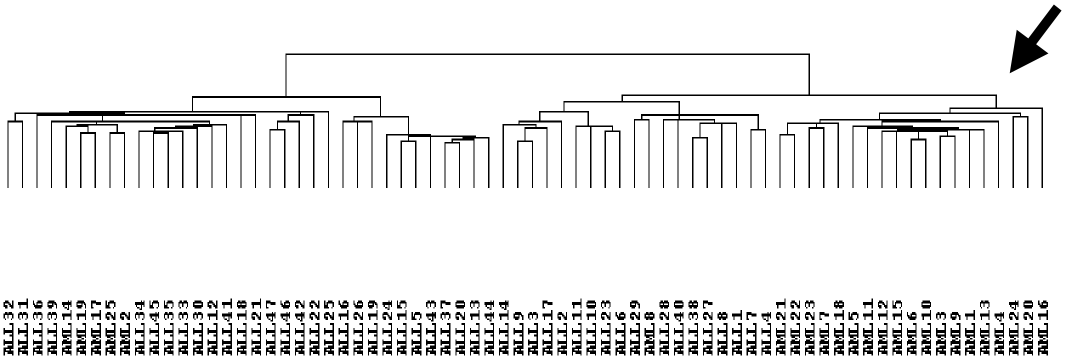


Figure 14: Hierarchical clustering for the patients of the leukemia data set. The terminal branches represent the individual patients (ALL/AML + number). Note that most patients with AML (19 out of 25) are grouped in the branch where the arrow is pointing.

5.2. Discovering leukemia classes by Gibbs sampling biclustering

In this example (which is some recent work [Sheng, 2003] for more details on the algorithm and results), we show how the biclustering algorithm of Section 4 can lead to interesting discoveries of genetic patterns shared by leukemia patients. The data set we start from is a leukemia data set, described in [Armstrong, 2002]. It consists of expression data from Affymetrix chips for 12,600 genes collected from 72 leukemia patients, of which 28 were diagnosed with acute lymphoblastic leukemia (ALL), 20 were mixed-linkage leukemia (MLL) patients, and 24 were acute myelogenous leukemia (AML) patients. The task here is to identify patients that share similar expression behavior over a subset of genes. Because data points with low values are noisy and non-reproducible, a threshold of 100 was put on the original data. A ceiling of 1,600 was also placed because of saturation effects. Next, the variation of each gene along all the patients was examined. Since genes that have consistent behavior over all the patients are not of much interest, only the first 15 percent of genes with the highest standard deviation were selected for further analysis. In this way, the size of the data set was reduced to 1887 genes by 72 patients. This reduced data set was then discretized according to the equal frequency principle. That is, for every gene, we first put its expression data over all the patients in an ascending order, and then divided the data points into a desired number of bins, (which is 3 in the case presented below), in a way such that the number of data points in every bin is the same. Note that the use of the equal frequency principle enables the application of the one-multinomial background discussed in Subsection 4.4.2. We use data from the last three patients of every category to construct a test data set (so the test data set consists of 9 patients, three for each category AML, ALL and MLL). Data from the rest of the patients (i.e. 72-9 = 63 patients) were used as a training set. By masking the patients found after each run, the algorithm succeeded in discovering three biclusters one after another for the training data set. The algorithm stopped after discovering three biclusters. Figure 15 demonstrates the ability of our algorithm to group patients based on their expression behavior over a subset of patients. Furthermore, the patients collected in every bicluster came from the right category. More specifically, (a) the first bicluster selected 19 patients, all of them out of the 25 AML patients in the training set, and found 80 genes to be ‘relevant’; (b) the second bicluster included 18 (out of 21) ALL patients, and 87 genes; (c) the third bicluster consisted of 14 (out of 17) MLL patients and 62 genes.

This example clearly shows the potential of the Gibbs sampling based biclustering technique for novel class discovery. As we will also elaborate on in Section 6, the finer the class distinction is, the more the treatment of the disease can be fine tuned and individualized.

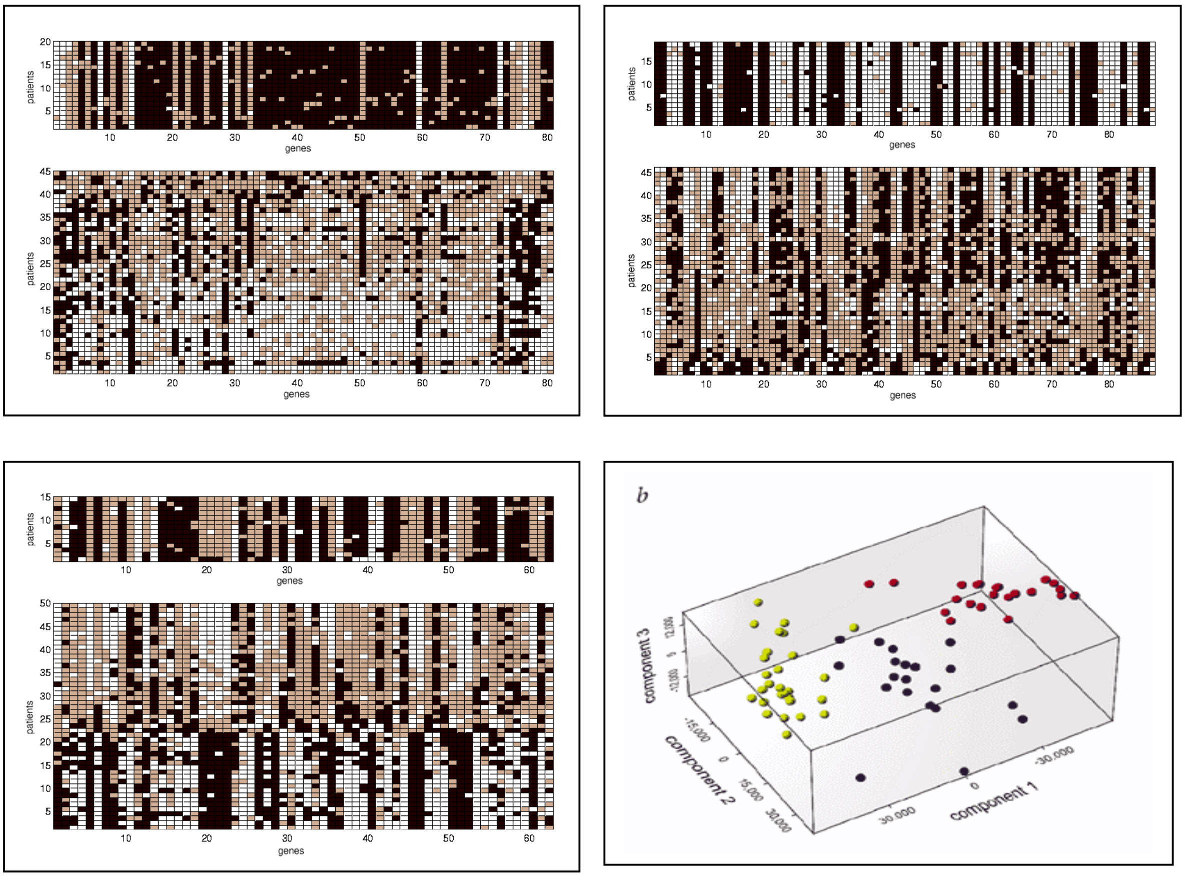


Figure 15. (a, top left) (b, top right) (c, bottom left), (d, bottom, right); Figure (a)-(b)-(c) show patient (rows) versus gene expression levels (columns). The gene expression levels are encoded in 3 grey scale bins. Figure (a): The first bicluster selected 19 patients, all of them out of the 25 AML patients in the training set, and 80 genes that can be considered to be ‘relevant’ for AML; In the upper plot of Figure (a), we clearly see the highly consistent gene pattern over those 80 genes, for the 19 patients in the AML cluster (it almost looks like a rank one matrix !). The lower plot in Figure (a) shows the gene pattern over those 80 genes, for the 63-19=44 remaining patients in the training set. (b) The second bicluster discovered 18 (out of 21) ALL patients, and found 87 genes to be characteristic for this class; The upper plot in (b) shows the gene expression patterns over those 87 genes (again, this looks very much like a rank one matrix !); The lower plot contains the expression patterns over the 63-18=45 remaining patients in the training set; (c) The third bicluster consisted of 14 (out of 17) MLL patients and 62 genes.The upper plot shows the gene expression profile over the 62 genes, while the lower one for the same genes, for the 49 remaining patients. Figure (d) Principal component analysis plot (first 3 dominant components) of ALL (cluster to the right), MLL (cluster in the middle) and AML (cluster to the left) carried out using 8700 genes (see [Armstrong, 2002]) for details).

5.3. AQB-clustering gene expression in the mitotic cell cycle of yeast

AQBC (Adaptive Quality Based Clustering) was tested on an expression profile experiment described in [Cho, 1998] (see also http://cellcycle-www.stanford.edu) studying the yeast cell cycle in a synchronized culture on an Affymetrix chip. The cell cycle of yeast is a fundamental biological system as it reveals the core processes involved in cell replication and growth in general. This knowledge is essential to the understanding of the aberrant processes involved in tumorigenesis and carcinogenesis. This data set can be considered as a benchmark and contains expression profiles for 6220 genes over 17 time points taken at 10-min intervals, covering nearly two full cell cycles. The majority of the genes included in the data set have been functionally classified, which makes this data set an ideal candidate to correlate the results of new clustering algorithms with the biological reality. Our pre-processing included the following steps: (1) data corresponding to the 90 and 100-min measurements were removed; (2) the 3000 most variable genes were selected and (3) the gene expression profiles were normalized. The main results of the cluster analysis with AQBC (minimal number of genes set to 10 and the significance level to 0.95) are summarized in Figure 16.

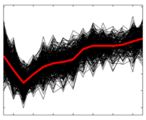
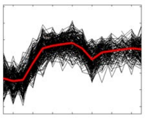
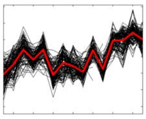
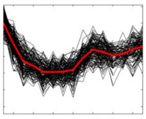
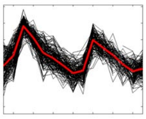
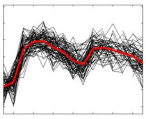
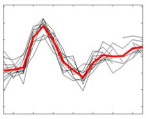
Cluster number	Graphical representation of cluster	Number of ORFs	MIPS functional category (top-level)	ORFs within functional category	P-value (-log$_{10}$)
1		426	energy transport facilitation	47 40	10 5
3		196	cell growth, cell division and DNA synthesis	48	5
4		149	protein synthesis cellular organization	71 107	50 19
5		159	cell rescue, defense, cell death and ageing	20	4
6		171	cell growth, cell division and DNA synthesis	76	24
9		78	cell growth, cell division and DNA synthesis	23	4
37		11	metabolism	9	6

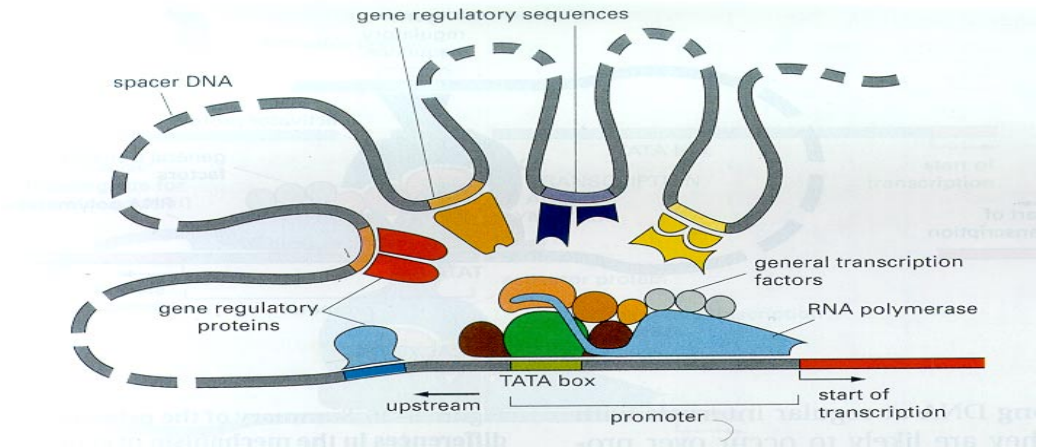
Figure 16. Main results of AQBC on the mitotic yeast cycle benchmark data set. The first column is the cluster number. The plots in the second column show, for each cluster, the normalized expression profiles (15 points on the x-axis) for each gene belonging to that cluster. The red line is the 'average' expression profile. For the biological validation of this result, we mapped the genes in each cluster to the top-level functional categories in the Munich Information center for Protein Sequences (MIPS) Comprehensive Yeast Genome Database. For each cluster we calculated P-values for observing the frequencies of genes in particular top-level functional categories using the cumulative hypergeometric probability distribution [see [De Smet, 2002] [Moreau, 2002] for details). Note that we were able to determine the role of every cluster presented within the yeast cell cycle context and to correlate this role with the behavior of the average profiles of the clusters. We have also found several protein complexes where nearly all members belonged to the same cluster.

