Functional Bioinformatics of Microarray Data: From Expression to Regulation

YVES MOREAU, FRANK DE SMET, GERT THIJS, STUDENT MEMBER, IEEE,
KATHLEEN MARCHAL, AND BART DE MOOR, SENIOR MEMBER, IEEE

Invited Paper

Using microarrays is a powerful technique to monitor the expression of thousands of genes in a single experiment. From series of such experiments, it is possible to identify the mechanisms that govern the activation of genes in an organism. Short deoxyribonucleic acid patterns (called binding sites) near the genes serve as switches that control gene expression. As a result similar patterns of expression can correspond to similar binding site patterns. Here we integrate clustering of coexpressed genes with the discovery of binding motifs. We overview several important clustering techniques and present a clustering algorithm (called adaptive quality-based clustering), which we have developed to address several shortcomings of existing methods. We overview the different techniques for motif finding, in particular the technique of Gibbs sampling, and we present several extensions of this technique in our Motif Sampler. Finally, we present an integrated web tool called INCLUSive (available online at http://www.esat.kuleuven.ac.be/~dna/BioI/Software.html) that allows the easy analysis of microarray data for motif finding.

Keywords*—Adaptive quality-based clustering, clustering, Gibbs sampling, microarray, motif finding, regulation.*

I. INTRODUCTION

Unraveling the mechanisms that regulate gene activity in an organism is a major goal of molecular biology. In the past few years, microarray technology has emerged as an effective technique to measure the level of expression of thousands of genes in a single experiment. Because of their capacity to monitor many genes, microarrays are becoming the workhorse of molecular biologists studying gene regulation. However, these experiments generate data in such amount and of such a complexity that their analysis requires powerful computational and statistical techniques. As a result, unraveling gene regulation from microarray experiments is currently one of the major challenges of bioinformatics.

Starting from microarray data, a first major computational task is to cluster genes into biologically meaningful groups according to their pattern of expression [23]. Such groups of related genes are much more tractable for study by biologists than the full data themselves. Classical clustering techniques such as hierarchical clustering [13] or K-means [51] have been applied to microarray data. Yet the specificity of microarray data (such as the high level of noise or the link to extensive biological information) has created the need for clustering methods specifically tailored to this type of data [17]. We overview both the first generation of clustering methods applied to microarray data as well as second-generation algorithms, which are more specific to microarray data. In particular, we address a number of shortcomings of classical clustering algorithms with a new method called adaptive quality-based clustering [10] in which we look for tight reliable clusters.

In a second step, we ask what makes genes belong to the same cluster. A main cause of coexpression of genes is that these genes share the same regulation mechanism at the sequence level. 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

Manuscript received March 15, 2002; revised July 15, 2002. This work was supported in part by the Research Council KUL: Concerted Research Action GOA-Mefisto 666 (Mathematical Engineering), IDO (IOTA Oncology, Genetic networks), several Ph.D., post-doc, and fellow grants; in part by the Flemish Government: Fund for Scientific Research Flanders (several Ph.D. and post-doc grants, G.0115.01 [bio-i and microarrays], G.0407.02 [support vector machines], research communities ICCoS, ANMMM), AWI (Bil. Int. Collaboration Hungary), IWT (STWW-Genprom [gene promoter prediction], GBOU-McKnow [knowledge management algorithms], several Ph.D. grants); and in part by the Belgian Federal Government: DWTC (IUAP IV-02 [1996–2001] and IUAP V-10-29 [2002–2006]: Dynamical Systems and Control: Computation, Identification and Modeling).

The authors are with the Department of Electrical Engineering, Katholieke Universiteit Leuven, Leuven, Belgium (e-mail: yves.moreau@esat.kuleuven.ac.be).

Digital Object Identifier 10.1109/JPROC.2002.804681

0018-9219/02$17.00 © 2002 IEEE

discover such binding sites in sets of deoxribonucleic acid (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.

Two classes of methods are available for motif finding: word-counting methods and probabilistic sequence models. Word-counting methods are string-matching methods based on counting the number of occurrences of each DNA word (called oligonucleotide) and comparing this number with the expected number of occurrences based on some statistical model. Probabilistic sequence models build a likelihood function for the sequences based on the motif occurrences and a model of the background sequence. Probabilistic optimization methods, such as expectation maximization (EM) and Gibbs sampling, are then used to search for good configurations (motif model and positions). After briefly presenting the word-counting methods and the method based on EM, we discuss the basic principles of Gibbs sampling for motif finding more thoroughly. We also present our Gibbs sampling method, called the Motif Sampler, where we have introduced a number of extensions to improve Gibbs sampling for motif finding, such as the use of a more precise model of the sequence background based on higher-order Markov chains. This improved model increases the robustness of the method significantly.

These two steps, clustering and motif finding, are interlocked and specifically dedicated to the discovery of regulatory motifs from microarray experiments. In particular, clustering needs to take into account that motif finding is sensitive to noise. Therefore, we need clustering methods that build conservative clusters for which coexpression can be guaranteed in an attempt to increase the proportion of coregulated genes in a cluster. This is one of the requirements that warranted the development of our adaptive quality-based clustering algorithm. Also, the motif-finding algorithms are specifically tailored to the discovery of transcription factor binding motifs (while related algorithms can be developed for slightly different problems in protein sequence analysis). These tight links mandate our integrated presentation of these two topics in this paper. Furthermore, the same links call for integrated software tools to handle this task in an efficient manner. Our INCLUSive web tool (http://www.esat.kuleuven.ac.be/~dna/BioI/Software.html) supports motif finding from microarray data. Starting with the clustering of microarray data by adaptive quality-based clustering, it then retrieves the DNA sequences relating to the genes in a cluster in a semiautomated fashion, and finally performs motif finding using our Motif Sampler (see Fig. 1). Integration is paramount in bioinformatics as, by optimally matching the different steps of the data analysis to each other, the total analysis becomes more effective than the sum of its parts.

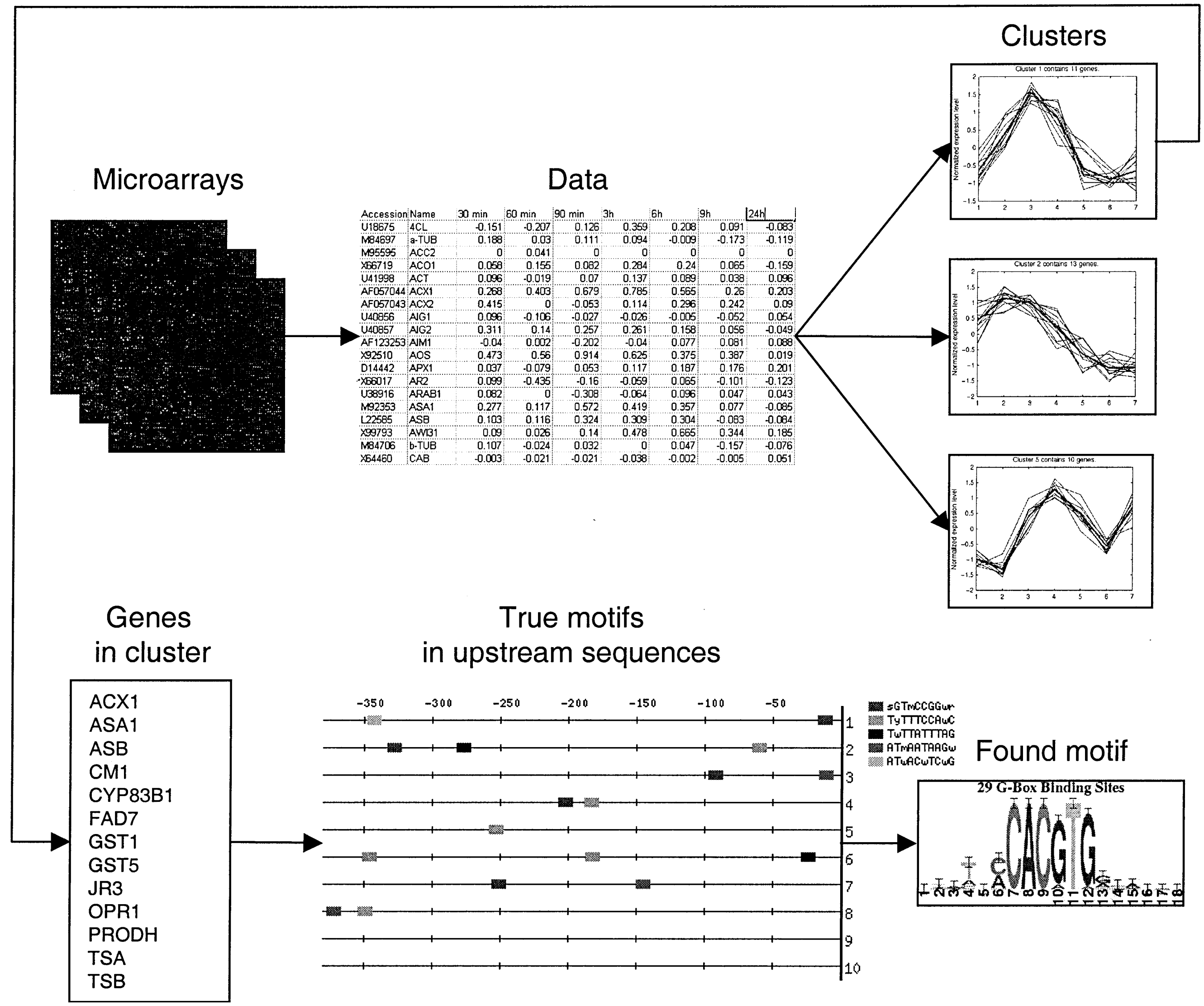
This paper is organized as follows. In Section II, we briefly describe microarray technology; in Section III, we summarize the basic concepts of molecular biology relevant to motif finding. In Section IV, we overview current clustering algorithms for microarray data, in particular our adaptive quality-based clustering algorithm. We also discuss methods for preprocessing microarray data to make them suitable for clustering and methods for assessing the quality of clustering results from the statistical and biological standpoints. Next, we describe in Section V the problem of motif finding and overview several of the methods available for this problem. We then explore, in Section VI, the basic principles of Gibbs sampling for motif finding and describe the extensions necessary for its efficient practical application. In Section VII, we describe our INCLUSive web tool for the integration of adaptive quality-based clustering and Gibbs sampling for motif finding.

II. Measuring Gene Expression Profiles

Cells produce the proteins they need to function properly by 1) *transcribing* the corresponding genes from DNA into messenger ribonucleic acid (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. After RNA extraction from a biological sample, fluorescently labeled complementary DNA (cDNA) or complementary RNA (cRNA) is prepared. This fluorescent sample is then hybridized to the DNA present on the array. 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. In this way, microarrays allow simultaneous measurement of expression levels of thousands of genes in a single hybridization assay.