5.4. Motif detection

5.4.1. Motifs and regulatory elements

Biodata feature 2

We will now elaborate on how to identify genetics mechanisms that govern the activation of genes in an organism. As explained briefly in Section 2, The DNA not only contains genes, but also all kinds of other short sequences, such as regulatory motifs (e.g. transcription factor binding sites) in the promotor regions of several genes.



© Alberts B. et al. Essential cell biology. Garland Pub. 1997

Figure 17. Short DNA patterns in the neighborhood of the genes serve as switches that control gene expression. Unraveling the mechanisms that regulate gene activity in an organism is a major goal of molecular biology. A main cause of coexpression of genes is that these genes share the same regulation mechanism at the sequence level. Such genes are called co-regulated. Specifically, some control regions (called promoter regions) in the neighborhood of the genes will contain specific short sequence patterns, called binding sites, which are recognized by activating or repressing proteins, called transcription factors. In such a situation, we say that the genes are transcriptionally regulated. Switching our attention from expression data to sequence data, we consider algorithms that discover such binding sites in sets of DNA sequences from coexpressed genes. We analyze the upstream region of those genes to detect patterns, also called motifs, that are statistically overrepresented when compared to some random model of the sequence.

The detection of overrepresented patterns in DNA or amino-acid sequences is called motif finding. Figure 18 provides a survey picture of the different steps for motif finding starting from microarray data. Motif finding is a non-trivial challenge. First of all, a motif typically consists of a limited number of nucleotides (e.g. 10) out of more than 3 billion nucleotides for instance in the human genome. Next, it is important to note that motifs can occur on both strands of the double helix. Transcription factors indeed bind directly on the double-stranded DNA and therefore motif detection software should take this fact into account. In addition, sequences could have either zero, one, or multiple copies of a motif as illustrated in Figure 19. Finally, there may be several special types of motifs such as palindromic motifs, which are a special type of transcription factor binding site from a computational point of view as it is a subsequence that is exactly the same as its own reverse complement (e.g. TCACGTGA). Another complication is formed by gapped motifs or spaced dyads, consisting of two smaller conserved sites separated by a gap or spacer. The spacer occurs in the middle of the motif because the transcription factors bind as a dimer. This means that the transcription factor is made out of two subunits that have two separate contact points with the DNA sequence. The parts where the transcription factor binds to the DNA are conserved but are typically rather small (3-5bp). These two contact points are separated by a non-conserved gap or spacer. This gap is mostly of fixed length but might be slightly variable (see [Thijs, 2003] for details).

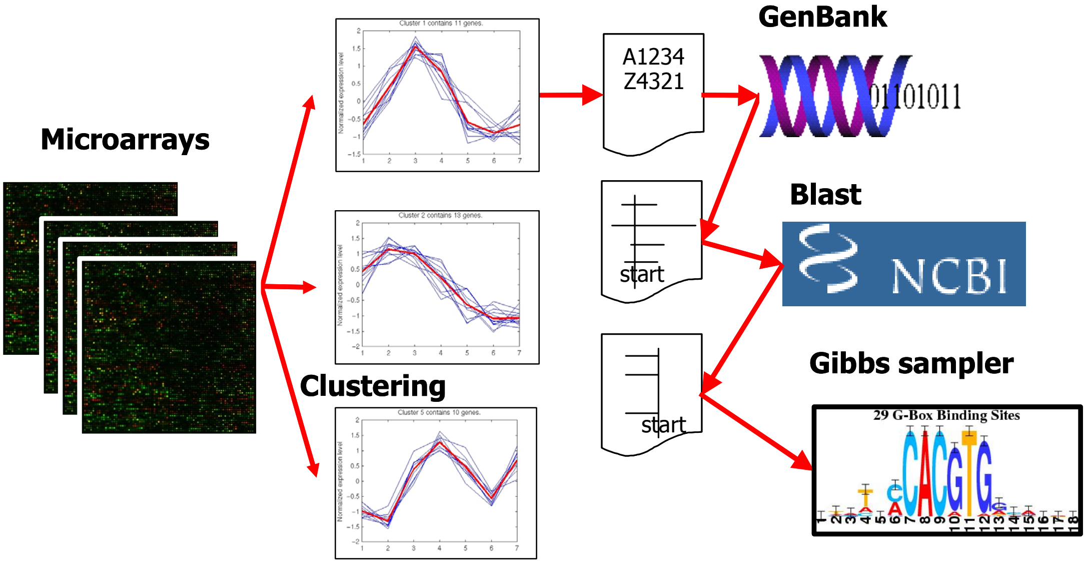


Figure 18. High-level description of data analysis for motif finding starting from microarray data. The cycle starts in the upper left corner, where data are generated from scanned microarray images. After proper quantification and preprocessing, the data are available for clustering in a data matrix, whereupon clustering techniques are applied to detect clusters of so-called co-regulated genes (genes for which the time response looks similar). Three of these clusters are visualized. Next, we concentrate on one cluster, in which the different genes are enumerated (see the arrow) and apply motif finding algorithms to detect sequences of control regions of the genes in this specific cluster. As these motifs are in general unknown, motif finding algorithms detect statistically overrepresented DNA patterns. The motif is visualized on a motif plot, in which the size of the letters is proportional to the probability of having a certain nucleotide at a certain location.

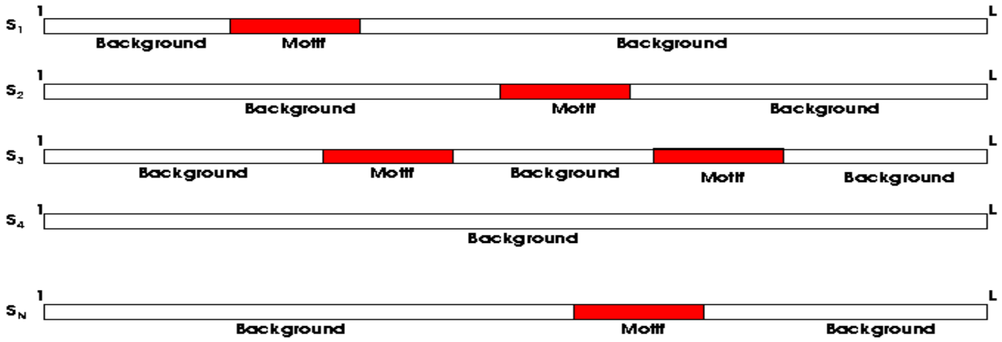


Figure 19. Schematic representation of a set of upstream sequences containing 0, 1, or more copies of a specific motif. In the simplest model, we have a set of DNA sequences where each sequence contains a single copy of the motif of fixed length. Except for the motif, a sequence is described as a sequence of independent nucleotides generated according to a single discrete distribution $\theta_0=(q_0^A, q_0^C, q_0^G, q_0^T)$, which is called the background model. The motif θ_W itself is described by what we call a position weight matrix, which is a 4 x W matrix with elements q_i^b, I=1,…,W and b=A,C,G,T, in which column i is the probability distribution for position i. If we known the location a_i of the motif in a sequence S_i, the probability $P(S_i |a_{i,} \theta_W, \theta_{0)}$ of this sequence, given the motif position, the motif matrix, and the background model can be easily calculated (we refer to [Thijs, 2003] for details). For a set of sequences, the probability of the whole set $S= (S_1,\ldots,S_N)$ given the alignment (i.e., the set of motif positions), the motif matrix, and the background model can also be computed.

Finally, cooperatively binding factors and modules are currently the topic of a lot of research: When only one of the transcription factors binds, there is no activation but the presence of two or more transcription factors activates the transcription of a certain gene (this is like a logical AND gate). If we translate this to the motif finding problem we could either search for individual motifs and try to find, among the list of possible candidates, motifs that tend to occur together. Another possibility is to search for multiple motifs at the same time.

Obviously, regulatory elements play a central role in the study of biological sequences and many databases are available to explore known regulatory elements. The following table provides a list of databases of promoters and gene regulation that are accessible online. Most of these sites are also portals to specific tools for the analysis of regulatory mechanisms.

Database	URL
EPD	www.epd.isb-sib.ch/
TRANSFAC	www.gene-regulation.de/
PlantCARE	sphinx.rug.ac.be:8080/PlantCARE
PLACE	www.dna.affrc.go.jp/htdocs/PLACE
TRRD	www.bionet.nsc.ru/
SCPD	cgsigma.cshl.org/jian/
HPD	zlab.bu.edu/~mfrith/HPD.html
COMPEL	compel.bionet.nsc.ru/compel/

Table 3: Web repositories of regulatory elements.

5.4.2. Algorithms for motif finding

Algorithms to find regulatory elements can be divided into two classes: (1) methods based on word counting and (2) methods based on probabilistic sequence models. The word counting methods analyze the frequency of oligonucleotides in the upstream region and use intelligent strategies to speed up counting and to detect significantly over-represented motifs. These methods then compile a common motif by grouping similar words. Word counting methods lead to a global solution as compared to the probabilistic methods. We refer to [Moreau, 2002b] for a literature survey.

The probabilistic methods represent the motif by a position probability matrix and they assume that the motif is hidden in a noisy background sequence. The simplest model is depicted in Figure 18. The parameters of the motif and background model are pooled in the parameter vector $\theta = (\theta_0, \theta_W)$.

There are basically two approaches for motif finding based on these probabilistic models:

- The idea of the Expectation-Maximization algorithm for motif finding is to find simultaneously the motif matrix, the alignment position, and the background model that maximize the likelihood of the weights and alignments.
- The idea of Gibbs sampling for motif finding extends Expectation-Maximization in a stochastic fashion by not looking for the maximum likelihood configuration but generating candidate motif matrices and alignments according to their posterior probability given the sequences.

We will only be discussing the Gibbs sampling methodology here (see [Moreau, 2002b] for a more complete discussion, including EM based algorithms).

In Gibbs sampling for motif finding, we generate candidate motif matrices and alignments according to their posterior probability given the sequences. Let’s first explicit further how sampling can be applied to motif finding. The idea is to generate plausible motifs and alignments by drawing samples $(\theta^{(i)} , A^{(i)})$ from the posterior $p(\theta , A| S)$. From these samples, we can then track a best motif matrix or alignment or compute an average motif matrix or alignment. However, we need to make an important semantic distinction. Indeed, the alignment A is a property of the data, not of the model. But, while the set of sequences S is available, the alignment is unknown. If the alignment was available in the form of sequence labels, our task of estimating the motif matrix would be greatly facilitated. So, when we set up the likelihood function $P(S|A, \theta)$, the alignment is in fact missing from our sequence data. Therefore, recovering the alignment is called *the missing data problem*. Moreover, recovering the alignment is often less important than estimating the model parameters θ. We could thus try to set up directly the likelihood $P(S|\theta)$. But writing down this likelihood function directly is next to impossible. It is only by introducing the alignment that we get a simple expression for our likelihood. Simplifying the likelihood by introducing new variables is called the *data augmentation method*.

For motif finding, a Gibbs sampler is needed to sample from P(θ,A|S). However, many variables are now involved, which leaves a great deal of leeway in how the exact sampling is set up. The derivation of the exact algorithm is very technical (see [Thijs, 2001] [Thijs, 2002a] [Thijs, 2002b] [Thijs, 2003]) for extensive treatment). Shortly said, the algorithm is basically the Markov chain described above, but the computation of the probability distributions involves the use of multinomial probability distributions (for the probability of the data based on the likelihood function presented previously and on the motif matrix and the background model) and of Dirichlet probability distributions (for the probability of the parameters of the motif matrix).

The following table gives an overview of some of the methods used for motif finding that can be accessed online or where the software is available for download.