Two basic array technologies are currently available: cDNA microarrays and gene chips. cDNA microarrays [12] are small glass slides on which double-stranded DNA is spotted. These DNA fragments are normally several hundred base pairs (bp) in length and are often derived from reference collections of expressed sequence tags (which are subsequences from an mRNA transcript that uniquely identify this transcript) extracted from many sources of biological materials so as to represent the largest possible number of genes. Usually each spot represents a single gene. *Two* samples are



Accession	Name	30 min	60 min	90 min	3h	6h	9h	24h
U18675	4CL	-0.151	-0.207	0.126	0.359	0.208	0.091	-0.083
M84697	a-TUB	0.188	0.03	0.111	0.094	-0.009	-0.173	-0.119
M95595	ACC2	0	0.041	0	0	0	0	0
X66719	ACO1	0.058	0.155	0.082	0.284	0.24	0.065	-0.159
U41998	ACT	0.096	-0.019	0.07	0.137	0.089	0.038	0.096
AF057044	ACX1	0.268	0.403	0.679	0.785	0.565	0.26	0.203
AF057043	ACX2	0.415	0	-0.053	0.114	0.296	0.242	0.09
U40856	AIG1	0.096	-0.106	-0.027	-0.026	-0.005	-0.052	0.054
U40857	AIG2	0.311	0.14	0.257	0.261	0.158	0.056	-0.049
AF123253	AIM1	-0.04	0.002	-0.202	-0.04	0.077	0.081	0.088
X92510	AOS	0.473	0.56	0.914	0.625	0.375	0.387	0.019
D14442	APX1	0.037	-0.079	0.053	0.117	0.187	0.176	0.201
X66017	AR2	0.099	-0.435	-0.16	-0.059	0.065	-0.101	-0.123
U38916	ARAB1	0.082	0	-0.308	-0.064	0.096	0.047	0.043
M92353	ASA1	0.277	0.117	0.572	0.419	0.357	0.077	-0.085
L22585	ASB	0.103	0.116	0.324	0.309	0.304	-0.083	-0.084
X99793	AWI31	0.09	0.026	0.14	0.478	0.665	0.344	0.185
M84706	b-TUB	0.107	-0.024	0.032	0	0.047	-0.157	-0.076
X64460	CAB	-0.003	-0.021	-0.021	-0.038	-0.002	-0.005	0.051

Fig. 1. A high-level description of data analysis for motif finding from microarray data. The analysis starts from scanned microarray images. After proper quantification and preprocessing, the data are available for clustering in the form of a data matrix. Clustering then determines clusters of potentially coregulated genes. Focusing on a cluster of genes of interest, motif finding analyzes the sequences of the control regions of the genes in the cluster. A number of true motifs are present in those sequences, but they are unknown. Motif finding analyzes those sequences for statistically overrepresented DNA patterns. Finally, candidate motifs are returned by the motif-finding algorithm and are available for further biological evaluation.

used in cDNA microarrays: a reference and a test sample (e.g., normal versus malignant tissue). A pair of cDNA samples is independently copied from the corresponding mRNA populations with the reverse transcriptase enzyme and labeled using distinct fluorescent molecules (green and red). These labeled cDNA samples are then 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 at both fluorescence wavelengths and calculating the ratios (here, only relative expression levels are obtained). A cDNA microarray is therefore a differential technique, which intrinsically normalizes for part of the experimental noise. An overview of the procedure that can be followed with cDNA microarrays is given in Fig. 2.

GeneChip oligonucleotide arrays (Affymetrix, Inc., Santa Clara, CA) [30] are high-density arrays of oligonucleotides synthesized using a photolithograpic technology similar to microchip technology. The synthesis uses *in situ* light-directed chemistry to build up hundreds of thousands of different oligonucleotide probes (25 nucleotides long). 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 estimation of cross-hybridization and significantly decrease the number of false positives. With this technology, absolute expression levels are obtained (no ratios).

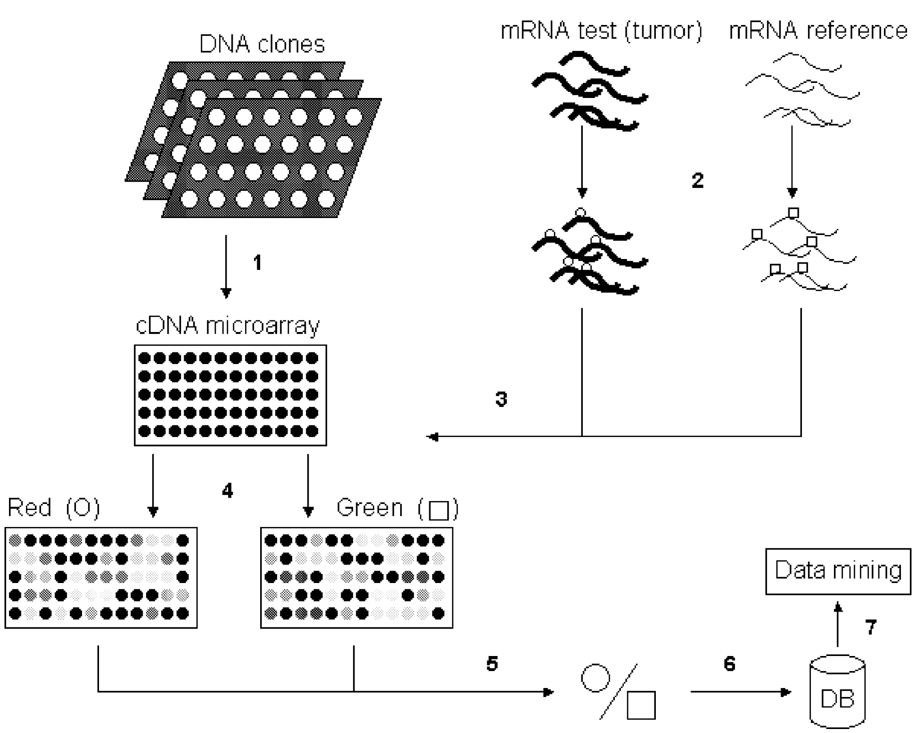


Fig. 2. Schematic overview of an experiment with a cDNA microarray. (1) Spotting of the presynthesized DNA probes (derived from the genes to be studied) on the glass slide. These probes are the purified products from polymerase chain reaction amplification of the associated DNA clones. (2) Labeling (by reverse transcriptase) of the total mRNA of the test sample (red channel ○) and reference sample (green channel □). (3) Pooling of the two samples and hybridization. (4) Readout 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 divided by the intensity in the green channel). (6) Storage of results in a database. (7) Data mining.

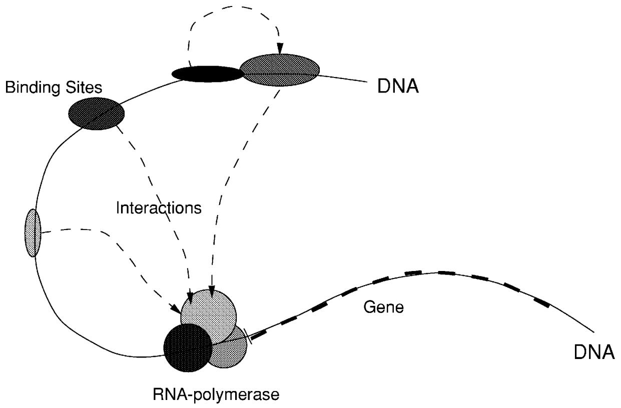


Fig. 3. Initiation of the transcription process by the association of the complex of transcription factors (gene regulatory proteins), the RNA polymerase, and the promoter region of a gene.

III. Introduction to Transcriptional Regulation

In this section, we present concisely the main concepts from biology relevant to our discussion of motif finding in DNA sequences.

A. *Structure of Genes*

Genes are segments of DNA that encode for proteins through the intermediate action of mRNA. A gene and the genomic region surrounding it consists of a transcribed sequence, which is converted into an mRNA transcript, and of various *untranscribed* sequences. The mRNA consists of a coding sequence that is translated into a protein and of several *untranslated* regions (UTRs). The untranscribed sequences and the 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. Moreover, the region upstream of the transcription start contains many binding sites for transcription factors that act as *activators* and *repressors* of gene expression (although some transcription factors can bind outside this region).

B. *Transcription*

Transcription means the assembly of ribonucleotides into a single strand of mRNA. The sequence of this strand of mRNA is dictated by the order of the nucleotides in the transcribed part of the gene. 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. In Fig. 3, the initiation of the transcription process is shown.

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 complemen-

Table 1
Databases on Transcriptional Regulation

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/

tary 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 nucleus, and translated into a protein.

C. Transcription Factors

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 silencers), 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 activator.

D. Regulatory Elements on the Web

Regulatory elements play a central role in the study of biological sequences and many databases are available to explore known regulatory elements. Table 1 gives 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.

IV. Clustering of Gene Expression Profiles

Using microarrays, 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. Also, as discussed in Section V, coexpressed genes are more likely to be coregulated.

Cluster analysis in a collection of gene expression profiles aims at identifying subgroups (i.e., clusters) of such coexpressed genes, which thus have a higher probability of participating in the same pathway. Note that cluster analysis of expression data is only a first rudimentary step preceding further analysis, which includes motif finding [49], [41], [54], functional annotation, genetic network inference, and class discovery in the microarray experiments or samples themselves [5], [17]. 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 still far away.

The first generation of cluster algorithms (e.g., direct visual inspection [9], K-means [51], self-organizing maps (SOMs) [44], hierarchical clustering [13]) applied to gene expression profiles were mostly developed outside biological research. Although it is possible to obtain biologically meaningful results with these algorithms, some of their characteristics often complicate their use for clustering expression data [43]. They require, for example, the predefinition of one or more user-defined parameters that are hard to estimate by a biologist (e.g., the predefinition of the number of clusters in K-means and SOM—this number is almost impossible to predict in advance). Moreover, changing these parameter settings will often have a strong impact on the final result. These methods therefore need extensive parameter fine-tuning, which means that a comparison of the results with different parameter settings is almost always necessary—with the additional difficulty that comparing the quality of the different clustering results is hard. Another problem is that first-generation clustering algorithms often force every data point into a cluster. In general, a considerable number of genes included in the microarray experiment do not contribute to the biological process studied, and these genes will therefore lack coexpression with other genes (they will have seemingly constant or even random expression profiles). Including these genes in one of the clusters will contaminate their content (these genes represent noise) and make these clusters less suitable for further analysis. Finally, the computational and memory complexity of some of these algorithms often limit the number of expression profiles that can be analyzed at once. Considering the nature of our data sets (number of expression profiles often running up into the tens of thousands), this constraint is often unacceptable.

Recently, many new clustering algorithms have started to tackle some of the limitations of earlier methods (e.g., the self-organizing tree algorithm [SOTA] [18], quality-based clustering [21], adaptive quality-based clustering [10], model-based clustering [15], [58], simulated annealing [35], gene shaving [17], the cluster affinity search technique [CAST] [5]). Also, some procedures were developed that could help biologists to estimate some of the parameters needed for the first generation of algorithms (such as the number of clusters present in the data [15], [35], [58]). We will discuss a selection of these clustering algorithms in the following sections.

An important problem that arises when performing cluster analysis of gene expression profiles is the preprocessing of the data. A correct preprocessing strategy is almost as important as the cluster analysis itself. First, it is necessary to normalize the hybridization intensities within a single array experiment. In a two-channel cDNA microarray experiment, for example, normalization adjusts for differences in labeling, detection efficiency, and in the quantity of initial RNA within the two channels [23]. Normalization is necessary before one can compare the results from different microarray experiments. Second, transformation of the data using a nonlinear function (often the logarithm is used, especially for two-channel cDNA microarray experiments where the values are expression ratios) can be useful [23]. Third, expression data often contain numerous missing values, and many clustering algorithms are unable to deal with them [52]. It is therefore imperative either to use appropriate procedures that can estimate and replace these missing values or to adapt existing clustering algorithms, enabling them to handle missing values directly (without actually replacing them [10], [24]). Fourth, it is common to (crudely) filter the gene expression profiles (removing the profiles that do not satisfy some simple criteria) before proceeding with the actual clustering [13]. A fifth preprocessing step is standardization or rescaling of the gene expression profiles (e.g., multiplying every expression vector with a scale factor so that its length is one [23]). This makes sense because the aim is to cluster gene expression profiles with the same relative behavior (expression levels go up and down at the same time) and not only the ones with the same absolute behavior. Some of these preprocessing steps will be discussed in more detail in the following sections.

Validation is another key issue when clustering gene expression profiles. The biologist using the algorithm is of course mainly interested in the biological relevance of these clusters and wants to use the results to discover new biological knowledge. This means that we need methods to (biologically and statistically) validate and objectively compare the results produced by new and existing clustering algorithms. Some methods for cluster validation have recently emerged (Figure of merit (FOM) [58], (adjusted) Rand index [60], and looking for enrichment of functional categories [46]), and will be discussed below. Note that no real benchmark data set exists to unambiguously validate novel algorithms (however, the measurements produced by Cho *et al.* [9] on the cell cycle of yeast are often used for this purpose).

Table 2
Availability of Clustering Algorithms

Package	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`
Adaptive Quality-based	`www.esat.kuleuven.ac.be/~dna/BioI/Software.html`

A. Clustering Algorithms

As stated at the beginning of this section, many clustering methods (first- and second-generation algorithms) are available; we will discuss some of the more important ones in more detail below.

1) First-Generation Algorithms: Notwithstanding some of the disadvantages of these early methods, it must be noted that many good implementations (see Table 2) of these algorithms exist ready for use by biologists (which is not always the case with the newer methods).

a) Hierarchical clustering: Hierarchical clustering [13], [23], [43] is the most widely used method for clustering gene expression data and can be seen as the *de facto* standard. Hierarchical clustering has the advantage that the results can be nicely visualized (see Fig. 4). Two approaches are possible: a top-down approach (divisive clustering; see [1]) and a bottom-up approach (agglomerative clustering; see [13]). The latter is the most commonly used and will be discussed here. In the agglomerative approach, each gene expression profile is initially assigned to a single cluster. The distance 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. Finally, 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

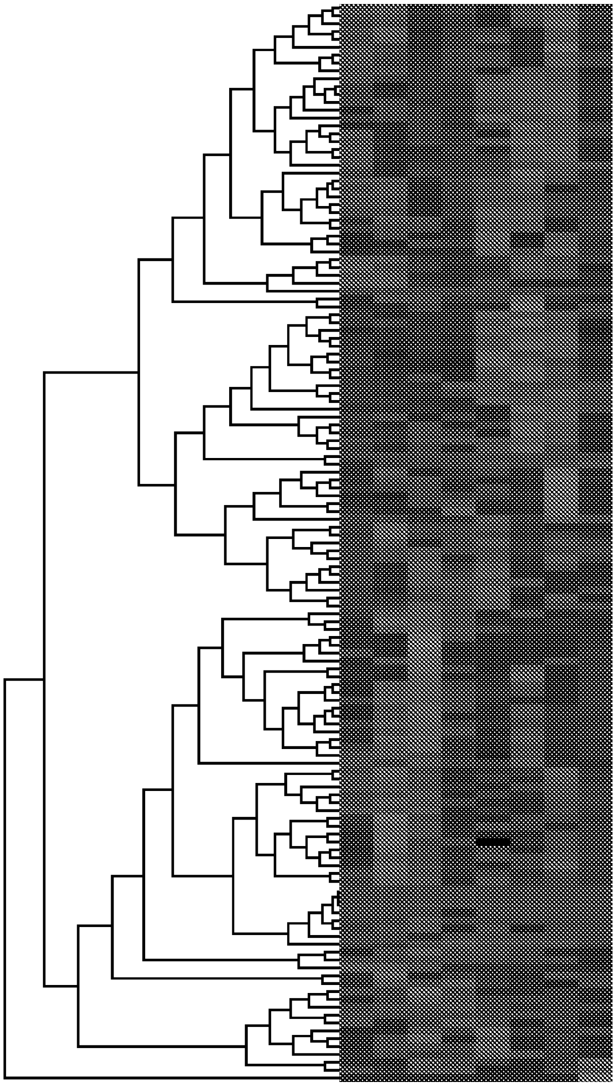


Fig. 4. Typical result from an analysis using hierarchical clustering using 137 expression profiles of dimension 8. The left side of the figure represents the tree structure. The terminal branches of this tree are linked with the individual genes and the height of all the branches is proportional to the pairwise distance between the clusters. The right side of the figure (also called a heat map) corresponds to the expression matrix where each row represents an expression profile, each column a microarray experiment, and the individual values are represented on a color (green to red) or gray scale.

memory complexity of hierarchical clustering is quadratic in the number of gene expression profiles, which can be a problem when considering the current size of the data sets.

b) K-means clustering: K-means clustering [46], [51] results in a partitioning of the data (every gene expression profile belongs to exactly one cluster) using a predefined number K of partitions or clusters. K-means starts by dividing up all the gene expression profiles among N initial clusters. Iteratively, the center (which is nothing more than the average expression vector) of each cluster is calculated, followed by a reassignment of the gene expression vectors to the cluster with the closest cluster center. Convergence is reached when the cluster centers remain stationary.

Note that the predefinition of the number of clusters by the user is also rather arbitrary (it is difficult to predict the number of clusters in advance). In practice, this makes it necessary to use a trial-and-error approach where a comparison and biological validation of several runs of the algorithm with different parameter settings are necessary.

c) Self-organizing maps: In SOMs [26], [44], the user has to predefine a topology or geometry of nodes (e.g., a two-dimensional grid—one node for each cluster), which again is not straightforward. These nodes are then mapped into the gene expression space, initially at random and iteratively adjusted. In each iteration, a gene expression profile is randomly picked, and the node that maps closest to it is selected. This selected node (in gene expression space) is then moved into the direction of the selected expression profile. The other nodes are also moved into the direction of the selected expression profile but to an extent proportional to the distance from the selected node in the initial two-dimensional node topology.

2) Second-Generation Algorithms: In this section we describe several of the newer clustering methods that have specifically been designed to cluster gene expression profiles.

a) Self-organizing tree algorithm: The SOTA [18] combines both self-organizing maps and divisive hierarchical clustering. The topology or node geometry here takes the form of a dynamic binary tree. Like SOMs, the gene expression profiles are sequentially and iteratively presented to the terminal nodes (located at the base of the tree—these nodes are also called cells). Subsequently, the gene expression profiles are associated with the cell that maps closest to it, and the mapping of this cell plus its neighboring nodes are updated (moved into the direction of the expression profile). The presentation of the gene expression profiles to the cells continues until convergence. After convergence the cell containing the most variable population of expression profiles (variation is defined here by the maximal distance between two profiles that are associated with the same cell) is split into two sister cells (causing the binary tree to grow), whereafter the entire process is restarted. The algorithm stops (the tree stops growing) when a threshold of variability is reached for each cell, which involves the actual construction of a randomized data set and the calculation of the distances between all possible pairs of randomized expression profiles.

The approach described in [18] has some properties that make it potentially useful for clustering gene expression profiles.

1) The clustering procedure itself is linear in the number of gene expression profiles (compare this with the quadratic complexity of standard hierarchical clustering).
2) The number of clusters does not have to be known in advance. Moreover, the procedure provides for a statistical procedure to stop growing the tree. Therefore, the user is freed from choosing an (arbitrary) level where the tree has to be cut (like in standard hierarchical clustering).

In our opinion, however, this method also has some disadvantages:

1) The procedure for finding the threshold of variability is time-consuming. The entire process described in [18] is in fact quadratic in the number of gene expression profiles.
2) No biological validation was provided showing that this algorithm indeed produces biologically relevant results.

b) Model-based clustering: Model-based clustering [14], [15], [58] is an approach that is not really new and has already been used in the past for other applications outside bioinformatics. In this sense it is not really a true second-generation algorithm. However its potential use for cluster analysis of gene expression profiles has been proposed only recently; thus, we treat it in this text as a second-generation method.

Model-based clustering assumes that the data are generated by a finite mixture of underlying probability distributions, where each distribution represents one cluster. Usually, multivariate normal distributions are used for these.

The covariance matrix for each cluster can be represented by its eigenvalues decomposition, which controls the orientation, shape, and volume of each cluster. Note that simpler forms for the covariance structure can be used (e.g., by having some of the parameters take the same values across clusters), thereby decreasing the number of parameters that have to be estimated but also decreasing the model flexibility (capacity to model more complex data structures).

First, the parameters of the model are estimated with an EM algorithm using a fixed value for the number of clusters and a fixed covariance structure. This parameter estimation is then repeated for different numbers of clusters and different covariance structures. The result of the first step is thus a collection of different models fitted to the data and all having a specific number of clusters and a specific covariance structure. Second, the best model in this group of models is selected (i.e., the most appropriate number of clusters and a covariance structure is chosen here). This model selection step involves the calculation of the Bayesian information criterion [42] for each model, which is not further discussed here.

Yeung *et al.* [58] reported good results using their MCLUST software [14] on several synthetic data sets and real expression data sets. They claimed that the performance of MCLUST on real expression data was at least as good as could be achieved with a heuristic cluster algorithm (CAST [5], not discussed here).

c) Quality-based clustering: In [21], a clustering algorithm (called QT_Clust) is described that produces clusters that have a quality guarantee which ensures that all members of a cluster should be coexpressed with all other members of this cluster. The quality guarantee itself is defined as a fixed and user-defined threshold for the maximal distance between two points within a cluster. Briefly said, QT_Clust is a greedy procedure that finds one cluster at a time satisfying the quality guarantee and containing a maximum number of expression profiles. The algorithm stops when the number of points in the largest remaining cluster falls below a prespecified threshold. Note that this stop criterion implies that the algorithm will terminate before all expression profiles are assigned to a cluster.

This approach was designed with cluster analysis of expression data in mind and has some properties that could make it useful for this task.

1) By using a stringent quality guarantee, it is possible to find clusters with tightly related expression profiles (containing highly coexpressed genes). These clusters might therefore be good “seeds” for further analysis.
2) Genes not really coexpressed with other members of the data set are not included in any of the clusters.

There are, however, also some disadvantages.

1) The quality guarantee of the clusters is a user-defined parameter that is hard to estimate and too arbitrary. This method is therefore hard for biologists to use in practice, and extensive parameter fine-tuning is necessary.
2) This algorithm produces clusters all having the same fixed diameter not optimally adapted to the local data structure.
3) The computational complexity is quadratic in the number of expression profiles.

Furthermore, no ready-to-use implementation is available.

d) Adaptive quality-based clustering: Adaptive quality-based clustering [10] was developed starting from the principles of quality-based clustering (finding clusters with a quality guarantee containing a maximal number of members) but was designed to circumvent some of its disadvantages.

Adaptive quality-based clustering is a heuristic iterative two-step approach. In the first step, a quality-based approach is followed. 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. Contrary to the original method [21], 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, that remains fixed—is re-estimated so that the genes belonging to the cluster are, in a statistical sense, significantly coexpressed (higher coexpression that 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 EM algorithm. Note that, the computational complexity of this step is negligible with respect to the computational complexity of the first step.

Finally, steps 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 adaptive quality-based clustering approach has some additional advantages over standard quality-based clustering that make it suited for the analysis of gene expression profiles.

1) In adaptive quality-based clustering, the user has to specify a significance level S. This parameter has a strict statistical meaning and is therefore much less arbitrary (contrary to the quality guarantee used in standard quality-based clustering). It can be chosen independently of a specific data set or cluster and it allows for a meaningful default value (95%) that in general gives good results. This makes this approach user friendly without the need for extensive parameter fine-tuning.
2) Adaptive quality-based clustering produces clusters adapted to the local data structure (the clusters do not have the same radius).
3) The computational complexity of the algorithm is linear in the number of expression profiles.
4) Adaptive quality-based clustering was extensively biologically validated.

However, the method also has some limitations.

1) It is a heuristic approach not proven to converge in every situation.
2) Because of the model structure used in the second step, some additional constraints have to be imposed. They include the fact that only standardized expression profiles are allowed and that the method has to be used in combination with the Euclidean distance and cannot directly be extended to other distance measures.

As a conclusion to this overview of clustering algorithms, Table 2 gives an overview of some clustering methods for which the software is available for download or can be accessed online.

B. *Preprocessing of the Data*

As stated at the start of this section, clustering also implies performing some additional operations on the data, preparing them for the actual cluster analysis. Below, we will discuss some of the most common preprocessing steps.

1) Normalization: The first step is the normalization of the hybridization intensities within a single array experiment. In a two-channel cDNA microarray experiment, several sources of noise (such as differences in labeling, in detection efficiency, and in the quantity of initial RNA within the two channels) create systematic sources of biases. The biases can be computed and removed to correct the data. Since many sources can be considered and since they can be estimated and corrected in a variety of ways, many different normalization procedures exist. We therefore do not cover this topic further here; see [23] for more details.

2) Nonlinear Transformations: It is common practice to pass expression values through a nonlinear function. Often the logarithm is used for this nonlinear function. This is especially suited for dealing with expression ratios (coming from two-channel cDNA microarray experiments, using a test and reference sample), since expression ratios are not symmetrical [23]. Upregulated genes have expression ratios between one and infinity, while downregulated genes have expression ratios squashed between one and zero. Taking the logarithms of these expression ratios results in symmetry between expression values of up- and downregulated genes.

3) Missing Value Replacement: Microarray experiments often contain missing values (measurements absent because of technical reasons). The inability of many cluster algorithms to handle such missing values necessitates the replacement of these values. Simple replacements such as a replacement by zero or by the average of the expression profile often disrupt these profiles. Indeed, replacement by average values relies on the unrealistic assumption that all expression values are similar across different experimental conditions. Because of an erroneous missing value replacement, genes containing a high number of missing values can be assigned to the wrong cluster. More advanced techniques of missing value replacement (which use the K-nearest neighbor method or the singular value decomposition) have been described [52] that take advantage of the rich information provided by the expression patterns of other genes in the data set.

Finally, note that some implementations of algorithms use only the measured values to derive the clusters and as such obviate the need for missing value replacement [10].