Package	URL
RSA tools	www.ucmb.ulb.ac.be/bioinformatics/rsa-tools/
YMF	abstract.cs.washington.edu/~\blanchem/cgi-bin/YMF.pl
Consensus	ural.wustl.edu/softwares.html
MEME	meme.sdsc.edu/meme/website/
Gibbs Sampler	bayesweb.wadsworth.org/gibbs/gibbs.html
AlignACE	atlas.med.harvard.edu
BioProspector	bioprospector.stanford.edu
INCLUSive	http://www.esat.kuleuven.ac.be/inclusive

Table 4: Websites with algorithms for motif finding

5.4.3. Robust motif finding: Motif sampler

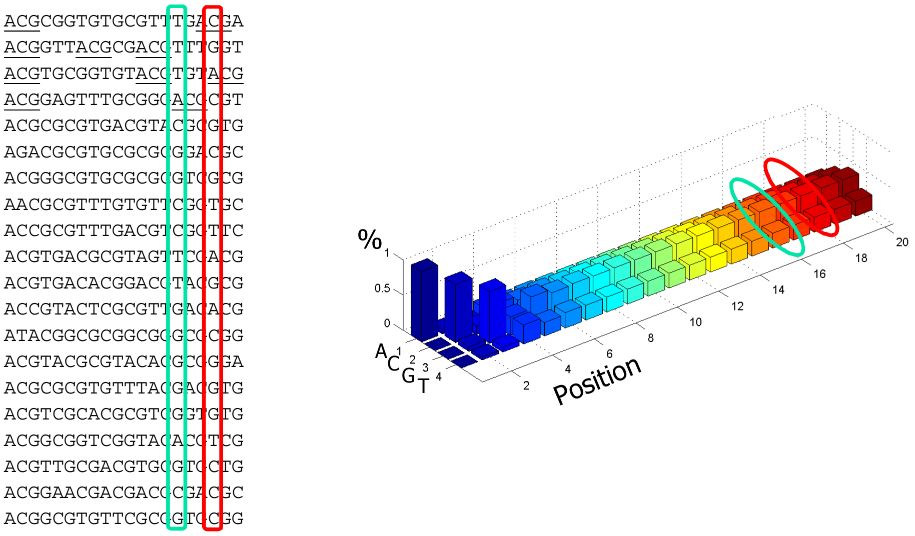
In a series of papers [Thijs, 2001] [Thijs, 2002a] [Thijs, 2002b] [Marchal, 2003] [Aerts, 2003] [Coessens, 2003] and a recent PhD [Thijs, 2003], we have introduced some important modifications to the original Gibbs sampling method. First, a probabilistic framework was used to estimate the expected number of copies of a motif in a sequence. Since both the microarray experiment and the clustering are subject to noise, only a subset of the co-expressed genes is actually coregulated. Furthermore, in higher organisms, regulatory elements can have several copies to increase the effect of the transcriptional binding factor in the transcriptional regulation.

When searching for possible regulatory elements in such a set of sequences we should take into account that the motif will only appear in a subset of the original data set or could have multiple copies. We therefore want to develop an algorithm that distinguishes between the sequences in which the motif is present and those in which it is absent. We reformulated the probabilistic sequence model in such a way that the number of copies of the motif in each sequence can be estimated. Second, we introduced the use of a higher-order background model based on a Markov chain. The use of a higher-order background model is justified by the fact that Markov models have been used extensively in the state-of-the-art gene detection software. Starting from the ideas incorporated in these gene and promoter prediction algorithms, we developed a background model based on a Markov process of order m. This means that the probability of the nucleotide b at a certain position in the sequence depends on the m previous bases in the sequence. A first order model hence has a 4 x 4 transition matrix. An example can be found in Figure 20. Important to know is that the background model can be either constructed from the original sequence data or from an independent data set. The latter approach is more sensible if the independent data set is carefully created, which means that the sequences in the training set only come from the intergenic region and thus do not overlap with coding sequences. Nevertheless the algorithm can also be used for other organisms by building the background model from the input sequences. Background models for several organisms such as *Arabidopsis thaliana, S. cerevisiae, E. coli, Helicobacter pylori, Caenorhabditis elegans*, have been compiled and can be found at http://www.esat.kuleuven.ac.be/~thijs/Work/MotifSampler.html).

Integrating the two proposed modifications into the original Gibbs sampling algorithm for motif finding, lead to our implementation, called Motif Sampler. The input of the Motif Sampler is a set of upstream sequences. In the first step of the algorithm the higher-order background model is chosen. The background model can be pre-compiled or it can be calculated from the input sequences. The algorithm then uses this background model to compute, for each segment of length W in every sequence the probability that the segment was generated by the background model. Second, the alignment vector and the corresponding motif model are initialized. In the next step, the actual core of the sampling procedure starts. The algorithm loops over all sequences and the alignment vector for each sequence is updated. First, the motif model is calculated based on the current alignment vector. This estimated motif model is used to compute for each segment of length W in the selected sequence a weight that is the ratio of the probability that the segment is generated by this motif model divided by the probability that the segment is generated by the background model. Finally, a new alignment vector is selected by taking samples from the normalized distribution of weights over all segments in the given sequence. This procedure is repeated until the motif model has converged. The implementation of our motif finding algorithm is part of our website [Thijs, 2002a] [Coessens, 2003] and is accessible through a web interface: http://www.esat.kuleuven.ac.be/inclusive.

	A	**C**	**G**	**T**
A	*0.0643*	*0.8268*	*0.0659*	*0.0430*
C	*0.0598*	*0.0484*	*0.8515*	*0.0403*
G	*0.1602*	*0.3407*	*0.1736*	*0.3255*
T	*0.1507*	*0.1608*	*0.3654*	*0.3231*

$$\begin{pmatrix} 0.1188 \\ 0.2788 \\ 0.3905 \\ 0.2119 \end{pmatrix}^T \cdot \begin{pmatrix} 0.0643 & 0.8268 & 0.0659 & 0.0430 \\ 0.0598 & 0.0484 & 0.8515 & 0.0403 \\ 0.1602 & 0.3407 & 0.1736 & 0.3255 \\ 0.1507 & 0.1608 & 0.3654 & 0.3231 \end{pmatrix} = \begin{pmatrix} 0.1188 \\ 0.2788 \\ 0.3905 \\ 0.2119 \end{pmatrix}$$



```
ACGCGGTGTGCGTTTGACGA
ACGGTTACGCGACGTTTGGT
ACGTGCGGTGTACGTGTACG
ACGGAGTTTGCGGGACGCGT
ACGCGCGTGACGTACGCGTG
AGACGCGTGCGCGCGGACGC
ACGGGCGTGCGCGCGTCGCG
AACGCGTTTGTGTTCGGTGC
ACCGCGTTTGACGTCGGTTC
ACGTGACGCGTAGTTCGACG
ACGTGACACGGACGTACGCG
ACCGTACTCGCGTTGACACG
ATACGGCGCGGCGGGCGCGG
ACGTACGCGTACACGCGGGA
ACGCGCGTGTTTACGACGTG
ACGTCGCACGCGTCGGTGTG
ACGGCGGTCGGTACACGTCG
ACGTTGCGACGTGCGTGCTG
ACGGAACGACGACGCGACGC
ACGGCGTGTTCGCGGTGCGG
```

Figure 20: Example of a first order Markov chain model of DNA. The transition matrix M in the upper left corner has as its element (i,j) representing the probability that nucleotide i (A=1, C=2, G=3, T=4) follows immediately after nucleotide j. Let the vector v(0) be a 1 x 4 vector with an initial distribution of A,C,T,G (in this example, v(0)=[1 0 0 0], i.e. we always start up with an A). By iterating v(k+1) = T v(k), as k → ∞, we find the stationary distribution, which in this case is v(∞)=[0.1188 0.2788 0.3905 0.2119], which is the eigenvector of M corresponding to the largest eigenvalue, which is 1. The green and red encircled positions in the lower right figure, represent the distribution of the letters A-C-T-G, that could have generated the samples encircled in green and red in the sequences to the left. Higher-order Markov models are based on the same principle and probabilistic matrix representation.

In order to illustrate the increased robustness of Gibbs sampling for motif finding, when using a higher-order Markov background model, we have taken a a data set of coregulated genes from plants [Thijs, 2001] . The data set consists of 33 genes known to be regulated in part by the G-box transcription factor, which is linked to the light response of plants. This set of 33 genes could be one cluster obtained after clustering a gene expression profile matrix. In order to experimentally verify the robustness of our algorithm, we add more and more noisy sequences that do not contain an active motif. We can observe that the performance of the higher-order algorithms is more robust to the addition of noisy sequence than that of the zero-order algorithm as is illustrated in Figure 21.

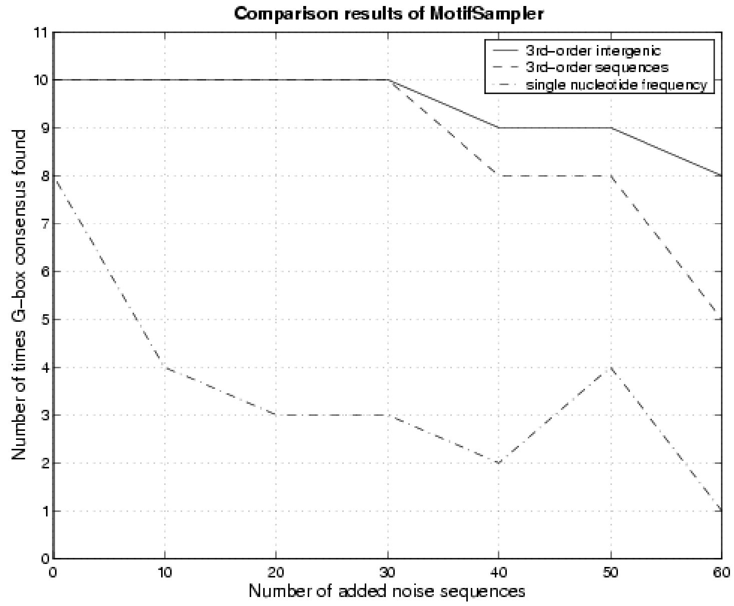
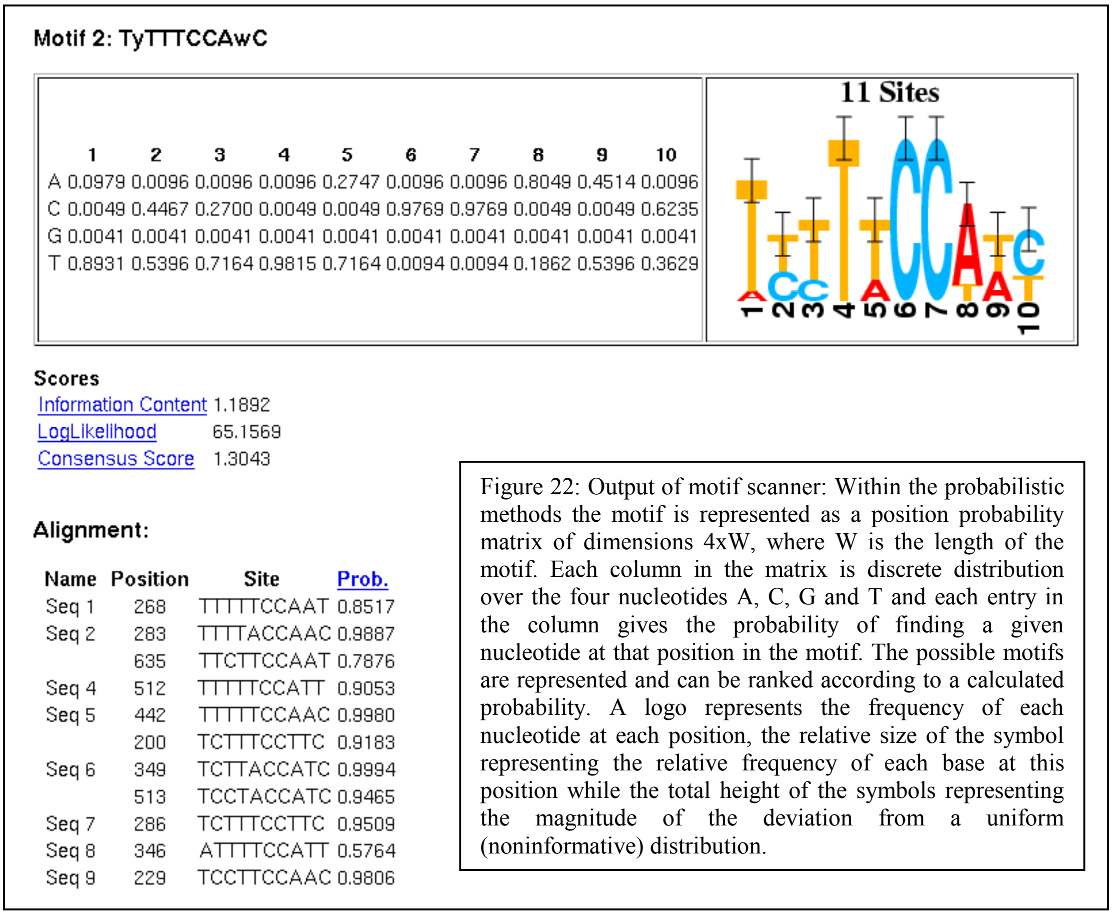


Figure 21: We see how the total number of times the G-box motif is found in 10 repeated runs, and this for three different background models (a 3rd-order intergenic model, a 3rd-order sequence model and a single-nucleotide frequency model, based on a reference set or on the data only). As we can clearly see, the performance of Gibbs sampling for the 0-th order model degrades quite rapidly as a function of added noise sequences, while up to 30 noise sequences can be added, before the motif is less successfully found when higher-order background models are included. This is an example where a priori biological information (in this case, knowledge of background model of the organism) is exploited to robustify an algorithm.