4) Filtering: As stated in the overview section, a set of microarray experiments, generating gene expression profiles, frequently contain a considerable number of genes that do not really contribute to the biological process that is being studied. The expression values of these profiles often show little variation over the different experiments (they are called constitutive with respect to the biological process studied). Moreover, these constitutive genes will have seemingly random and meaningless profiles after standardization (division by a small standard deviation results in noise inflation), which is also a common preprocessing step (see further). Another problem comes from highly unreliable expression profiles containing many missing values.

The quality of the clusters would significantly degrade if these data were passed to the clustering algorithms as such. Filtering [13] removes gene expression profiles from the data set that do not satisfy some simple criteria. Commonly used criteria include a minimum threshold for the standard deviation of the expression values in a profile (removal of constitutive genes) and a threshold on the maximum percentage of missing values.

5) Standardization or Rescaling: Biologists are mainly interested in grouping gene expression profiles that have the same relative behavior; i.e., genes that are up- and downregulated together. Genes showing the same relative behavior but with diverging absolute behavior (e.g., gene expression profiles with a different baseline or a different amplitude but going up and down at the same time) will have a relatively high Euclidean distance. Cluster algorithms based on this distance measure will therefore wrongfully assign these genes to different clusters.

This effect can largely be prevented by applying standardization or rescaling to the gene expression profiles to have zero mean and unit standard deviation. Gene expression profiles showing the same relative behavior will have a small(er) Euclidean distance after rescaling [23].

C. Cluster Validation

As mentioned before, clustering will produce different results. Even random data often produce clusters depending on the specific choice of preprocessing, algorithm, and distance measure. Therefore, validation of the relevance of the cluster results is of utmost importance. Validation can be either statistical or biological. Statistical cluster validation can be done by assessing cluster coherence, by examining the predictive power of the clusters, or by testing the robustness of a cluster result against the addition of noise.

Alternatively, the relevance of a cluster result can be assessed by a biological validation. Of course it is hard, not to say impossible, to select the best cluster output, since "the biologically best" solution will be known only if the biological system studied is completely characterized. Although some biological systems have been described extensively, no such completely characterized benchmark system is now available. A common method to biologically validate cluster outputs is to search for enrichment of functional categories within a cluster. Detection of regulatory motifs (see [46]) is also an appropriate biological validation of the cluster results. Some of the recent methodologies described in literature to validate cluster results will be highlighted in the following.

1) Testing Cluster Coherence: Based on biological intuition, a cluster result can be considered reliable if the within-cluster distance is small (i.e., all genes retained are tightly co-expressed) and the cluster has an average profile well delineated from the remainder of the data set (maximal intercluster distance). Such criteria can be formalized in several ways, such as the sum-of-squares criterion of K-means [51], silhouette coefficients [24], or Dunn's validity index [2]. These can be used as stand-alone statistics to mutually compare cluster results. They can also be used as an inherent part of cluster algorithms, if their value is optimized during the clustering process.

2) Figure of Merit: FOM [59] is a simple quantitative data-driven methodology that allows comparisons between outputs of different clustering algorithms. The methodology is related to the jackknife and leave-one-out cross-validation. The method goes as follows. The clustering algorithm (for the genes) is applied to all experimental conditions (the data variables) except for one left-out condition. If the algorithm performs well, we expect that if we look at the genes from a given cluster, their values for the left-out condition will be highly coherent. Therefore, we compute the FOM for a clustering result by summing, for the left-out condition, the squares of the deviations of each gene relative to the mean of the genes in its cluster *for this condition*. The FOM measures the within-cluster similarity of the expression values of the removed experiment and therefore reflects the predictive power of the clustering. It is expected that removing one experiment from the data should not interfere with the cluster output if the output is robust. For cluster validation, each condition is subsequently used as a validation condition, and the aggregate FOM over all conditions is used to compare cluster algorithms.

3) Sensitivity Analysis: Gene expression levels are the superposition of real biological signals and experimental errors. A way to assign confidence to a cluster membership of a gene consists in creating new *in silico* replicas of the microarray data by adding to the original data a small amount of artificial noise (similar to the experimental noise in the data) and clustering the data of those replicas. If the biological signal is stronger than the experimental noise in the measurements of a particular gene, adding small artificial variations (in the range of the experimental noise) to the expression profile of this gene will not drastically influence its overall profile and therefore will not affect its cluster membership. In this case, the cluster membership of that particular gene is robust with respect to sensitivity analysis, and a reliable confidence can be assigned to the clustering result of that gene. However, for genes with low signal-to-noise ratios, the outcome of the clustering result will be more sensitive to adding artificial noise. Through some robustness statistic [6], sensitivity analysis lets us detect which clusters are robust within the range of experimental noise and therefore trustworthy for further analysis.

The main issue in this method is to choose the noise level for sensitivity analysis. Bittner *et al.* [6] perturb the data by adding random Gaussian noise with zero mean and a standard deviation that is estimated as the median standard deviation for the log-ratios for all genes across the experiments. This implicitly assumes that ratios are unbiased estimators of relative expression, yet reality shows often otherwise.

The bootstrap analysis methods described by Kerr *et al.* [25] to identify statistically significant expressed genes or to assess the reliability of a clustering result offers a more statistically founded basis for sensitivity analysis and overcomes some of the problems of the method described by Bittner *et al.* [6]. Bootstrap analysis uses the residual values of a linear analysis of variance (ANOVA) model as an estimate of the measurement error. By using an ANOVA model, nonconsistent measurement errors can be separated from variations caused by alterations in relative expression or by consistent variations in the data set. These errors are assumed to be independent with mean zero and constant variance σ^2 but no explicit assumption on their distribution is made. The residuals are subsequently used to generate new replicates of the data set by bootstrapping (adding residual noise to estimated values).

4) Use of Different Algorithms: Just as clustering results are sensitive to adding noise, they are sensitive to the choice of clustering algorithm and to the specific parameter settings of a particular algorithm. Many clustering algorithms are available, each of them with different underlying statistics and inherent assumptions about the data. The best way to infer biological knowledge from a clustering experiment is to use different algorithms with different parameter settings. Clusters detected by most algorithms will reflect the pronounced signals in the data set. Again statistics similar to that of Bittner *et al.* [6] are used to perform these comparisons.

Biologists tend to prefer algorithms with a deterministic output, since this gives the illusion that what they find is "right." However, nondeterministic algorithms offer an advantage for cluster validation, since their use implicitly includes a form of sensitivity analysis.

5) Enrichment of Functional Categories: One way to *biologically* validate results from clustering algorithms is to compare the gene clusters with existing functional classification schemes. In such schemes, genes are allocated to one or more functional categories [15], [46] representing their biochemical properties, biological roles, and so on. Finding clusters that have been significantly enriched for genes with similar function is proof that a specific clustering technique produces biologically relevant results.

As stated in the overview section, the results of the expression profiling experiment of Cho *et al.* [9] studying the cell development cycle of yeast in a synchronized culture is often used as a benchmark data set. It contains 6220 expression profiles taken over 17 time points (measurements over 10-min intervals, covering nearly two cell cycles; see also http://cellcycle-www.stanford.edu). One of the reasons that these data are so frequently used as benchmark data for the validation of new clustering algorithms is because of the striking cyclic expression patterns and because the majority of the genes included in the data have been functionally classified [38] (MIPS database; see http://mips.gsf.de/proj/yeast/catalogues/funcat/index.html), making it possible to biologically validate the results.

Assume that a certain clustering method finds a set of clusters in the Cho *et al.* data. We could objectively look for functionally enriched clusters as follows: Suppose that one of the clusters has n genes where k genes belong to a certain functional category in the MIPS database, and suppose that this functional category in its turn contains f genes in total. Also suppose that the total data set contains g genes (in the case of Cho *et al.* [9], g would be 6220). Using the cumulative hypergeometric probability distribution, we could measure the degree of enrichment by calculating the probability or P-value of finding by chance at least k genes in this specific cluster of n genes from this specific functional category that contains f genes out of the whole g annotated genes

$$P = 1 - \sum_{i=0}^{k-1} \frac{\binom{f}{i}\binom{g-f}{n-i}}{\binom{g}{n}} = \sum_{i=k}^{\min(n,f)} \frac{\binom{f}{i}\binom{g-f}{n-i}}{\binom{g}{n}}.$$

These P-values can be calculated for each functional category in each cluster. Since there are about 200 functional categories in the MIPS database, only clusters where the P-value is smaller than 0.0003 for a certain functional category, are said to be significantly enriched (level of significance 0.05). Note that these P-values can also be used to compare the results from functionally matching clusters identified by two different clustering algorithms on the same data.

As an example of cluster validation and as an illustration of our adaptive quality-based clustering, we compared K-means with adaptive quality-based clustering on the Cho *et al.* data. We performed adaptive quality-based clustering [10] using the default setting for the significance level (95%) and compared these results with those for K-means reported by Tavazoie *et al.* [46]. As discussed above, the genes in each cluster have been mapped to the functional categories in the MIPS database and the negative base-10 logarithm of the hypergeometric P-values (representing the degree of enrichment) have been calculated for each functional category in each cluster. In Table 3, we compare enrichment in functional categories for the three most significant clusters found by each algorithm. To compare K-means and adaptive quality-based clustering, we identified functionally matching clusters manually. The first column ("Cl. #, AC") gives the index of the cluster identified by adaptive quality-based clustering. The second column ("Cl. #, KM") gives the index of the matching cluster for K-means as described in Tavazoie *et al.* [46]. The third column ("# Gene, AC") gives the number of genes of in the cluster for adaptive quality-based clustering. The fourth column ("# Gene, KM") gives the number of genes of in the cluster for K-means. The fifth column ("MIPS functional category") lists the significant functional categories for the two functionally matching clusters. The sixth column ("In cat., AC") gives the number of genes of the corresponding functional category in the cluster for adaptive quality-based clustering. The seventh column ("In cat., KM") gives the number of genes of the corresponding functional category in the cluster for K-means. The eighth column ("P-val., AC") gives the negative logarithm in base 10 of the hypergeometric P-value for adaptive quality-based clustering. The ninth column ("P-val., KM") gives the negative logarithm in base 10 of the hypergeometric P-value for K-means (NR = not reported). Although we do not claim to draw any conclusion from this single table, we observe that the enrichment in functional categories in stronger for adaptive quality-based clustering than for K-means. This result and several others are discussed extensively in [10].

V. Searching for Common Binding Sites of Coregulated Genes

In the previous section, we described the basic ideas underlying several clustering techniques together with their advantages and shortcomings. We also discussed the preprocessing steps necessary to make microarray data suitable for clustering. Finally, we described methodologies for validating the result of a clustering algorithm. We can now make the transition toward looking at the groups of genes generated by clustering and study the sequences of these genes to detect motifs that control their expression (and cause them to cluster together in the first place).

Given a cluster of genes with highly similar expression profiles, the next step in the analysis is the search for the mechanism responsible for their coordinated behavior. We basically assume that coexpression frequently arises from transcriptional coregulation. As coregulated genes are known to share some similarities in their regulatory

Table 3
Comparison of Functional Enrichment for the Yeast Cell Cycle Data of Cho *et al.* Using Adaptive-Quality Based Clustering and K-Means

Cl. #	Cl. #	# Gene	# Gene	MIPS functional category	In cat.	In cat.	P-val.	P-val.
AC	KM	AC	KM		AC	KM	AC	KM
1	1	302	164	ribosomal proteins	101	64	80	54
				organization of cytoplasm	146	79	77	39
				protein synthesis	119	NR	74	NR
				cellular organization	211	NR	34	NR
				translation	17	NR	9	NR
				organization of chromosome structure	4	7	1	4
2	4	315	170	mitochondrial organization	62	32	18	10
				energy	35	NR	8	NR
				proteolysis	25	NR	7	NR
				respiration	16	10	6	5
				ribosomal proteins	24	NR	4	NR
				protein synthesis	33	NR	4	NR
				protein destination	49	NR	4	NR
5	2	98	186	DNA synthesis and replication	20	23	18	16
				cell growth and division, DNA synthesis	48	NR	17	NR
				recombination and DNA repair	12	11	8	5
				nuclear organization	32	40	8	4
				cell-cycle control and mitosis	20	30	7	8

mechanism, possibly at transcriptional level, their promoter regions might contain some common motifs that are binding sites for transcription regulators. A sensible approach to detect these regulatory elements is to search for statistically overrepresented motifs in the promoter region of such a set of coexpressed genes [7], [40], [41], [46], [61].

In this section, we describe the two major classes of methods to search for overrepresented motifs. The first class of methods are string-based methods that mostly rely on counting and comparing oligonucleotide frequencies. The second class of methods is based on probabilistic sequence models. For these methods, the model parameters are estimated using maximum-likelihood (ML) or Bayesian inference.

Table 4 gives an overview of some of the methods described in the section that can be accessed online or where the software is available for download.

In this section, we begin with a discussion of the important facts that we can learn by looking at a realistic biological example. Prior knowledge about the biology of the problem at hand will facilitate the definition of a good model. Next, we discuss the different string-based methods, starting from a simple statistical model and gradually refining the models and the statistics to handle more complex configurations. Then we switch to the probabilistic methods and introduce EM for motif finding. In Section VI, we discuss Gibbs

Table 4
Availability of Motif-Finding Algorithms

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	`www.esat.kuleuven.ac.be/~dna/BioI/Software.html`

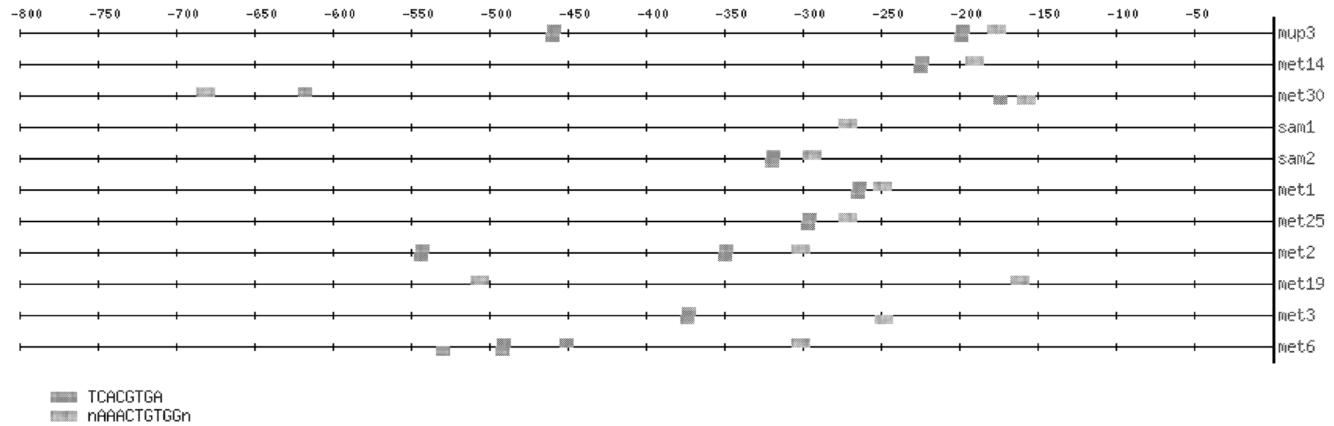


Fig. 5. Schematic representation of the upstream region of a set of coregulated genes. Several possible combinations of the two motifs are present: 1) motifs occur on both strands; 2) some sequences contain one or more copies of the two binding sites; or 3) some sequences do not contain a copy of a motif.

sampling for motif finding. This method is less well known than EM, yet it is more effective for motif finding in DNA sequences. We therefore explain the basic ideas underlying this method and overview the extensions, including our own work, that are necessary for the practical use of this method.

A. Realistic Sequence Models

To search for common motifs in sets of upstream sequences, a realistic model should be proposed. Simple motif models are designed to search for conserved motifs of fixed length, while more complex models will incorporate variability like insertions and deletions into the motif model. But not only the model of the binding site itself is important; the model of the background sequence in which the motif is hidden and the number of times a motif occurs in the sequence also play important roles.

To illustrate this complexity, we look at an example in baker's yeast (*Saccharomyces cerevisiae*). Fig. 5 gives a schematic representation of the upstream sequences from 11 genes in *S. cerevisiae* which are regulated by the Cbfl-Met4p-Met28p complex and Met31p or Met32p in response to methionine [53]. The consensus (which is the dominant DNA pattern describing the motif) for these binding sites is given by TCACGTG for the Cbfl-Met4p-Met28p complex and AAAACTGTGG for Met31p or Met32p [53]. A logo representation of the aligned instances of the two binding sites is shown in Fig. 6. This logo represents the frequency of each nucleotide at each position; the relative size of the symbol represents the relative frequency of each base at this position, and the total height of the symbol represents the magnitude of the deviation from a uniform (noninformative) distribution. Fig. 6 shows the locations of the two binding sites in the region 800 bp upstream of translation start. It is clear from this picture that there are several possible configurations of the two binding sites present in this data set. First of all, it is important to note that motifs can occur on both strands. Transcription factors indeed bind directly on the double-stranded DNA; therefore, motif detection software should take this fact into account. Second, sequences could have either zero, one, or multiple copies of a motif. This example gives an indication of the kind of data that come with a realistic biological data set.

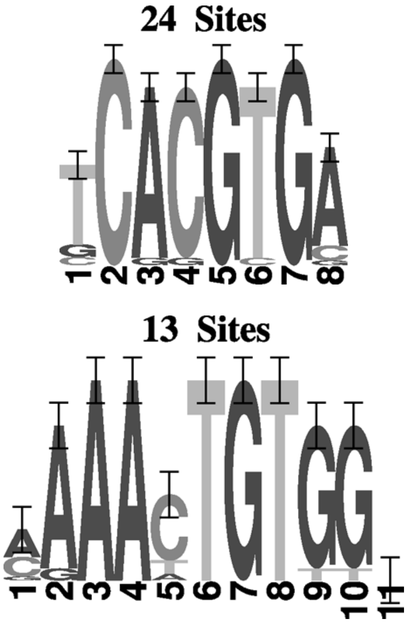


Fig. 6. Logo representation of the transcription factor binding sites present in the MET data set.

1) Palindromic Motifs: *Palindromic* motifs are a special type of transcription factor binding site from a computational point of view. This kind of motif is a subsequence that is exactly the same as its own reverse complement. The first motif in Fig. 5 is an example of a motif with a palindromic core.

2) Gapped Motifs: A second class of special motifs are *gapped* motifs or spaced dyads. Such a motif consists 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 con-

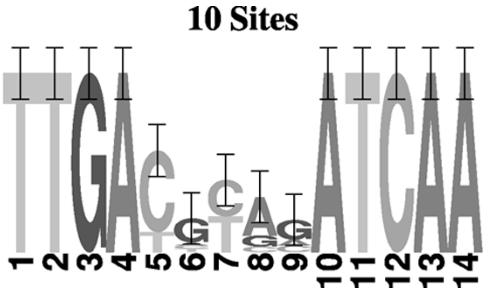


Fig. 7. Logo representation of the FNR binding site. The motif consists of two conserved parts, `TTGAy` and `ATCAA`, separated by a spacer of length 4.

served but are typically rather small (3–5 bp). These two contact points are separated by a nonconserved gap or spacer. This gap is mostly of fixed length but might be slightly variable. Fig. 7 shows a logo representation of the FNR binding site in bacteria.

3) Cooperatively Binding Factors and Modules: Currently another important research topic is the search for cooperatively binding factors [56]. 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. If we translate this to the motif-finding problem, we could 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.

B. Oligonucleotide Frequency Analysis

The most intuitive approach to extract a consensus pattern for a binding site is a string-based approach, where typically overrepresentation is measured by exhaustive enumeration of all oligonucleotides. The observed number of occurrences of a given motif is compared with the expected number of occurrences. The expected number of occurrences and the statistical significance of a motif can be estimated in many ways. In this section we give an overview of the different methods.

A basic version of the enumeration methods was implemented by van Helden *et al.* [53]. They presented a simple and fast method for the identification of DNA binding sites in the upstream regions from families of coregulated genes in *S. cerevisiae*. This method searches for short motifs 5–6 bp long. First, for each oligonucleotide of a given length, we compute the expected frequency of the motif from all the noncoding, upstream regions in the genome of interest. Based on this frequency table, we compute the expected number of occurrences of a given oligonucleotide in a specific set of sequences. Next, the expected number of occurrences is compared with the actual counted number of occurrences in the data set. Finally, we compute a significance coefficient that takes into account the distinct number of oligonucleotides. A binomial statistic is appropriate in the case where there are nonoverlapping segments.

Later, van Helden *et al.* [54] extended their method to find spaced dyads, motifs consisting of two small conserved boxes separated by a fixed spacer. The spacer can be different for distinct motifs; therefore, the spacer is systematically varied between 0 and 16. The significance of this type of motif can be computed based on the combined score of the two conserved parts in the input data or based on the estimated complete dyad frequency from a background data set.

The greatest shortcoming of this method is that there are no variations allowed within an oligonucleotide. Tompa [50] addressed this problem when he proposed an exact method to find short motifs in DNA sequences. Tompa used a different measure from that used by van Helden *et al.* to calculate the statistical significance of motif occurrences. First, for each k-mer s with an allowed number substitutions, the number of sequences in which s is found is calculated. Next, the probability p_s of finding at least one occurrence of s in a sequence drawn from a random distribution is estimated. Finally, the associated *z*-score is computed as

$$z_s = \frac{N_s - Np_s}{\sqrt{Np_s(1-p_s)}}.$$

z_s gives a measure of how unlikely it is to have N_s occurrences of s given the background distribution. Tompa proposed an efficient algorithm to estimate p_s from a set of background sequences based on a Markov chain.

Another interesting string-based approach is based on the representation of a set of sequences with a suffix tree [36], [55]. Sagot *et al.* [55] have used suffix trees to search for single motifs in whole bacterial genomes. Marsan *et al.* [36] later extended the method to search for combinations of motifs. The proposed configuration of a structured motif is a set of p motifs separated by a spacer that might be variable. The variability is limited to ± 2 bp around an average gap length. They also allow for variability within the binding site. The representation of upstream sequences as suffix trees resulted in an efficient implementation despite the large number of possible combinations.

C. Probabilistic Methods

In the previous section, a binding site was modeled as a set of strings. The following methods are all based on a representation of the motif model by a position weight matrix.

1) Probabilistic Model of Sequence Motifs: In the simplest model, we have a set of DNA sequences where each sequence contains a single copy of the motif of fixed length. (For the sake of simplicity, we will consider here only models of DNA sequences, but the whole presentation applies directly to sequences of amino acids.) Except for the motif, a sequence is described as a sequence of independent nucleotides generated according to a single discrete distribution $\theta_0 = (q_0^A, q_0^C, q_0^G, q_0^T)^T$ called the background model. The motif θ_W itself is described by what we call a position weight matrix, which are W independent positions generated according to different discrete distributions q_i^b:

$$\theta_W = \begin{pmatrix} q_1^A & q_2^A & \cdots & q_W^A \\ q_1^C & q_2^C & \cdots & q_W^C \\ q_1^G & q_2^G & \cdots & q_W^G \\ q_1^T & q_2^T & \cdots & q_W^T \end{pmatrix}.$$

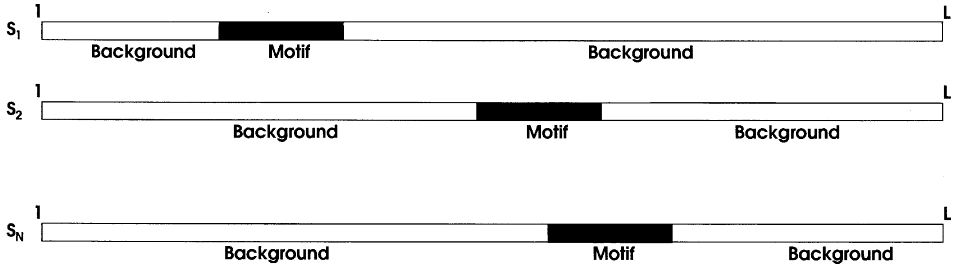


Fig. 8. In this basic sequence model, each sequence contains one and only one copy of the motif. The first part of the sequence is generated according to the background model θ_0, then the motif is generated by the motif matrix θ_W, after which the rest of the sequence is again generated according to the background model.