5.3.4. INCLUSive: A software platform for motif finding

Biodata feature 4& 5

The tight link that exists between the clustering of gene expression profiles and the subsequent motif finding, led to the development of a web tool, called INCLUSive, which stands for ***IN****tegrated* ***CL****ustering,* ***U****pstream sequence retrieval, and motif* ***S****ampling* and which is available at http://www.esat.kuleuven.ac.be/inclusive [Thijs, 2002a] [Aerts, 2003] [Coessens, 2003]. Analysis of microarray experiment is not restricted to a single cluster experiment. Inferring ``biological knowledge" from a microarray analysis usually involves a complete analysis going from preprocessing, sequential use of distinct data preparation steps to the use of different complex procedures that make predictions on the data. Clustering predicts whether genes behave similarly while motif finding aims at retrieving the underlying mechanism of this similar behavior. These data-mining procedures make thus predictions about the same biological system. These predictions are in the best case consistent with each other but they can also contradict each other. Combining these methods into a global approach therefore increases their relevance for biological analysis. Moreover, this integration also allows the optimal matching of the different procedures (such as the quality requirements in adaptive quality-based clustering that reduce the noise level for Gibbs sampling for motif finding). Furthermore, such global approaches require extensive integration at the information technology level. Indeed, as is often underestimated, the collection of data from multiple data sources and transformation of the output of one algorithm to the input of the next algorithm are often tedious tasks.



Motif 2: TyTTTCCAwC

	1	2	3	4	5	6	7	8	9	10
A	0.0979	0.0096	0.0096	0.0096	0.2747	0.0096	0.0096	0.8049	0.4514	0.0096
C	0.0049	0.4467	0.2700	0.0049	0.0049	0.9769	0.9769	0.0049	0.0049	0.6235
G	0.0041	0.0041	0.0041	0.0041	0.0041	0.0041	0.0041	0.0041	0.0041	0.0041
T	0.8931	0.5396	0.7164	0.9815	0.7164	0.0094	0.0094	0.1862	0.5396	0.3629

Scores

Information Content	1.1892
LogLikelihood	65.1569
Consensus Score	1.3043

Alignment:

Name	Position	Site	Prob.
Seq 1	268	TTTTTCCAAT	0.8517
Seq 2	283	TTTTACCAAC	0.9887
	635	TTCTTCCAAT	0.7876
Seq 4	512	TTTTTCCATT	0.9053
Seq 5	442	TTTTTCCAAC	0.9980
	200	TCTTTCCTTC	0.9183
Seq 6	349	TCTTACCATC	0.9994
	513	TCCTACCATC	0.9465
Seq 7	286	TCTTTCCTTC	0.9509
Seq 8	346	ATTTTCCATT	0.5764
Seq 9	229	TCCTTCCAAC	0.9806

Figure 22: Output of motif scanner: Within the probabilistic methods the motif is represented as a position probability matrix of dimensions 4xW, where W is the length of the motif. Each column in the matrix is discrete distribution over the four nucleotides A, C, G and T and each entry in the column gives the probability of finding a given nucleotide at that position in the motif. The possible motifs are represented and can be ranked according to a calculated probability. A logo represents the frequency of each nucleotide at each position, the relative size of the symbol representing the relative frequency of each base at this position while the total height of the symbols representing the magnitude of the deviation from a uniform (noninformative) distribution.

As a final example of motif detection, in Table 5, we show the results of motif finding starting from microarray experiments on the response to mechanical wounding of the plant *Arabidopsis thaliana.* The microarray consists of 150 genes related to stress response in plants [Reymond, 2000]. The experiment consists of expression measurements for those 150 genes at 7 time points following wounding (after 30 min, 60 min, 90 min, 3 h, 6 h, 9 h, and 24 h). The expression data was clustered using adaptive quality-based clustering with a significance level of 95 %. Four clusters where identified that contained at least 5 genes and those were selected for motif finding. The Motif Sampler was used to search for 6 motifs of length 8 bp and for 6 motifs of length 12 bp. A background model of order 3 was selected. The analysis was repeated 10 times and only the motifs identified in at least 5 runs were retained. Table 5 presents the motifs found.

Cluster	Consensus motif	Runs	PlanCARE	Description
1	TAArTAAGTCAC	7/10	TGAGTCA	Tissue specific GCN4-motif
[11 seq.]			CGTCA	MeJA-responsive element
	ATTCAAATTT	8/10	ATACAAAT	element assoc. to GCN4-motif
	CTTCTTCGATCT	5/10	TTCGACC	elicitor responsive element
2	TTGACyCGy	5/10	TGACG	MeJa responsive element
[6 seq.]			(T)TGAC(C)	elicitor responsive element
	mACGTCACCT	7/10	CGTCA	MeJA responsive element
			ACGT	Abcissic acid response element
3	wATATATATmTT	5/10	TATATA	TATA-box like element
[5 seq.]	TCTwCnTC	9/10	TCTCCCT	TCCC-motif,light response element
	ATAAATAkGCnT	7/10	-	-
4 [5 seq.]	yTGACCGTCCsA	9/10	CCGTCC	meristem specific activation of H4 gene
			CCGTCC	A-box, light or elicitor responsive element
			TGACG	MeJA responsive element
			CGTCA	MeJA responsive element
	CACGTGG	5/10	CACGTG	G-box light responsive element
			ACGT	Abcissic acid response element
	GCCTymTT	8/10	-	-
	AGAATCAAT	6/10	-	-

Table 5. Results of the motif search in four clusters from a microarray experiment on mechanical wounding in Arabidopsis thaliana [Reymond, 2000] for for the third-order background model. In the first column, the cluster is identified together with the number of genes it contains. The second column gives the consensus of the motif found. The consensus of a motif is the dominant DNA pattern in the motif described using a degenerate alphabet, in which capitals are for strong positions while lower letters are for degenerate positions (e.g. m=A/C, r = A/G, s=C/G , w=A/T, y=C/T). The third column gives the number of times this motif was found in the 10 runs. The fourth column gives matching known motifs found in the PlantCARE database [Lescot, 2002], if any. Finally, the last column gives a short explanation of the matching known motifs. The motifs ATAAATAkGCnT, GCCTymTT and AGAATCAAT are not contained in PlantCARE. This could lead to a biological validation of the fact that these are candidate motifs, as yet undiscovered.

6. Delphi's oracle: Perspectives for the post-genome era

As we will see, Jupiter (technology) will generate an exponentially increasing amount of data, Mars (algorithms and information technology) will drive us towards systems biology and increasing levels of integration, and Venus (biological organisms) will get modified once we have implemented systems for computational biomedicine ! In this Section, we present some visionary views and potential future perspectives on technology, algorithms, systems biology and computational biomedicine.

6.1.Technology for '- omics' and '-omes'

Bioinformatics is much more than we have been describing so far. We have concentrated on *transcriptomics*, which basically studies the relation between the DNA sequence and gene expression profiles over conditions or patients, starting from microarray observations. Incredible as it may seem, DNA sequencing and transcriptomics are only the first 'minor' waves of a new data flood that is awaiting us. In contrast to the fact that (soon) the sequence of an estimated 30000 human genes will be known, so far only a few thousand proteins have been identified, a number that is expected to increase most rapidly. As we have seen in Section 2, proteins are built from 20 amino acids. *Proteomics* (see e.g. [Ezzel, 2002] for a popular account and the series of papers on Proteomics in Nature Insight in Nature, Vol. 422, March 13, 2003, pp.193-237.) stands for three main challenges: identifying all the proteins that are made in a given cell, tissue or organism; determining how those proteins join forces to form networks akin to electrical circuits; outline the precise 3D-structure of the proteins, as this is extremely relevant to discover targets for drugs as the structure largely determines their potential functional associations with other macromolecules[16]. Unfortunately (or luckily enough?), the proteome is much more complicated that the genome ! There is for instance no single human proteome: the pancreas makes a very different set of proteins than the brain does, for instance, and many external conditions and 'triggers' can affect the type of proteins that the body produces. Listing human proteins takes you just so far, but to understand what proteins do in a body, and to develop useful drugs, one needs to unravel how the mix of proteins varies from one cell type to another, and within a cell as conditions change ! By integrating functional genomic and proteomic mapping approaches, biological hypotheses can be formulated with increasing levels of confidence. The availability of complete genome sequences for many organisms, will initiate an evolution from the classical reductionist approach of studying one gene at a time, to a more global and integrative approach that considers all genes at once ! Maps describing different aspects of protein function are to be compiled in what could be considered a 'biological atlas' (see Figure 23 that is borrowed from [Vidal, 2001]). There are many more '-omes' to come[17] ! Technology is also rapidly developing here, with about the same time constants as those in PC-technology (i.e. new (smaller, faster, more accurate) generations of equipment every 2 to 3 years), from the current 2D-gel electrophoresis, over mass spectroscopy to (maybe in the near future) nano-protein arrays, each of which will generate Gb of data!

Not included in Figure 23, but also very important for the near future is the development of *metabolomics.* This is the post-genomic technology that seeks to provide a comprehensive profile of *all* metabolites in a biological sample. This complements the mRNA profiles (that we have been discussing in this paper, the *transcriptomics*), as well as the protein profiles provided by *proteomics*. Central here is the large-scale, parallel interrogation of cell states under different stages of development and defined environmental conditions. It is interesting to note that typically, there are fewer metabolic types than genes or proteins: in the order of 1000 per organism (as compared to several thousand genes for the smallest bacterial genomes and 10's of thousands of genes for complex multicellular organisms). Technology that will be used here includes Gas Chromatography Mass Spectroscopy Machines (see e.g. [Phelps, 2002] for an example).

It goes without saying that the required information technology and the amount of computing required to analyse the generated data will blow up exponentially. Much of the progress in the post-human-genome-project era will be determined by the efficiency of data processing, progress in algorithms and interpretation.

[16] One of IBM's new supercomputers, called Blue Gene, will specifically be devoted to the protein structural form problem. It will cost more than 100 million USD and will have a performance of about 200 teraflops (with a possible extension later towards 1 petaflop).

[17] A glossary of 'omics' and '-omes' can be found at www.genomicglossaries.com/content/omes.asp.

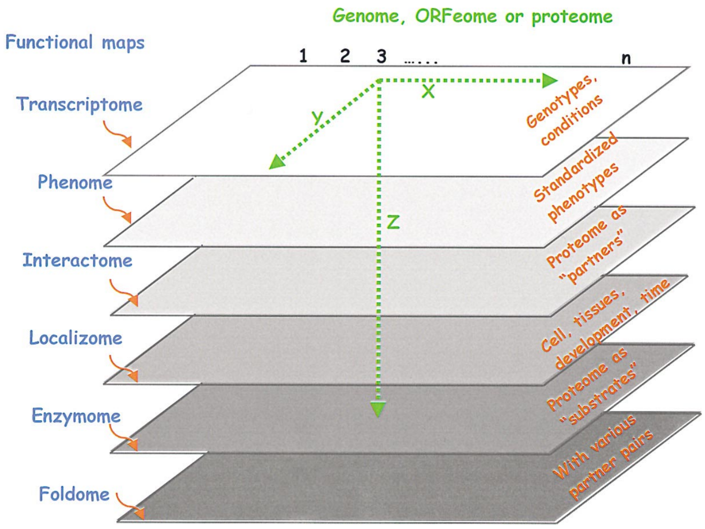


Figure 23. An atlas of functional maps. Each ‘ome’ can be conceived conceptually as a matrix. The x-axis represents the genome, the complete set of ORFs, or the proteome. The y-axis represents a set of conditions, genetic backgrounds, patients or phenotypes. Overlapping maps in the z-axis constitute the biological atlas of an organism. *Interactome* data (protein-protein interaction networks and clusters) can be combined with data from large scale phenotypic analysis (*phenome* mapping, clusters of similar phenotypes) and *transcriptome* profiling (expression clusters of coregulated genes). In the layer of the *localizome*, one looks for information on where various proteins of interest can be found (cells, cellular compartments, localization clusters) and when. It is conceivable that a complete set of proteome’s proteins could be tested for the ability to modify posttranslationally the same set of proteins with the goal of defining a complete *ensymome.* Also, projects will be launched to determine the 3D structure of most protein folds or the *foldome* of several proteomes.

6.2. Algorithms and software

Currently, there are many exciting developments underway for bioinformatics algorithms. In this Subsection, we present a (not complete nor exhaustive) survey of interesting ideas to be developed in the near future.

6.2.1. Advanced tools from linear algebra, statistics and information theory

There is quite some active research to find novel and fast clustering methods. One particular interesting research topic here is to determine a distance measure that would have some biological relevance, contrary to the cosine between two gene expression profile vectors that have been normalized to have norm 1, which is now often used (as e.g. in our AQB-clustering method of Subsection 4.2.2). A possibility here is to use information theoretic criteria, such as the notion of mutual information (see e.g. [Gokcay, 2002] [Kasturi,2003]). But another approach proceeds in two steps: First, the data set is ‘model reduced’, and then clustering is performed in the reduced (model parameter) space. ‘Static’ model reduction typically first performs an SVD (or PCA) and then represents the profiles by their coordinates with respect to the ‘dominant’ left or right singular vectors. Next, clustering is done on the ‘reduced’ coordinate vectors, as we have done to derive the results presented in Figure 13. Another idea is to do ‘dynamical’ modeling: Aprroximately model each individual gene expression profile as a signal being generated by a dynamical system (this is especially relevant when there are several time points in the gene expression profile as one then considers the gene expression explicitly as the response of some dynamical system). One could then define a distance measure between dynamical systems, e.g. the cepstrum-based mutual information measure between ARMA systems as explored in [De Cock, 2002a] [De Cock, 2002b] [Veldhuis, 2003]. In order to give an extreme but illustrative example of what could go wrong when the distance meaure used is not appropriate, consider two hypothetical gene profiles, one of which is a pure sine and the other a cosine with the same frequency. When considered as vectors over a sufficiently large amount of time (and with enough sampling points), these vectors will be (almost) orthogonal. When first modeled as the output of a linear autonomous system, it turns out to be the same dynamical behavior, the only difference being an initial state (a pi/2 difference in phase). Hence, the distance between the two 'generating' dynamical systems would be 0 !

Finally, we are also convinced of the fact that some new statistical techniques, like *Independent Component Analysis* (see e.g. [Hyvarinnen, 2001] or *Higher-Order Tensor Decompositions* (e.g. SVD for 3D-matrices, see [De Lathauwer, 2000] [De Lathauwer, 2001]) might be very useful in the near future for analyzing bioinformatics data, besides the generalizations of the SVD for multiple matrices [De Moor 1992] [De Moor, 1994], which may become important for analyzing several microarray gene profile matrices simultaneously: One of these generalized SVDs, called the *quotient SVD,* was proposed in [Alter, 2003] to describe a comparative mathematical framework for different genome-scale expression data sets where expression is formulated as a superposition of the effects of biological processes common to the different data sets.

6.2.2. Support vector machines and kernel methods

Support vector machines have been around for a while now (see [Vapnik, 1995] with first roots dating back to 1963). Originally, these algorithms were developed to calculate a separating hyperplane in a multi-dimensional data set, a problem that can be solved with a quadratic programming approach. However, due to the introduction of so-called data kernels, SVMs have been generalized towards nonlinear classification problems (exploiting the so-called Mercer condition, also nick-named 'the kernel trick'), based on a quite intuitive idea: If one wants to solve a nonlinear classification or regression problem in a low-dimensional space, the data are first projected into a high-dimensional data space (possibly infinite-dimensional), in which the classification or regression problem becomes linear. This is ultimately achieved by introducing Lagrange multipliers, so that in the dual space the problem becomes an 'easy' one. In our recent book [Suykens, 2002], we have described a new approach, that allows to obtain nonlinear nonparametric but data-driven regressions and classifiers, just by solving a (large) set of linear equations in a least squares sense or a large (symmetric) eigenvalue problem. The resulting algorithms are called *least-squares Support Vector Machines* (LS-SVMs). We refer to the references in this book (and the book chapters itself) for more details.

Least-squares support vector machines are very promising towards the analysis of microarray data (and later on to data generated from proteomics, metabolomics, etc…), because of some interesting built-in features:

- LS-SVM are non-parametric, data-driven, kernel-based methods, that can be used both in regression problems as well as classification problems; The only algorithmic complication that arises is in the solution of a large set of linear equations or a large scale eigenvalue problem, for which nowadays quite some algorithms are publicly available (and which in itself is still a very active area of research); As a matter of fact, this reduction to solving a set of linear equations, has allowed us to start exploiting the full machinery that is available for analysis of data using linear techniques. So we have been developing the kernel versions of least squares regression, linear classification and linear Principal Component Analysis and Canonical Correlation Analysis (e.g. for the LS-SVM version of PCA, see [Suykens, 2003] and Fisher Discriminant Analysis, see [Van Gestel, 2002b]). This is a whole research program in its own, a first outline of which can be found in [Van Gestel, 2002a].
- Moreover, in the near future, exploiting this 'analogy' between linear and kernel-based techniques, we will also *robustify* our algorithms, in much the same way as 'linear' algorithms can be made statistically robust with respect to outliers, deviating data and model assumptions etc…
- LS-SVMS have an inherent data reduction capability, which is very convenient to deal with one of the typical data features of microarrays, namely that the data matrix consists of gene expression levels for many genes (say 5000 to 10000 with current day chips), which correspond to the rows, but only a relatively small amount of columns (which corresponds to the number of patients or the different conditions).

In the near future, we will be applying our expertise in LS-SVM algorithms (see http://www.esat.kuleuven.ac.be/sista/lssvmlab, which is a publicly accessible website where one can download LS-SVM software) to microarray datasets and data coming from other –omics applications as mentioned in Figure 23. Some early references where SVMs have been used for analysing microarray data include [Mukherjee, 1998] [Brown, 2000] [Furey, 2000] [Guyon, 2002].

6.2.3. Bayesian networks – Graphical models

Bayesian reasoning and graphical probablilistic models – familiar AI tools for reasoning under uncertainty – can be used to unravel the mysteries of biological systems, genetic networks and genetic regulation and control. In bioinformatics, probabilistic graphical models have emerged as a dominant approach for data analysis ([see e.g. [Moreau, 2003]). By probabilistic models, we mean here models that express the probability of some observations given a set of model parameters (i.e., the likelihood). Such models are graphical when this probability can be broken down into the combination of several elementary contributions and the probability can then be represented as a graph. Examples of probabilistic graphical models (or graphical models for short) are Hidden Markov Models (HMMs) (for example, for the modeling of protein families) and belief networks (for the reconstruction of gene networks from expression data). An example on one of our ovarian cancer projects is shown in Figure 24. Once the probabilistic model has been set up, the goal is to find good sets of model parameters matching the observed data. This goal can be achieved by maximum likelihood or maximum a posteriori estimation or by Bayesian inference. In Bayesian inference, we use the data to update a prior probability distribution over the parameters into a posterior probability distribution over the parameters given the data. While this approach is computationally intensive and has only recently become really practical, it has been convincingly argued [Baldi, 1998] that this Bayesian framework offers distinctive advantages, such as a systematic way of "incorporating prior knowledge and constraints into the modeling process" and such as the fact that probability distributions over parameters or observations are more informative than optimal point estimates. After the modeling criterion has been chosen, a variety of algorithms are available for estimating the model, such as gradient descent, Expectation-Maximization (EM), or Markov Chain Monte Carlo (MCMC) methods (Gibbs sampling, the Metropolis-Hastings algorithm, or simulated annealing). The application of graphical models in bioinformatics is extremely broad. For example, DNA, RNA, and protein sequences lend themselves to simple probabilistic modeling thanks to their sequential structure and their discrete alphabet. In fact, and this is essential to our argument, probabilistic graphical models are not limited to sequence analysis. In medical informatics, belief networks provide a powerful tool for decision support in diagnosis. Another domain where probabilistic graphical models play an important role is statistical genomics. The goal here is to use patterns of genetic inheritance to determine relationships between genes or genetic loci, or relationships between genes and traits or diseases. One important application is the identification of disease-causing genes (which means genes for which some variants contribute to a disease) from affected families (linkage analysis) or populations (association studies). Similarly, graphical models are powerful tools for phylogeny (which is the reconstruction of the tree of evolution based on genomic sequences) thanks to the graphical description of evolutionary trees and of DNA and protein sequences. Furthermore, the patterns of expression of genes and proteins can be efficiently analyzed with graphical models for clustering and with belief networks.

6.2.4. Open source software and ontologies

There is a growing concern that software and algorithms for bioinformatics, published in journals, should also be publicly available on the Web. The largest repository of Open Source software is the SourceForge (http://sourceforge.net), which hosts nearly 200 bioinformatics software development projects (including GeneX (http://genex.sourceforge.net), the Gene Ontology Consortium (http://www.geneontology.org) etc... Other examples are Bioperl (http://bioperl.org), www.open-bio.org, www.openinformatics.org, and a search on the web reveals many other packages available (see the commentary [Stein, 2002] and the recent Editorial in the journal Bioinformatics [Jamison, 2003]). The algorithms we have been discussing in this paper, can also be downloaded from our website (see URL in the heading of this paper).

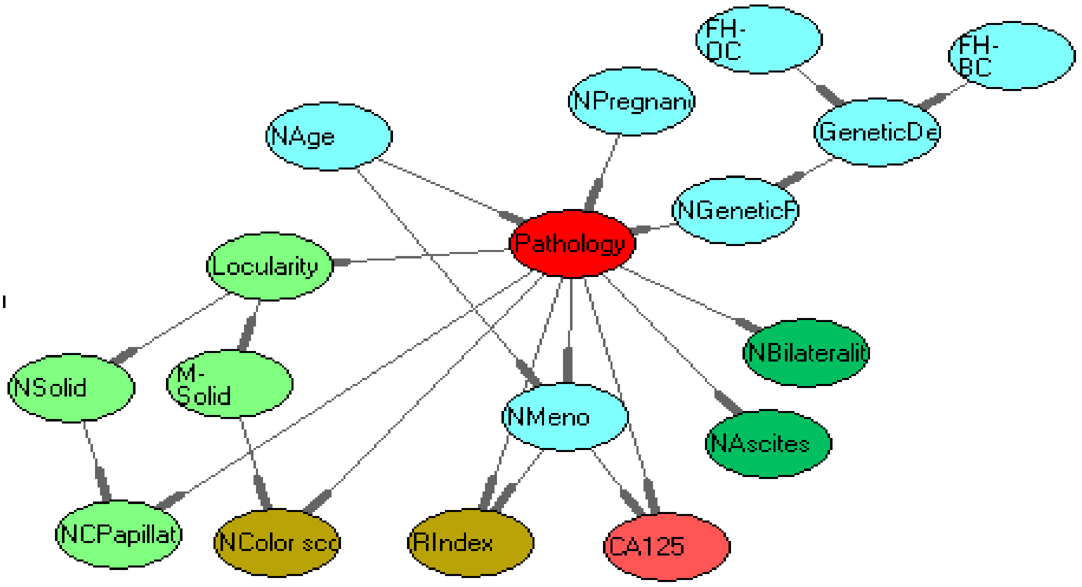


Figure 24. This Bayesian network represents the joint probability distribution of the measurements in the record of a patient with an ovarian tumor. Nodes represent the variables, such as age, pathology (benign vs. malignant), and CA125 serum level. Edges represent the probabilistic conditional dependency between variables. For example, the probability of the CA125 level being low, medium, or high given the menopausal status and pathology is independent of all other variables. Edges are quantified by a probabilistic model, such as a probability table. For example, the presence or absence of a genetic defect (GeneticD = 0 or 1) is a probability value for each of the configurations of the family history of ovarian cancer (FH-OC = 0 or 1) and family history of breast cancer (FH-BC = 0 or 1). How to use Bayesian models, using statistical and machine learning methods, for medical decision support, specifically for tumor diagnosis in oncology, is explained [Antal, 2001] [Moreau, 2002a]. There we summarize the algorithms that are used within the International Ovarian Tumor Analysis (IOTA) consortium (see e.g. http://www.iota-group.org), in which 15 centers from Europe collaborate to set up a diagnosis system for the preoperative discrimination between malignant and benign tumours. One of the extensions we are currently working on, is to include microarray data (obtained from samples of a tumour base) into this diagnosis system.