If we know the location a_i of the motif in a sequence S_i, the probability of this sequence given the motif position, the motif matrix, and the background model is

$$P(S_i|a_i, \theta_W, \theta_0) = \prod_{j=1}^{a_i-1} q_0^{S_{ij}} \prod_{j=a_i}^{a_i+W-1} q_{j-a_i+1}^{S_{ij}} \prod_{j=a_i+W}^{L} q_0^{S_{ij}}.$$

Wherever appropriate, we will pool the motif matrix and the background model into a single set of parameters $\theta = (\theta_0, \theta_W)$. 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 is

$$P(S|A, \theta) = \prod_{i=1}^{N} P(S_i|a_i, \theta).$$

The sequence model is illustrated in Fig. 8. The idea of the EM 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. Gibbs sampling for motif finding extends EM stochastically by not looking for the ML configuration but generating candidate motif matrices and alignments according to their posterior probability given the sequences.

2) Expectation Maximization: One of the first implementations to find a matrix representation of a binding site was a greedy algorithm by Hertz *et al.* [19] to find the site with the highest information content (which is the entropy of the discrete probability distribution represented by the motif matrix). This algorithm was capable of identifying a common motif that is present once in every sequence. This algorithm has been substantially improved over the years [20]. In their latest implementation, CONSENSUS, Hertz and Stormo have provided a framework to estimate the statistical significance of a given information content score based on large deviation statistics.

Within the ML estimation framework, EM is the first choice of optimization algorithm. EM is a two-step iterative procedure for obtaining the ML parameter estimates for a model of observed data and missing values. 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.

EM for motif finding was introduced by Lawrence and Reilly [28] and was an extension of the greedy algorithm of Hertz *et al.* [19]. It was primarily intended for searching motifs in related proteins, but the described method could also be applied to DNA sequences. The basic models assumption is that each sequence contains exactly one copy of the motif, which might be reasonable in proteins but is too strict in DNA. The starting position of each motif instance is unknown and is considered as being a missing value from the data. If the motif positions are known, then the observed frequencies of the nucleotides at each position in the motif are the ML estimates of model parameters. To find the starting positions, each subsequence is scored with the current estimate of the motif model. These updated probabilities are used to re-estimate the motif model. This procedures is repeated until convergence.

Since assuming there is exactly one copy of the motif per sequence is not really biologically sound, Bailey and Elkan proposed an advanced EM implementation for motif finding called MEME [3]. Although MEME was also primarily intended to search for protein motifs, MEME can also be applied to DNA sequences.

To overcome the problem of initialization and getting stuck in local minimums, MEME proposes to initialize the algorithm with a motif model based on a contiguous subsequence that gives the highest likelihood score. Therefore, each substring in the sequence set is used as a starting point for a one-step iteration of EM. Then the motif model with the highest likelihood is retained and used for further optimization steps until convergence. The corresponding motif positions are then masked and the procedure is repeated. Finally, Cardon and Stormo proposed an EM algorithm to search for gapped motifs [8]. However, while performing well for extended protein motifs, EM often suffers badly from local minimums for short DNA motifs.

VI. Gibbs Sampling for Motif Finding

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 ML such as EM, we choose a set of parameters to describe our data by

$$\omega^* = \underset{\omega}{\operatorname{argmax}} P(D|\omega).$$

However, the likelihood function $P(D|\omega)$ contains much more information about the data than just the point estimate $P(D|\omega^*)$. In fact, the posterior distribution $p(\omega|D) = P(D|\omega)P(\omega)/P(D)$ provides a more accurate representation of which parameter values are good candidates to describe our data. For example, if $p(\omega|D)$ 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 our sequence model, 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 [39] (which is well-known as the foundation of the simulated annealing algorithm for global optimization), the hybrid MCMC method [11], and Gibbs sampling [45].

Here we will describe the general Gibbs sampling method and how it can be applied to motif finding. If we assume that we can generate samples $w^{(i)}$ according to the posterior distribution $p(\omega|D)$, we can use these samples to approximate quantities of interest (possibly using Monte Carlo integration). For example, we can approximate a *global* solution with maximum posterior probability by tracking the sample with the highest posterior probability if we draw enough samples from the posterior distribution. Further, we can approximate the posterior mean solution

$$\omega^{\mathrm{PME}} = \int_\omega \omega P(\omega|D)\, d\omega$$

by averaging the samples drawn from the posterior distribution.

A. The Missing Data Problem and Data Augmentation Methods

Before describing of the general Gibbs sampling method, we need to explain 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. However, although the set of sequences S is available, the alignment is unknown. If the alignment were 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 [3]. 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.

B. The Gibbs Sampler

Gibbs sampling is an MCMC method introduced by Tanner and Wong [45] for data augmentation problems. The idea is to describe a complex probability distribution in terms of a Markov chain built with the simpler marginals of the distribution. Suppose we have only three (possibly continuous) variables described by the probability distribution $P(x_1, x_2, x_3)$. Gibbs sampling consists of sampling $x_1^{(i+1)}$ according to $P(x_1|x_2^{(i)}, x_3^{(i)})$, then sampling $x_2^{(i+1)}$ according to $P(x_2|x_1^{(i+1)}, x_3^{(i)})$, and then sampling $x_3^{(i+1)}$ according to $P(x_3|x_1^{(i+1)}, x_2^{(i+1)})$; and repeating this process indefinitely. We denote the fact that, under mild conditions, this Markov chain converges to the joint distribution by the *chain operator*:

$$\begin{aligned} &P(x_1, x_2, x_3)\\ &\quad = P(x_1|x_2, x_3) \bigotimes P(x_2|x_1, x_3) \bigotimes P(x_3|x_1, x_2). \end{aligned}$$

C. The Collapsed Gibbs Sampler

For motif finding, we thus want to build a Gibbs sampler to sample from $P(\theta, A|S)$. However, many variables are now involved, which leaves a great deal of leeway in how the exact sampling is set up. In the basic Gibbs sampler with three variables as an illustration, we have

$$\begin{aligned} &P(x_1, x_2, x_3)\\ &\quad = P(x_1|x_2, x_3) \bigotimes P(x_2|x_1, x_3) \bigotimes P(x_3|x_1, x_2). \end{aligned}$$

But in some cases, we may be able to *group* variables together. For example, we may have

$$P(x_1, x_2, x_3) = P(x_1|x_2, x_3) \bigotimes P(x_3, x_2|x_1)$$

with $P(x_3, x_2|x_1) = P(x_3|x_2, x_1)P(x_2|x_1)$. Or we may be able to *collapse* one of the variables

$$P(x_1, x_2, x_3) = P(x_3|x_1, x_2)\Big(P(x_1|x_2) \bigotimes P(x_2|x_1)\Big).$$

Liu [31] showed that the collapsed Gibbs sampler converges faster than the grouped Gibbs sampler, which itself converges faster than the basic Gibbs sampler.

For motif finding, Liu then proposed to set up a collapsed Gibbs sampler as

$$\begin{aligned} &P(\theta, A|S)\\ &\quad = P(\theta|A, S)\left(\bigotimes_{i=1}^{N} P(a_i|a_1, \ldots, a_{i-1}, a_{i+1}, a_N, S)\right) \end{aligned}$$

where the chain directly approximates $P(A|S)$.

Table 5
Basic Gibbs Sampling Algorithm for Motif Finding

```
INPUT: A set of sequences $S$ and the length $W$ of the motif to search.

1. Compute the background model $\theta_0$ from the nucleotide frequencies observed in $S$.
2. Initialize the alignment vector $A = \{a_i | i = 1, \ldots, N\}$ uniformly at random .
3. For each sequence $S_z, z = 1, \ldots, N$,
   (a) Create subsets $\tilde{S} = \{S_i | i \neq z\}$ and $\tilde{A} = \{a_i | i \neq z\}$.
   (b) Compute $\theta_W$ from the segments indicated by $\tilde{A}$.
   (c) Assign to each possible alignment start $(x_{ij}; i \neq z, j = 1, \ldots, L_i - W + 1)$ in $S_z$ a weight
       $W(x_{ij})$ given by the probability that the corresponding segment is generated by the motif versus
       the background:
           $W(x_{ij}) = \frac{P(S_{ij}, \ldots, S_{i(j+W-1)} | \theta_W)}{P(S_{ij}, \ldots, S_{i(j+W-1)} | \theta_0)}$
                      $= \prod_{k=1}^{W} \frac{q_k^{S_i(j+k+1)}}{q_k^{S_i(j+k-1)}}.$
   (d) For $i \neq z$, draw new alignment positions $a_i$ according to the normalized probability distribution
       $W(x_{ij}) / \sum_{k=1}^{L_i - W + 1} W(x_{ij})$.
4. Repeat Step 3 until the Markov chain reaches convergence (fixed number of iterations).

OUTPUT: A motif matrix $\theta_W$ and an alignment $A$.
```

D. The Basic Algorithm for Gibbs Sampling for Motif Finding

In the previous sections, we have discussed the main ideas behind Gibbs sampling for motif finding. The derivation of the exact algorithm as presented by Lawrence *et al.* [27] and by Liu *et al.* [33] is more technical; we do not discuss it here (Liu covers the technical details). Briefly, 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 in Section V-C1 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 derivation of the collapsed Gibbs sampler involves several properties of integrals of Dirichlet distributions, and a number of approximations are used to speed up the algorithm further. The resulting algorithm appears in Table 5.

E. Extended Gibbs Sampling Methods

Several groups proposed advanced methods to fine-tune the Gibbs sampling algorithm for motif finding in DNA sequences. A first version of the Gibbs sampling algorithm that was especially tuned toward finding motif in DNA sequences is AlignACE [41]; this version was later refined [22]. This algorithm was the first reported to be used for the analysis of gene clusters. Several modifications were made in AlignACE with respect to the original Gibbs sampling algorithm. First, one motif at the time was retrieved and the positions were masked instead of simultaneous multiple motif searching. Second, AlignACE was implemented with a fixed single nucleotide background model based on base frequency in the sequence set. Also, both strands were included in the search. Finally, in the latest version, the *maximum a posteriori* likelihood score was used to judge different motifs.

BioProspector [34] also uses a Gibbs sampling strategy to search for common motifs in the regulatory region of coexpressed genes. In this implementation various extensions are proposed. First, BioProspector uses zero to third-order Markov background models. The predictive update formula is changed so that the probability of the instance being generated by the background model is given by this higher-order background model

$$P_b(x) = P(b_l)P(b_{l+1}|b_l)P(b_{l+2}|b_{l+1}b_l) \cdots P(b_{l+W-1}|b_{l+W-2}b_{l+W-3}b_{l+W-4}).$$

Second, the core sampling step was replaced by a *threshold sampler*. This threshold sampling step was incorporated to estimate the number of copies of a motif in a sequence. The program defines two thresholds, T_L and T_H. Instances with a score W_x higher than T_H will be automatically selected, while there will be one motif sampled from those motifs that have a score between T_L and T_H. T_H is set proportional to the product of the average length of the input sequences and the motif width. T_L is initialized at zero and linearly increased till it reaches the value of $T_H/8$. This threshold sampling step ensures faster convergence. As another modification, BioProspector proposes two possible alternative motif models. The first possibility is to search for palindromic motifs. The second possibility is to search for a gapped version of the motif model, where the motif consists of two blocks separated by a gap of variable length. The gapped version searches for two motifs at the same time that occur within a given range.

Within the INCLUSive framework [49], we designed the Motif Sampler [47], [48] specifically to search in sets of upstream sequences from groups of coexpressed genes. Such groups typically come from a cluster analysis of the expression profiles. Since the results of clustering are known to be subject to noise, only a subset of the set of coexpressed genes will be actually coregulated and have one or more copies of the binding site. Therefore, it is important to have an algorithm that can cope with this noise. The Motif Sampler uses the framework of the probabilistic sequence model to estimate the number of copies of a motif in a sequence. For each sequence in the data set, the number of copies of the motif is estimated, which is more accurate than earlier methods. Furthermore, we demonstrated [48] that the use of a higher-order background model built from an independent data set significantly improves the performance of the Gibbs sampling algorithm. We also provide precompiled background models for several organisms (*Arabidopsis thaliana, S. cerevisiae, E. coli, Helicobacter pylori, Caenorhabditis elegans*).

To exemplify the improvements obtained by further refinements of the Gibbs sampling strategy, we report here briefly the use of higher-order background models on a data set of coregulated genes from plants (see Fig. 9). 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. Additionally, noisy sequences not suspected to contain an active motif are added gradually. 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. The improved robustness of the method thanks to the higher-order background model is discussed extensively in [47].