As the number and the size of sequence and structure databases on the web keeps increasing exponentially, so-called *ontologies* will help to improve the sharing of semantics across the web. The creation of methods for defining and maintaining shared domain models within biology will become – or rather already is - critical. Most biological knowledge nowadays is stored in natural language text, hence impeding ‘access’ for computational approaches that require more structured (e.g. numerical) input. One possibility to cope with this, is to integrate *information retrieval* and *text mining* tools into bioinformatics environments and biological knowledge management systems. Such tools can mine extensive publications databases, such as e.g. Medline (see e.g. [Berry, 2001] [Glenisson, 2003]). Another approach is to structure this input in a conceptual space and set of shared vocabularies that allow at least a subset of biological discourse to be written down formally. This is exactly the purpose of Gene Ontology [Ashburner, 2000], in which hierarchical data models are created and description logics are defined. Knowledge models in systems biology (see below) will be based on ontologies in order to offer a uniform access to the knowledge implied in the system. An ontology is an abstract model of concepts and relations between concepts from a given (research) domain. Ontology in the molecular-biology field will contain concepts such as ‘gene’, ‘gene name’, ‘protein’, ‘molecular function’, ‘biological process’… Relations between these concepts can be defined. For instance, the concept ‘protein’ ‘participates in’ (relation) a certain ‘biological process’ (related concept) and ‘has’ a given ‘molecular function’. Already nowadays, biologists have access to multiple information resources such as public databases, sequence data, scientific literature, and public microarray datasets. A multitude of algorithms have been developed to analyze these data. Since almost every data source and algorithm uses its own data model, combining several algorithms or data sources for data analysis can be a very complex and time-consuming task. Therefore the knowledge models of different data sources need to be integrated. A classic example in the molecular biology field is the problem of genes being stored under different names in different public databases. By defining a 'is a synonym of' relation for the concept 'gene name', information retrieval from multiple data sources can be improved and simplified.

Thus, the use of ontologies for the creation of a knowledge system and for the integration of different components (databases and tools) has many advantages. Because the knowledge in the system is made available in a uniform model, the ontology improves the controlled exchange of biological information among the integrated tools. A uniform representation of prior knowledge and results is also required to compare the results of algorithms with similar functionality, possibly based on different input data (e.g., clustering of text profiles vs. clustering of expression profiles) in a meaningful manner (see [Glenisson, 2003]).

6.3. Systems biology – dynamical systems – Computational cell biology

Interesting challenges lie ahead for researchers active in dynamical systems and control ! Indeed, many computer scientists experience the fact that the DNA sequence is digital, as misleading once they find out how 'fuzzy' the rest of biology is. It should however come as no surpise that increasingly, there are strong indications that dynamical systems modeled by ordinary or partial differential equations, and even hybrid systems, could be the next step in the mathematical characterization of biological phenomena. It should come as no surprise that molecular biology gives raise to many problems that can be studied within the framework of *dynamical systems* and *control theory and feedback*. Up to know, only a limited amount of attempts has been made to unravel the *dynamics* of transcription, gene regulation and gene expression, etc… not in the least because of the unavailability of the appropriate technology to measure things. However, this situation is changing rapidly nowadays as more and more technology is being developed to obtain *in vivo* observations (e.g. as a function of time). Obviously, when for instance looking at the gene expression profiles over a time axis, it is obvious that these are responses (outputs) of a dynamical system and therefore should be modeled as such. On a 'macroscopic' level (which in biology is represented by the cells), also quite some efforts are underway to understand the cell's dynamical behavior (see e.g. [Fall, 2002] for a nice and very interesting survey of what is called *Computational Cell Biology*, including dynamic phenomena in cells, fast and slow time scales, whole-cell models (i.e. *in silico* cell models), intercellular communication (synchronization of oscillators !), spatial modeling (diffusion), biochemical oscillations, cell cycle controls, molecular motors, etc ….)

The concept of a 'system' has pervaded all fields of science, and a new way of thinking about biological systems nowadays is called *Systems Biology*. An interesting survey of this new field can be found in the March 1 2002 issue of Science [Kitano, 2002], where it is stated that a system-level understanding of a biological system can be derived from insight into 4 key properties:

- System structures: Networks of gene interactions, biochemical pathways as well as the mechanisms by which such interactions modulate the physical properties of intracellular and multicellular structures;
- System dynamics: How a system behaves over time under various conditions, to be understood through 'dynamic' metabolic analysis, in which also all kinds of 'causality' problems will be prominently present (see e.g. [Wolkenhauer, 2001] [Wolkenhauer, 2002]);
- Control method: Analyse how mechanisms systematically control the state of a cell, which can lead to potential therapeutic targets for disease treatment;
- Design method: Strategies to modify and construct biological systems (e.g. genetic modifications [Primrose, 2001]), having a priori defined properties.

For the dynamic analysis of networks, mathematical models will have to be created, for which first the scope and abstraction level will have to be defined. *Robustness* is an inherent and essential property of biological systems, which is revealed in three basis mechanisms:

- Adaptation, the ability to cope with environmental changes;
- (Relative) parameter insensitivity;
- Graceful degradation after damage (rather than catastrophic failure).

As we know from engineering systems, these three features are realized by control (negative feedback and feedforward), redundancy, structural stability and modularity. Not surprisingly, these features are also present in biological systems and therefore define a new and challenging road ahead for systems and control researchers. We also refer to the article of John Doyle in the same issue of Science [Csete, 2002] about reverse engineering of biological complexity.

In all of this, the unraveling of gene regulatory networks is of major importance. The clustering analysis we have been describing in this paper, reveals correlation among genes, but it doesn't say anything about 'causality'. Regulatory networks are structured sets of genes and proteins that influence each other's activity. Unraveling regulatory networks helps biologists to understand the regulatory mechanisms that govern protein and gene activity. Gaining such global insight into the cellular behavior has a major impact on applied and fundamental molecular biological research. As mentioned previously, high throughput molecular biological techniques (microarray, proteomics, metabolomics,…) allow making a snapshot of the global cellular behavior and will open the door to holistic approaches. In a regulatory network, the connectivity between genes is hierarchically structured. In this respect genetic circuits are comparable to electronic circuits. When a gene, located on top of a regulation cascade, is activated, the corresponding protein will in turn be responsible for the activation of a next set of genes (Figure 25). In many cases it is not known how the regulatory network acts, i.e., the causal cascaded relationships between the genes are unknown. Regulatory network inference is a methodology to reconstruct from experimental data the underlying cellular regulatory network responsible for the observed behavior.

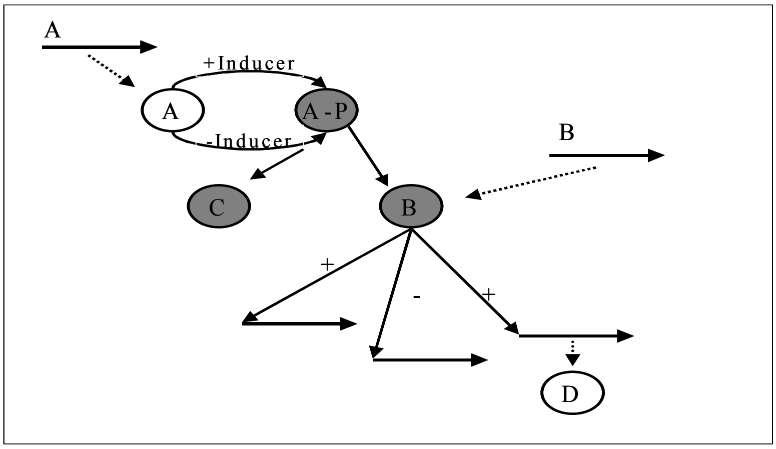


Figure 25. Example of a regulatory network. A key regulator (protein sensor A) is triggered by internal or external factors (inducer). By a posttranslational modification (often a phosphorylation) the key regulator is transformed into its active state and starts inducing a cascade of downstream reactions. Once activated, the key regulator on its turn regulates (activates or represses) the downstream regulators B and C. The reaction proceeds until, in a final step, DNA binding transcriptional regulator proteins (such as B) are activated. Activated or repressed transcriptional regulators influence mRNA transcription. Genes induced by the transcriptional activators are transformed into the corresponding proteins (transcription and translation process). If one of the induced genes encodes a regulator (D) the cascade continues proliferating. Bold arrows indicate Genes. Circles represent inactive proteins. Activated proteins are highlighted. P: indicates a phosphorylation. (+): activation, (-): repression. Dashed arrows represent transcription and translation processes. Other arrows represent the connectivity in the network.

Regulatory network inference implies the reconstruction of the interactions between a large number of variables (theoretically, all genes or gene products whose expression level was measured). Financial, biological and experimental restrictions, however, put a limit on the number of available measurements, which results in a seriously under-determined problem. Typically the interactions between thousands of genes needs to be derived from a few hundred of experiments only. Fortunately, biology puts a number of restrictions on the number of candidate solutions. To begin with, it is known that regulatory networks are sparse. Moreover, a global network is known to consist of small modules of which only a few will be triggered by the experimental conditions tested. Based on scientific a priori knowledge (literature, information retrieval, text mining) or on an integrated bioinformatics analysis of the experimental data (feature extraction methods, clustering, PCA, motif finding, phylogenetic footprinting, and so on), the important units of a functional subnetwork can be identified and formally incorporated during the inference process to reduce the search space of candidate networks. Adding to the complexity of the problem is that high-throughput measurements are inherently noisy, which may lead to inconsistent observations. Moreover, the measurements are typically incomplete in the sense that data may be missing and many variables cannot be observed (e.g., when using microarray data measurements protein-protein interactions are unobserved). These characteristics of the inference procedure and the data imply the need

- For a robust inference algorithm, able to cope with the noise in the data, with unobserved variables and creating a formal framework to incorporate prior knowledge.
- For a knowledge system that retrieves information from different sources and presents it as prior information to the genetic network inference algorithm.

Currently, we are working on Bayesian networks to cope with these challenges. A Bayesian network allows both a compact representation of the joint probability distribution over a large number of variables, and provides an efficient way to use this representation for statistical inference. It consists of a directed acyclic graph that models the interdependencies between the variables, and a conditional probability distribution for each node with incoming edges. In the context of genetic network inference the nodes in the network represent the genes (variables). The edges correspond to the interactions. Bayesian networks are an almost natural choice to model regulatory pathways: As already pointed out, biological networks are structured hierarchically and therefore connections between genes are sparse. In a Bayesian network such sparse connections can easily be represented by conditional independencies. Since Bayesian networks are probabilistic in nature they can capture the stochasticity (either biological or experimental) of the system. Moreover, Bayesian networks can cope with the presence of unobserved values (hidden variables; for example, unmeasured protein-protein interactions). The graphical representation reflects the real biological structure and this structure can be inferred independently from the parameter estimation (maximum a posteriori, Monte-Carlo sampling).

Most important is probably the natural way by which prior information can be introduced into the model. The most important aspect of network inference is to learn as many dependencies between genes and gene products as possible from the raw expression levels of an expression profiling experiment. Given a graph it is possible to learn the probability distributions from the available data and Bayesian priors. One then searches in the space of candidate graphs for a graph that models the dependencies in the data best. The final step from a Bayesian network to a regulatory network is then a minor one.

6.4. Computational biomedicine

In the near future, gene-detecting microarrays could be used to identify an individual’s genetic propensity to a host of disorders. Most genetic differences in people probably take the form of single nucleotide polymorphisms (SNPs, pronounce as ‘snips’), in which a single DNA letter substitutes for another. There is an ongoing quest to characterize the major sources of variation, in which those of functional importance might have significant implications for finding drug targets, predicting disease risk or providing other prognostic information. A chip bearing illness-linked gene variants could be constructed to reveal an individual’s SNPs and thus predict the person’s likelihood of acquiring Alzheimer’s, diabetes, specific cancers, etc… The gene variants we possess influence how our bodies process the medicines we take, which in turn influences the effectiveness of the drugs and the intensity of their side effects. Therefore, microarrays would help physicians to choose those drugs and the corresponding doses that work best for each individual (‘customized medicine’). As the operations of cells become better understood, physicians will be able to make more precise diagnoses, offer more sophisticated therapies (maybe even including gene therapy[18]) and tailor these interventions to an individual’s genetic background and current state of physiological functioning. Many physicians are hoping also that microarrays will evolve into rapid diagnostic tools that would divide patients with similar symptoms into separate groups that would benefit from different treatment plans.

Following the landmark of the Human Genome Project, genomics and bioinformatics are revolutionizing the industry, promising fast and cost-effective development of new drugs. The fight against the impending menace will take place on many fronts: genomics, chemo- and bioinformatics, virtual testing, pharmacogenomics, and a tighter integration of the discovery, development, and trial

[18] Alhtough at the time of writing of this article, there is a (temporary ?) moratorium on gene-therapy trials, as some of the treated patients develop cancers as a side effect (see Nature, vol.421, p.305, 2003).

phases. The completion of the human genome and the advent of the post-genomic era promise a flood of new drug targets to the pharmaceutical industry and a bonanza of biomarkers to the diagnostics industry. Current drugs use only about 500 different molecular targets while it is estimated that genomics and proteomics could eventually provide between 5.000 and 10.000 targets. The question is then moving from discovering targets to predicting which targets have the best potential. As mentioned before, the amount of data produced by new techniques from molecular biology and chemistry is exploding. Chemoinformatics and bioinformatics will be essential to mining these mountains of data. Data handling and analysis will cover the whole drug development processes, tackling questions such as which genes are involved in a pathology, which compounds are likely to show toxic effects, or which patients could present rare side effects. An especially exciting trend is the emerging combination of genomics and bioinformatics for the development of *in silico* models of cells, organs, or even patients. By building extensive mathematical models of biological processes on the basis of genomics measurements, it will become possible to prescreen targets and compounds *in silico*. This improves the quality of the candidates that enter the development phase, thereby significantly reducing development costs. Another trend is that of *pharmacogenomics*, which links drug response to the specific genetic profile of an individual (see e.g [Kalow, 2001]). By identifying those individuals who present rare side effects as having specific genetic variations, it will be possible to rescue some drugs that fail late in the development process (and for which the investment has been maximal) by linking their use to a genetic screening of the patient. Similarly, drugs that fail because they are not active on a sufficiently large portion of the patients could be rescued in some cases (e.g., anti-cancer drugs). Finally, a tighter integration of the whole process (for example, by feeding back genomic patient information into the discovery process) will also increase the efficiency of the development process.

Clearly, for both the healthcare and the pharmaceutical industry, the only way out is the way forward, which means delivering better medical procedures and better drugs more efficiently and more safely, together with targeting problems for which there is a high social demand (chronic and degenerative diseases (such as AIDS, Alzheimer's disease, or arthritis), cardiovascular and metabolic diseases, or cancer). This goal implies an integrated view of the patient in the healthcare process and an intimate understanding of pathologies from the socioeconomic and psychological levels to the genetic and molecular levels. Our work contributes humbly to the technical side of this social endeavor. It addresses questions in oncology stretching from the clinic to the wet lab, such as collecting data from patients for clinical studies, predicting diagnosis from clinical variables, moving new methods from molecular biology towards clinical practice, and studying basic processes in biology as a foundation to medical research. Recurring themes in our work are the focus on a more personalized medicine and the development of computational models that achieve a better understanding of the biological processes at hand, in particular pathologies.

Finally, let us not forget that medicine is for people. To reach its full effect, technical work, like ours, must be embedded in the social, economic, legal, and psychological dimensions of our society. We must make better medicine available to the largest number. Finally, we must insist that medical care is much more than a technical act – empathy and communication are just as essential.

7. References

[Aerts, 2003] Aerts S., Thijs G., Coessens B., Staes M., Moreau Y., De Moor B., ``TOUCAN : Deciphering the Cis-Regulatory Logic of Coregulated Genes", *Nucleic Acids Research*, vol. 31, no. 6, Mar. 2003, pp. 1753-1764.

[Alon, 1999] Alon U., Barkai N., Notterman D.A., Gish K., Ybarra S., Mack D., Levine A.J. Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays. Proc. Of the Nat. Ac. Of Sciences, 1999, 96(12), pp6745-6750.

[Alter, 2000] Alter, O., Brown, P.O., and Botstein, D. (2000) Singular value decomposition for genome-wide expression data processing and modeling. *Proc.Natl.Acad.Sci.USA*, 97, 10101-10106.

[Alter, 2003] Alter, O., Brown, P.O., and Botstein, D. (2003) Generalized singular value decomposition for comparative analysis of genome-scale expression data sets of two different organisms. *Proc.Natl.Acad.Sci.USA*,, **100**, 3351-3356.

[Altman, 2001] Altman R.B. Challenges for Intelligent Systems in Biology. IEEE Intelligent Systems, Nov./Dec. 2001, pp. 14-18.

[Anastassiou, 2001] Anastassiou D. Genomic Signal Processing. IEEE Signal Processing Magazine, pp.8-20, July 2001.

[Antal, 2001] Antal P., Fannes G., De Moor,B., Vandewalle,J., Moreau,Y., and Timmerman,D. (2001a). Extended bayesian regression models: a symbiotic application of belief networks and multilayer perceptrons for the classification of ovarian tumors. Proceedings of the Eight European Conference on Artificial Intelligence in Medicine (AIME'01), Cascais, Portugal, *8*, 177-187.

[Armstrong, 2002] Armstrong S.A., Staunton J.E., Silverman L., Pieters R., Den Boer M., Minden M., Sallan S., Lander E., Golub T., Korsmeyer S. MLL translocations specify a distinct gene expression profile that distinguishes a unique leukemia. Nature Genetics, Vol.30, pp41-47, January 2002.

[Ashburner, 2002] Ashburner M. et al. Gene Ontology: Tool for the unification of biology. Nature Genetics, vol.25, no.1., May 2000, pp.25-29.

[Baldi, 1998] Baldi P., Brunak S. Bioinformatics, the machine learning approach. MIT Press 1998.

[Baldi, 2001] Baldi,P. and Long,A.D. A Bayesian framework for the analysis of microarray expression data: regularized t -test and statistical inferences of gene changes. Bioinformatics *17*, 509-519, 2001

[Berry, 2001] Berry M. (Ed.)Computational Information Retrieval, Proceedings of CIR'00, SIAM Proceedings in Applied Mathematics, SIAM, Philadelphia, 2001, 185 p.

[Bishop, 1997] Bishop M.J., Rawlings C.J. (eds.) DNA and Protein Sequence Analysis, a practical approach. Oxford University Press, 1997.

[Brazma, 2001] Brazma A., Hingamp P., Quakenbush J., Sherlock G., Spellman P., Stoeckert C., Aach J., Ansorge W., Ball C., Causton H., Gaasterland T., Glenisson P., Holstege F., Kim I., Markowitz V., Matese J., Parkinson H., Robinson A., Sarkans U., Schulze-Kremer S., Stewart J., Taylor R., Vilo J., Vingron M., ``Minimum information about a microarray experiment (MIAME) - towards standards for microarray data", *Nature Genetics*, vol. 29, Dec. 2001, pp. 365-371.

[Brown, 2000] Brown M., Grundy W., Lin D., Cristianini N., Sugnet C., Furey T., Ares M., Haussler D. Knowledge-based analysis of microarray gene expression data by using support vector machines. Proc. Of the National Academy of Science, 97, pp.262-267, 2000.

[Brown, 2002] Brown T. Genomes. BIOS Scientific Publishers, 2002.

[Casella, 1992] G. Casella and E. I. George. Explaining the Gibbs sampler. The American Statistician, 46(3):167–174, 1992.

[Cattaneo, 2002] E. Cattaneo, D. Rigamonti, C. Zuccato. The enigma of Huntington’s disease. Scientific American, p.61-65. December 2002.

[Csete M., 2002] Csete M., Doyle J. Reverse engineering of biological complexity. Sience, March 1, 2002, vol.295, pp.1664-1669.

[Coessens, 2003] Coessens B., Gert Thijs, Stein Aerts, Kathleen Marchal, Frank De Smet, Kristof Engelen, Patrick Glenisson, Yves Moreau, Janick Mathys and Bart De Moor. INCLUSive—A Web Portal and Service Registry for Microarray and Regulatory Sequence Analysis. Acc. For publication in Nucleic Acid Research, June 2003.

[Cho, 1998] Cho,R.J., Campbell,M.J., Winzeler,E.A., Steinmetz,L., Conway,A., Wodicka,L., Wolfsberg,T.G., Gabrielian,A.E., Landsman,D., Lockhart,D.J., and Davis,R.W. (1998). A genome-wide transcriptional analysis of the mitotic cell cycle. Mol. Cell. *2*, 65-73.

[Dabrowski, 2002] Dabrowski M., Aerts S., Van Hummelen P., Craessaerts K., De Moor B., Annaert W., Moreau Y., De Strooper B., ``The genetic program driving differentiation of hippocampal neurons is largely cell-autonomous'', Internal Report 02-91, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2002. Accepted for publication in *J. Neurochem*.

[Davies, 2001] Davies K. Cracking the genome; Inside the race to unlock human DNA. Prometheus, 2001.

[De Cock, 2002a] De Cock K., *Principal Angles in System Theory, Information Theory and Signal Processing*, PhD thesis, Faculty of Engineering, K.U.Leuven (Leuven, Belgium), May 2002, 337 p.

[De Cock, 2002b] De Cock K., De Moor B., ``Subspace angles between ARMA models'', *Systems and Control Letters*, vol. 46, 2002, pp. 265-270.

[De Lathauwer, 2000] De Lathauwer L., De Moor B., Vandewalle J., ``On the Best rank-1 and Rank-$(R_1, R_2, ..., R_N)$ Approximation and Applications of Higher-Order Tensors'', *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, Apr. 2000, pp. 1324-1342.

[De Lathauwer, 2001] De Lathauwer L., De Moor B., Vandewalle J., ``Independent component analysis and (simultaneous) third-order tensor diagonalization'', *IEEE Transactions on Signal Processing*, vol. 49, no. 10, Oct. 2001, pp. 2262-2271.

[De Moor, 1992] De Moor B., Van Dooren P., ``Generalizations of the QR and the singular value decomposition'', *SIAM Journal on Matrix Analysis and Applications*, vol.13, no.4, Oct. 1992, pp. 993-1014.

[De Moor, 1994] De Moor B., ``On the structure of generalized singular value and QR decompositions'', Siam Journal on Matrix Analysis & Applications, vol.15-1, Jan. 1994, pp. 347-358.

[DeRisi, 1997] DeRisi,J.L., Iyer,V.R., and Brown,P.O. (1997). Exploring the metabolic and genetic control of gene expression on a genomic scale. Science *278*, 680-686.

[De Smet, 2002] De Smet F., Mathys J., Marchal K., Thijs G., De Moor B., Moreau Y., ``Adaptive quality-based clustering of gene expression profiles'', *Bioinformatics*, vol. 18, no. 5, May 2002, pp. 735-746.

[Duda, 2001] Duda R.O., Hart P.E., Stork D.G. Pattern classification. John Wiley & Sons, New York, 2001.

[Duggan, 1999] Duggan, D. J., Bittner, M., Chen, Y., Meltzer, P., Trent, J. M, Expression profiling using cDNA microarrays. Nature Genetics, vol. 21, no.1, suppl., pp.10-14, 1999.

[Ewens, 2001] Ewens W.J., Grant G.R. Statistical methods in bioinformatics: An introduction. Springer-Verlag, New York, 2001

[Ezzel, 2002] Ezzel C. Proteins rule. Acientific American, April 2002, p.27-33,

[Fall, 2002] Fall C.P., Marland E.S., Wagner J.M. Tyson J.J. (Eds.) Computational Cell Biology. Springer-Verlag, New York, 2002.

[Friend, 2002] Friend S., Stoughton R.B. The magic of microarrays. Scientific American, pp.34-41, February 2002.

[Furey, 2000] Furey T., Duffy N., Cristianini N., Bednarski D., Schummer M., Haussler D. Support vector machine classification and validation of cancer tissue samples using microarray data. Bioinformatics, 16, 10, pp.906-914, 2000.

[Glenisson, 2003] Glenisson P., Antal P., Mathys J., Moreau Y., De Moor B., ``Evaluation of the vector space representation for text-based gene clustering'', Internal Report 02-121, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2002. Accepted for publication in *Proceedings of the Eighth Annual Pacific Symposium on Biocomputing (PSB 2003).*

[Gokcay, 2002] Gokcay E., Principe J. Information Theoretic Clustering. IEEE Transactions on Pattern Analysis and Macine Intelligence. Vol. 24, no.2, February 2002, pp.158-171.