Ann_spec [57] has a slightly different approach to model the motif. The motif model is represented with a sparsely encoded perceptron with one processing unit. The weights of the perceptron resemble the position weight matrix. This model is based on the approximation of the total protein binding energy by the sum of partial binding energies at the individual nucleotides in the binding sites. The use of a perceptron is also justified by the fact that it can be used to approximate posterior probability distributions. A gradient descent training method is used to find the parameters of the perceptron. For the training set for the perceptron, positive examples are selected using a Gibbs sampling procedure. Negative examples can be either constructed from random sequence or from genomic data. To improve the specificity of the motif model, a background model based on an independent data set is preferred.

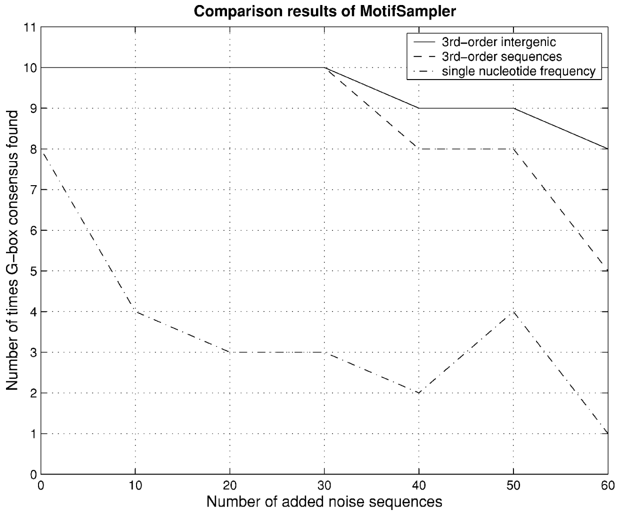


Fig. 9. Total number of times the G-box motif is found in 10 repeated runs of the tests for three different background models. The data set consists of the 33 G-box sequences and a fixed number of added noisy sequences. The background models are order 0 and order 3 based on a reference set or on the data only. A background model of order 0 corresponds to the classical background model of earlier versions of Gibbs sampling for motif finding.

Ann_spec was recently extended to search for cooperatively acting transcription binding factors by GuhaThakurta and Stormo [16]. Co-Bind searches for two motifs simultaneously by combining the weights that optimize the objective functions of the two individual perceptrons. The identification of two motifs simultaneously improved significantly the detection of the true motifs compared with the classical methods searching for one motif at the time.

McCue *et al.* have used a Gibbs motif-finding algorithm for phylogenetic footprinting [37]. They also proposed a motif model that accounts for palindromic motifs. Their most important contribution lies in the use of a position-specific background model estimated with a Bayesian segmentation model [32]. This model accounts for the varying composition of the DNA upstream of a gene.

VII. INCLUSive: INtegrated CLustering, Upstream Sequence Retrieval, and Motif Sampling

Analysis of a 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

Table 6
Results of the Motif Search in Four Clusters From a Microarray Experiment on Mechanical Wounding in *A. thaliana* for the Third-Order Background Model

Cluster	Consensus	Runs	PlantCARE	Descriptor
1	TAArTAAGTCAC	7/10	TGAGTCA	tissue specific GCN4-motif
(11 seq.)			CGTCA	MeJA-responsive element
	ATTCAAATTT	8/10	ATACAAAT	element associated to GCN4-motif
	CTTCTTCGATCT	5/10	TTCGACC	elicitor responsive element
2	TTGACyCGy	5/10	TGACG	MeJa responsive element
(6 seq.)			(T)TGAC(C)	Box-W1, 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	yTGACCGTCCsA	9/10	CCGTCC	meristem specific activation of H4 gene
(5 seq.)			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	-	-

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.

To make such an integrated analysis of microarray data possible, we have developed and made publicly available our INCLUSive Web tool (INtegrated CLustering, Upstream sequence retrieval, and motif Sampling; http://www.esat.kuleuven.ac.be/~dna/BioI) (see also the flowchart of Fig. 1). As an illustration of the results obtained by combined adaptive quality-based clustering and the Motif Sampler, we show the results of motif finding on a microarray experiment in plants. The data are from a microarray experiment on the response to mechanical wounding of the plant *A. thaliana*. The microarray consists of 150 genes related to stress response in plants. The experiment consists of expression measurements for those 150 genes at seven time points following wounding (after 30 min, 60 min, 90 min, 3 h, 6 h, 9 h, and 24 h). The expression data were clustered using adaptive quality-based clustering with a significance level of 95%. Four clusters where identified that contained at least five genes and those were selected for motif finding. The Motif Sampler was used to search for six motifs of length 8 bp and for six motifs of length 12 bp. A background model of order 3 was selected as it gave the most promising results. The analysis was repeated ten times, and only the motifs identified in at least five runs were retained. Table 6 presents the motifs found. 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 (e.g., $r = A/G$); capitals are for strong positions, and lower letters are for degenerate positions. The third column gives the number of times this motif was found in the ten runs. The fourth column gives matching known motifs found in the PlantCARE database [29], if any. Finally, the last column gives a short explanation of the matching known motifs.

VIII. Conclusion

We have presented algorithmic methods for the analysis of microarray data for motif finding. Using microarrays is a powerful technique to monitor the expression of thousands of genes, and a key technique for biologists attempting to unravel the regulation mechanisms of genes in an organism. After reviewing the basics of microarray technology, we introduced some concepts from molecular biology to describe how transcription factors recognize binding sites to control gene activation. We then introduced the strategy of integrating clustering (to detect groups of potentially coregulated genes) with motif finding (to detect the DNA motifs that control this coregulation). We presented several clustering techniques (such as hierarchical clustering, K-means, self-organizing maps, quality-based clustering, and our adaptive quality-based clustering) and discussed their respective advantages and shortcomings. We also discussed the preprocessing steps necessary to prepare microarray data for clustering: normalization, nonlinear transformation, missing value replacement, filtering, and rescaling. We also presented several strategies to validate the results of clustering biologically as well as statistically. Turning to motif finding, we described the two main classes of methods for motif finding: word counting and probabilistic sequence models. We focused on the particular technique of Gibbs sampling for motif finding. After reviewing the basic ideas underlying this MCMC method, we discussed several extensions that improve the effectiveness of this method in practice. We introduced our Motif Sampler, which in particular includes the use of higher-order background models that increase the robustness of Gibbs sampling for motif finding. Finally, we briefly presented our integrated Web tool INCLUSive, which allows the easy analysis of microarray data for motif finding. Furthermore, we illustrated the different steps of this integrated data analysis at the hand of several practical examples.

It should be emphasized that a major endeavor of bioinformatics is to develop methodologies that integrate multiple types of data (here expression data together with sequence data) to obtain robust and biologically relevant results in an efficient and user-friendly manner. Only such powerful tools can deliver the necessary support for 21st-century molecular biology.

Acknowledgment

The authors would like to thank for their cooperation Prof. P. Rouzé and S. Rombauts of the Department of Plant Systems Biology of the University of Ghent, Belgium and of the Flemish Interuniversity Institute for Biotechnology, and Magali Lescot of the Department of Genetics and Physiology of Development of the University of Marseille, France.

References

[1] U. Alon, N. Barkai, D. A. Notterman, K. Gish, S. Ybarra, D. Mack, and A. J. Levine, "Broad patterns of gene expression revealed by clustering analysis of tumor and normal colon tissues probed by oligonucleotide arrays," in *Proc. Natl. Acad. Sci. USA*, vol. 96, 1999, pp. 6745–6750.

[2] F. Azuaje, "A cluster validity framework for genome expression data," *Bioinformatics*, vol. 18, pp. 319–320, 2002.

[3] T. L. Bailey and C. Elkan, "Unsupervised learning of multiple motifs in biopolymers using expectation maximization," *Machine Learning*, vol. 21, pp. 51–80, 1995.

[4] A. Ben-Dor, N. Friedman, and Z. Yakhini, "Class discovery in gene expression data," in *Proc. 5th Annu. Conf. Comput. Mol. Biol.*, 2001, pp. 31–38.

[5] A. Ben-Dor, R. Shamir, and Z. Yakhini, "Clustering gene expression patterns," *J. Comput. Biol.*, vol. 6, pp. 281–297, 1999.

[6] M. Bittner, P. Meltzer, Y. Chen, Y. Jiang, E. Seftor, M. Hendrix, M. Radmacher, R. Simon, Z. Yakhini, A. Ben-Dor, N. Sampas, E. Dougherty, E. Wang, F. Marincola, C. Gooden, J. Lueders, A. Glatfelter, P. Pollock, J. Carpten, E. Gillanders, D. Leja, K. Dietrich, C. Beaudry, M. Berens, D. Alberts, and V. Sondak, "Molecular classification of cutaneous malignant melanoma by gene expression profiling," *Nature*, vol. 406, pp. 536–540, 2000.

[7] P. Bucher, "Regulatory elements and expression profiles," *Current Opinion in Structural Biol.*, vol. 9, pp. 400–407, 1999.

[8] L. R. Cardon and G. D. Stormo, "Expectation maximization for identifying protein-binding sites with variable lengths from unaligned DNA fragments," *J. Mol. Biol.*, vol. 223, pp. 159–170, 1992.

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

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

[11] S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth, "Hybrid Monte Carlo," *Phys. Lett. B*, vol. 195, pp. 216–222, 1990.

[12] D. J. Duggan, M. Bittner, Y. Chen, P. Meltzer, and J. M. Trent, "Expression profiling using cDNA microarrays," *Nature Genetics*, vol. 21, no. 1 Suppl., pp. 10–14, 1999.

[13] M. B. Eisen, P. T. Spellman, P. O. Brown, and D. Botstein, "Cluster analysis and display of genome-wide expression patterns," in *Proc. Nat. Acad. Sci. USA*, vol. 95, 1998, pp. 14 863–14 868.

[14] C. Fraley and E. Raftery, "MCLUST: Software for model-based cluster analysis," *J. Classification*, vol. 16, pp. 297–306, 1999.

[15] D. Ghosh and A. M. Chinnaiyan, "Mixture modeling of gene expression data from microarray experiments," *Bioinformatics*, vol. 18, pp. 275–286, 2002.

[16] D. GuhaThakurta and G. D. Stormo, "Identifying target sites for cooperatively binding factors," *Bioinformatics*, vol. 17, pp. 608–621, 2001.

[17] T. Hastie, R. Tibshirani, M. B. Eisen, A. Alizadeh, R. Levy, L. Staudt, W. C. Chan, D. Botstein, and P. Brown. (2000) 'Gene shaving' as a method for identifying distinct sets of genes with similar expression patterns. *Genome Biology* [Online]. Available: http://genome-biology.com/2000/1/2/research/0003/.

[18] J. Herrero, A. Valencia, and J. Dopazo, "A hierarchical unsupervised growing neural network for clustering gene expression patterns," *Bioinformatics*, vol. 17, pp. 126–136, 2001.

[19] G. Z. Hertz, G. W. Hartzell, and G. D. Stormo, "Identification of consensus patterns in unaligned DNA sequences known to be functionally related," *Comput. Appl. Biosci.*, vol. 6, pp. 81–92, 1990.

[20] G. Z. Hertz and G. D. Stormo, "Identifying DNA and protein patterns with statistically significant alignments of multiple sequences," *Bioinformatics*, vol. 15, no. 7/8, pp. 563–577, 1999.

[21] L. J. Heyer, S. Kruglyak, and S. Yooseph, "Exploring expression data: Identification and analysis of coexpressed genes," *Genome Res.*, vol. 9, pp. 1106–1115, 1999.

[22] J. D. Hughes, P. W. Estep, S. Tavazoie, and G. M. Church, "Computational identification of *cis*-regulatory elements associated with groups of functionally related genes in *Saccharomyces cerevisiae*," *J. Mol. Biol.*, vol. 296, pp. 1205–1214, 2000.

[23] J. Quackenbush, "Computational analysis of microarray data," *Nat. Rev. Genetics*, vol. 2, pp. 418–427, 2001.

[24] L. Kaufman and P. J. Rousseeuw, *Finding Groups in Data: An Introduction to Cluster Analysis*. New York: Wiley, 1990.

[25] M. K. Kerr and G. A. Churchill, "Bootstrap cluster analysis: Assessing the reliability of conclusions from microarray experiments," *Proc. Nat. Acad. Sci. USA*, vol. 98, pp. 8961–8965, 2001.