[Golub, 1999] Golub, T.R., Slonim, D.K., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J.P., Coller, H., Loh, M.L., Downing, J.R., Caligiuri, M.A., Bloomfield, C.D., and Lander, E.S. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, **286**, 531-537.

[Griffiths, 1996] Griffiths A.J.F., Miller J.H., Suzuki D.T., Lewontin R.C., Gelbart W.M. An introduction to genetic analysis. W.H. Freeman and co., New York, 1996.

[Griffiths, 1999] Griffiths A.J.F., Gelbart W.M., Miller J.H., Lewontin R.C. Modern genetic analysis. W.H. Freeman and co., New York, 1999.

[Guyon, 2002] Guyon I., Weston J., Barnhill S., Vapnik V. Gene selection for cancer classification using support vector machines. Machine Learning, 46 (1/3), pp.389-422, January 2002.

[Hastie, 2001] Hastie T, Tibshirani and Friedman (2001). The elements of statistical learning. Data mining, inference, and prediction. (page: 236-242). Springer-verlag

[Henig, 2000] Henig R.M. The Monk in the Garden. Houghton Mifflin Company, 2000.

[Hyvarinnen, 2001] Hyvarinnen A., Karhunen J., Oja E. Independent Component Analysis. John Wiley & Sons, 2001.

[Jamison, 2003] Jamison D.C. Editorial: Open bioinformatics. Bioinformatics, Vol.19, no.6, 2003, pp.679-680.

[Kadota, 2001] Kadota,K., Miki,R., Bono,H., Shimizu,K., Okazaki,Y., and Hayashizaki,Y. (2001). Preprocessing implementation for microarray (PRIM): an efficient method for processing cDNA microarray data. Physiol Genomics *4*, 183-188.

[Kalow, 2001] Kalow W. et al. (eds.). Pharmacogenomics. Marcel Dekker Inc., New York, 2001.

[Kari, 1997] Kari L. DNA Computing, Arrival of biological mathematics. The Mathematical Intelligencer, Springer Verlag, New York, Vol.19, No.2, 1997.

[Karp, 2002] Karp G. Cell and molecular biology. Concepts and experiments. John Wiley & Sons, 2002.

[Kasturi, 2003] Kasturi J., Acharya R., Ramanathan M. An information theoretic approach for analysing temporal patterns of gene expression. Bioinformatics, Vol.19, no.4, 2003, pp.449-458.

[Kerr, 2001] Kerr,M.K. and Churchill,G.A. (2001). Statistical design and the analysis of gene expression microarray data. Genet. Res. *77*, 123-128.

[Kitano, 2002] Kitano H. Systems biology: A brief overview. Science, Vol.295, March 1, 2002, pp.1662-1664.

[Knight,2001] Knight,J. (2001). When the chips are down. Nature *410*, 860-861.

[Kreuzer, 1996] Kreuzer H., Massey A. Recombinant DNA and Biotechnology. A guide for teachers. ASM Press (American Society for Microbiology), Washington DC, 1996.

[Lander, 1999] Lander,E.S. (1999). Array of hope. Nat. Genet. *21*, 3-4.

[Lander, 2001] Lander E.S. et al. Initial sequencing and analysis of the human genome. Nature, Vol. 409, no.6822, pp.860-921, Feb. 15, 2001.

[Lescot, 2002] Lescot M., Déhais P., Thijs G., Marchal K., Moreau Y., Van de Peer Y., Rouzé P., Rombauts S., ``PlantCARE, a database of plant cis-acting regulatory elements and a protal to tools for in silico analysis of promoter sequences", *Nucleic Acids Research*, Special Issue on databases, vol. 30, no. 1, Jan. 2002, pp. 325-327.

[Lesk, 2000] Lesk A. The unreasonable effectiveness of mathematics in molecular biology. The Mathematical Intelligencer, Springer Verlag, New York, Vol. 22, no.2, 2000, p.29-37.

[Lipschutz, 1999] R. J. Lipschutz, S. P. A. Fodor, T. R. Gingeras, D. J. Lockheart. High density synthetic oligonucleotide arrays. Nature Genet Suppl., Vol.21, pp.20-24, 1999.

[Marchal, 2002] Marchal K., Engelen K., De Brabanter J., Aerts S., De Moor B., ``Comparison of different methodologies to identify differentially expressed genes in two-sample cDNA arrays", *Journal of Biological Systems*, vol. 10, no. 4, 2002, pp. 409-430.

[Marchal, 2003] Marchal K., Thijs G., De Keersmaecker S., Monsieurs P., De Moor B., Vanderleyden J., ``Genome-specific higher-order background models to improve motif detection", *Trends in Microbiology*, vol. 11, no. 2, Feb. 2003, pp. 61-66.

[Moreau, 2002a] Moreau Y., Marchal K., Mathys J., *Computational biomedicine : a multidisciplinary crossroads*, Siemens Prize, FWO (Flanders, Belgium), 2002, 89 p.

[Moreau, 2002b] Moreau Y., De Smet F., Thijs G., Marchal K., De Moor B., ``Functional bioinformatics of microarray data : from expression to regulation", *Proceedings of the IEEE*, vol. 90, no. 11, Nov. 2002, pp. 1722-1743.

[Moreau, 2003] Moreau Y., Antal P., Fannes G., De Moor B. Probabilistic Graphical Models for Computational Biomedicine. Methods in Information in Medicine, 41, 2/2003.

[Mount, 2001] Mount D. Bioinformatics. Sequence and Genome Analysis. Cold Spring Harbor Laboratory Press, 2001.

[Mukherjee, 1998] Mukherjee S., Tamayo P., Mesirov J., Slonim D., Verri A., Poggio T. Support vector machine classification of microarray data. A.I. memo 1677, MIT Artificial Intelligence Lab, 1998.

[Nielsen, 2002] Nielsen, T.O., West, R.B., Linn, S.C., Alter, O., Knowling, M.A., O'Connell, J.X., Zhu, S., Fero, M,, Sherlock, G., Pollack, J.R., Brown, P.O., Botstein, D., and van de Rijn, M. (2002) Molecular characterisation of soft tissue tumours: a gene expression study. *Lancet*, **359**, 1301-1307.

[Phelps, 2002] Phelps T.J., Palumbo A.V., Beliaev A.S. Metabolomics and microarrays for improved understanding of phenotypic characteristics controlled by both genomics and environmental constraints. Current Opnion in Biotechnology, 13, pp.20-24, 2002.

[Primrose, 2001] Primrose S., Twyman R., Old R. Principles of gene manipulation. Blackwell Science, 2001.

[Reymond, 2000] Reymond P., Weber H., Damond M., Farmer E., Differential gene expression in response to mechanical wounding and insect feeding in Arabidopsis. Plant Cell, 2000,12, pp. 707-719.

[Quackenbush, 2001] Quackenbush J. Computational analysis of microarray data. Nature Reviews Genetics, Vol. 2, pp 418-427, 2001.

[Ridley, 1999]. Ridley M. Genome: The autobiography of a species in 23 chapters. Harper Collins, New York, 1999.

[Schena, 1995] Schena,M., Shalon,D., Davis,R.W., and Brown,P.O. (1995). Quantitative monitoring of gene expression patterns with a complementary DNA microarray. Science *270*, 467-470.

[Sheng, 2003] Sheng Q., Moreau Y., De Moor B., ``Biclustering Microarray data by Gibbs sampling'', Internal Report 03-09, ESAT-SISTA, K.U.Leuven (Leuven, Belgium), 2003.

[Stein, 2002] Stein L. Creating a bioinformatics nation. Nature, Vol.417, May 9 2002, pp.119-120.

[Suykens, 2002] Suykens J.A.K., Van Gestel T., De Brabanter J., De Moor B., Vandewalle J., *Least Squares Support Vector Machines*, World Scientific Publishing Co., Pte, Ltd. (Singapore), (ISBN : 981-238-151-1), 2002.

[Suykens, 2003] Suykens J.A.K., Van Gestel T., Vandewalle J., De Moor B. A support vector machine formulation to PCA analysis and its kernel version. IEEE Transactions on Neural Networks, vol. 14, no.2, pp.447-450, March 2003.

[Sykes, 2002] Sykes B. The seven daughters of Eve. Bantam Press, 2002.

[Thijs, 2001] Thijs G., Lescot M., Marchal K., Rombauts S., De Moor B., Rouze P., Moreau Y., ``A higher-order background model improves the detection by Gibbs sampling of potential promoter regulatory elements'', *Bioinformatics*, vol. 17, no. 12, Dec. 2001, pp. 1113-1122.

[Thijs, 2002a] Thijs G., Moreau Y., De Smet F., Mathys J., Lescot M., Rombauts S., Rouze P., De Moor B., Marchal K., ``INCLUSive : INtegrated Clustering, Upstream sequence retrieval and motif sampling'', *Bioinformatics*, vol. 18, no. 2, Feb. 2002, pp. 331-332.

[Thijs, 2002b] Thijs G., Marchal K., Lescot M., Rombauts S., De Moor B., Rouze P., Moreau Y., ``A Gibbs sampling method to find over-represented motifs in the upstream regions of co-expressed genes'', *Journal of Computational Biology*, Special Issue RECOMB'2002, vol. 9, no. 3, April 2002, pp. 447-464.

[Thijs, 2003] Thijs G. Probabilistic methods to search for regulatory elements in sets of coregulated genes. PhD, Katholieke Universiteit Leuven, Belgium, Department of Electrical Engineering, June 2003.

[Van Gestel, 2002a] Van Gestel T. From linear to kernel based methods in classification, modelling and prediction. PhD Thesis, Department of Electrical Engineering, Katholieke Universiteit Leuven, May 2002, 286 pp.

[Van Gestel, 2002b] Van Gestel T., Suykens J., Lanckriet G., Lambrechts A., De Moor B., Vandewalle J., ``Bayesian Framework for Least Squares Support Vector Machine Classifiers, Gaussian Processes and Kernel Fisher Discriminant Analysis'', *Neural Computation*, vol. 15, no. 5, May 2002, pp. 1115-1148.

[Van Helden, 1998] van Helden J., B. André, L. Collado-Vides. Extracting regulatory sites from upstream region of yeast genes by computational analysis of oligonucleotide frequencies. J. Mol. Biol., 1998, 281, pp827-842.

[Van Helden, 2000] Van Helden J., B. André, L. Collado-Vides. A web site for the computational analysis of yeast regulatory sequences. Yeast, 2000, 16, pp. 177-187.

[Van ‘t Veer, 2002] van 't Veer, L.J., Dai, H., van de Vijver, M.J., He, Y.D., Hart, A.A., Mao, M., Peterse, H.L., van der Kooy, K., Marton, M.J., Witteveen, A.T., Schreiber, G.J., Kerkhoven, R.M., Roberts, C., Linsley, P.S., Bernards, R., and Friend, S.H. (2002) Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, **415**, 530-536.

[Vapnik, 1995] Vapnik V. The nature of statistical learning theory. Springer-Verlag, New York, 1995.

[Veldhuis, 2003] Veldhuis R., Klabbers E. On the computation of the Kullback-Leibler measure for spectral distances. IEEE Transactions on Speech and Audio Processing, Vol.11, no.1, January 2003.

[Venter, 2001] Venter J.C. et al. The sequence of the human genome. Science, vol.291, no.5507, pp.1304-1351, Feb.16, 2001.

[Vidal, 2001] Vidal M. A biological atlas of functional maps. Cell, Vol. 104, 333–339, February 9, 2001.

[Watson, 1953] Watson J., Crick F. A structure for deoxyribose nucleic acid. Nature vol. 171, pp.737-738, 1953.

[Wolkenhauer, 2001] Wolkenhauer O. Systems biology: The reincarnation of systems theory applied in biology ? Henry Stewart Publications, Briefings in bioinformatics. Vol.2, no.3, 258-270, September 2001.

[Wolkenhauer 2002] Wolkenhauer O. Mathematical modeling in the post-genome era: Understanding genome expression and regulation – a system theoretic approach. BioSystems, 65, pp.1-18, 2002.

[Yang, 2002] Yang Y.H., Dudoit S., Luu P., Lin D., Peng V., Ngai J., Speed T.P. Normalization for cDNA microarray data: a robust composite method addressing single and multiple slide systematic variation. Nucleic Acids Res., 30, 2002, pp. E15.