[26] T. Kohonen, *Self-Organizing Maps*. Berlin, Germany: Springer-Verlag, 1997.

[27] C. E. Lawrence, S. F. Altschul, M. S. Boguski, J. S. Liu, A. F. Neuwald, and J. C. Wootton, "Detecting subtle sequence signals: A Gibbs sampling strategy for multiple alignment," *Science*, vol. 262, pp. 208–214, 1993.

[28] C. E. Lawrence and A. A. Reilly, "An expectation maximization (EM) algorithm for the identification and characterization of common sites in unaligned biopolymer sequences," *Proteins*, vol. 7, pp. 41–51, 1990.

[29] M. Lescot, P. Déhais, G. Thijs, K. Marchal, Y. Moreau, Y. Van de Peer, P. Rouzé, and S. Rombauts, "PlantCARE, a database of plant *cis*-acting regulatory elements and a portal to tools for *in silico* analysis of promoter sequences," *Nucleic Acids Res.*, vol. 30, pp. 325–327, 2002.

[30] R. J. Lipschutz, S. P. A. Fodor, T. R. Gingeras, and D. J. Lockheart, "High density synthetic oligonucleotide arrays," *Nature Genetics*, vol. 21, no. 1, pp. Suppl. 20–24, 1999.

[31] J. S. Liu, "The collapsed Gibbs sampler in Bayesian computations with applications to a gene regulation problem," *J. Amer. Statist. Assoc.*, vol. 89, no. 427, pp. 958–966, 1994.

[32] J. S. Liu and C. E. Lawrence, "Bayesian inference on biopolymer models," *Bioinformatics*, vol. 15, pp. 38–52, 1999.

[33] J. S. Liu, A. F. Neuwald, and C. E. Lawrence, "Bayesian models for multiple local sequence alignment and Gibbs sampling strategies," *J. Amer. Statist. Assoc.*, vol. 90, no. 432, pp. 1156–1170, 1995.

[34] X. Liu, D. L. Brutlag, and J. S. Liu, "BioProspector: Discovering conserved DNA motifs in upstream regulatory regions of co-expressed genes," in *Proc. Pacific Symp. Biocomputing*, vol. 6, 2001, pp. 127–138.

[35] A. V. Lukashin and R. Fuchs, "Analysis of temporal gene expression profiles: Clustering by simulated annealing and determining the optimal number of clusters," *Bioinformatics*, vol. 17, pp. 405–414, 2001.

[36] L. Marsan and M.-F. Sagot, "Algorithms for extracting structured motifs using a suffix tree with application to promoter and regulatory site consensus identification," *J. Comp. Biol.*, vol. 7, pp. 345–360, 2000.

[37] L. A. McCue, W. Thompson, C. S. Carmack, M. P. Ryan, J. S. Liu, V. Derbyshire, and C. E. Lawrence, "Phylogenetic footprinting of transcription factor binding sites in proteobacterial genomes," *Nucleic Acids Res.*, vol. 29, pp. 774–782, 2001.

[38] H. W. Mewes, D. Frishman, C. Gruber, B. Geier, D. Haase, A. Kaps, K. Lemcke, G. Mannhaupt, F. Pfeiffer, C. Schuller, S. Stocker, and B. Weil, "MIPS: A database for genomes and protein sequences," *Nucleic Acids Res.*, vol. 28, pp. 37–40, 2000.

[39] R. M. Neal, *Bayesian Learning for Neural Networks*. New York: Springer, 1996, Lecture Notes in Statistics.

[40] U. Ohler and H. Niemann, "Identification and analysis of eukaryotic promoters: Recent computational approaches," *Trends in Genetics*, vol. 17, no. 2, pp. 56–60, 2001.

[41] F. P. Roth, J. D. Hughes, P. W. Estep, and G. M. Church, "Finding DNA regulatory motifs within unaligned noncoding sequences clustered by whole genome mRNA quantitation," *Nature Biotech.*, vol. 16, pp. 939–945, 1998.

[42] G. Schwarz, "Estimating the dimension of a model," *Ann. Statist.*, vol. 6, pp. 461–464, 1978.

[43] G. Sherlock, "Analysis of large-scale gene expression data," *Current Opinion in Immunology*, vol. 12, pp. 201–205, 2000.

[44] P. Tamayo, D. Slonim, J. Mesirov, Q. Zhu, S. Kitareewan, E. Dmitrovsky, E. S. Lander, and T. R. Golub, "Interpreting patterns of gene expression with self-organizing maps: methods and application to hematopoietic differentiation," *Proc. Nat. Acad. Sci. USA*, vol. 96, pp. 2907–2912, 1999.

[45] M. A. Tanner and W. H. Wong, "The calculation of posterior distributions by data augmentation (with discussion)," *J. Amer. Statist. Assoc.*, vol. 82, no. 398, pp. 528–550, 1987.

[46] S. Tavazoie, J. D. Hughes, M. J. Campbell, R. J. Cho, and G. M. Church, "Systematic determination of genetic network architecture," *Nature Genetics*, vol. 22, no. 7, pp. 281–285, 1999.

[47] G. Thijs, M. Lescot, K. Marchal, S. Rombauts, B. De Moor, P. Rouzé, and Y. Moreau, "A higher order background model improves the detection of promoter regulatory elements by Gibbs sampling," *Bioinformatics*, vol. 17, no. 12, pp. 1113–1122, 2001.

[48] G. Thijs, K. Marchal, M. Lescot, S. Rombauts, B. De Moor, P. Rouzé, and Y. Moreau, "A Gibbs sampling meyhod to detect over-represented motifs in the upstream regions of co-expressed genes," *J. Comput. Biol.*, vol. 9, no. 2, pp. 447–464, 2002.

[49] G. Thijs, Y. Moreau, F. De Smet, J. Mathys, M. Lescot, S. Rombauts, P. Rouzé, B. De Moor, and K. Marchal, "INCLUSive: INtegrated CLustering, Upstream sequence retrieval and motif sampling," *Bioinformatics*, vol. 18, no. 2, pp. 331–332, 2002.

[50] M. Tompa, "An exact method for finding short motifs in sequences, with application to the ribosome binding site problem," in *Proc. 7th Intl. Conf. Intelligent Syst. for Mol. Biol.*, Heidelberg, Germany, Aug. 1999, pp. 262–271.

[51] J. T. Tou and R. C. Gonzalez, "Pattern classification by distance functions," in *Pattern Recognition Principles*. Reading, MA: Addison-Wesley, 1979, pp. 75–109.

[52] O. Troyanskaya, M. Cantor, G. Sherlock, P. Brown, T. Hastie, R. Tibshirani, D. Botstein, and R. B. Altman, "Missing value estimation methods for DNA microarrays," *Bioinformatics*, vol. 17, pp. 520–525, 2001.

[53] J. van Helden, B. André, and L. Collado-Vides, "Extracting regulatory sites from upstream region of yeast genes by computational analysis of oligonucleotide frequencies," *J. Mol. Biol.*, vol. 281, pp. 827–842, 1998.

[54] J. van Helden, A. F. Rios, and J. Collado-Vides, "Discovering regulatory elements in noncoding sequences by analysis of spaced dyads," *Nucleic Acids Res.*, vol. 28, no. 8, pp. 1808–1818, 2000.

[55] A. Vanet, L. Marsan, A. Labigne, and M. F. Sagot, "Inferring regulatory elements from a whole genome. An analysis of helicobacter pylori sigma^{80} family of promoter signals.," *J. Mol. Biol.*, vol. 297, no. 2, pp. 335–353, 2000.

[56] T. Werner, "Models for prediction and recognition of eukaryotic promoters," *Mammalian Genome*, vol. 10, pp. 71–80, 1999.

[57] C. T. Workman and G. D. Stormo, "Ann-spec: A method for discovering transcription binding sites with improved specificity," in *Proc. Pacific Symp. Biocomputing*, vol. 5, Honolulu, HI, 2000, pp. 464–475.

[58] K. Y. Yeung, C. Fraley, A. Murua, A. E. Raftery, and W. L. Ruzzo, "Model-based clustering and data transformations for gene expression data," *Bioinformatics*, vol. 17, pp. 977–987, 2001.

[59] K. Y. Yeung, D. R. Haynor, and W. L. Ruzzo, "Validating clustering for gene expression data," *Bioinformatics*, vol. 17, pp. 309–318, 2001.

[60] K. Y. Yeung and W. L. Ruzzo, "Principal component analysis for clustering gene expression data," *Bioinformatics*, vol. 17, pp. 763–774, 2001.

[61] J. Zhu and M. Q. Zhang, "Cluster, function and promoter: Analysis of yeast expression array," in *Proc. Pacific Symp. Biocomputing*, vol. 5, 2000, pp. 467–486.

Yves Moreau was born in Haine-St-Paul, Belgium, in July 1970. He received the M.S. of electrical engineering from the Faculté Polytechnique de Mons, Mons, Belgium in 1992. He was a Fulbright Grantee at the Division of Applied Mathematics, Brown University, Providence, RI, and there received the M.S. degree in applied mathematics in 1993. He received the Ph.D. degree in electrical engineering from the Katholieke Universiteit Leuven, Leuven, Belgium, in 1998.

He is currently an Assistant Professor at the Department of Electrical Engineering of K.U. Leuven and a Postdoctoral Researcher with FWO—Vlaanderen (the Fund for Scientific Research, Flanders, Belgium). He is also a cofounder of the spin-off company Data4s NV. His research focuses on Bayesian and maximum-likelihood methods for graphical models in bioinformatics and medical informatics, with an emphasis on microarray data analysis.

Dr. Moreau was awarded the biannual Siemens Prize in 2002.

Frank De Smet was born in Bonheiden, Belgium, in August 1969. He received the M.S. degree in electrical and mechanical engineering and is a Medical Doctor with an additional degree in electrocardiography from the Katholieke Universiteit, Leuven, Belgium, in 1992 and 1998, respectively. He is currently pursuing the Ph.D. degree at the Department of Electrical Engineering at K.U. Leuven. He is a Reviewer for the Bioinformatics journal and the International conference on Intelligent Systems for Molecular Biology (ISMB). His research interests are in the area of bioinformatics and biostatistics, more specifically in the clinical management of malignant neoplasms using microarrays, in clustering of gene expression profiles and in the development of artificial intelligence models for the assessment of endometrial, ovarian, and breast cancer.

Gert Thijs (Student Member, IEEE) was born in Bilzen, Belgium, on June 1, 1973. He received the M.S. degree in electrical engineering at the Katholieke Universiteit Leuven, Leuven, Belgium, in 1998. He is currently pursuing the Ph.D. degree at the Department of Electrical Engineering (ESAT) at K.U. Leuven.

He is a Research Assistant with the Institute for the Promotion of Innovation by Science and Technology (IWT), Flanders, Belgium. The subject of his Ph.D. research is the application of advanced pattern recognition and data mining methods to detect regulatory elements in DNA sequences. His main focus is on the application of Gibbs sampling for motif finding. He has also a strong interest in the practical application and integration of diverse bioinformatics tools.

Kathleen Marchal was born in Leuven, Belgium in 1972. She received the M.S. degree in bioengineering and the Ph.D. degree in applied biological sciences from the Faculty of Agricultural and Applied Biological Sciences, Katholieke Universiteit Leuven, Leuven, Belgium, in 1995 and 1999, respectively.

She is currently working in the ESAT/SCD bioinformatics group as a Postdoctoral Researcher with FWO—Vlaanderen (the Fund for Scientific Research, Flanders, Belgium). Her research has been focusing on the analysis and inference of relevant biological information from global expression profiling experiments (preprocessing and cluster analysis of microarray data, retrieval of regulatory motifs and genetic network inference).

Dr. Marchal was laureate of the DSM prize for Chemistry and Technology in 2000. In 2002, together with Dr. J. Mathys and Dr. Y. Moreau, she received the biannual Siemens prize.

Bart De Moor (Senior Member, IEEE) was born in Halle, Brabant, Belgium, on July 12, 1960. He received the Ph.D. degree in applied sciences in 1988 from the Katholieke Universiteit Leuven, Leuven, Belgium.

He was a Visiting Research Associate with the Department of Computer Science and Electrical Engineering, Stanford University, Stanford, CA, from 1988 to 1989. Since 1989, he has been with the Electrical Engineering Department, K.U. Leuven, where he has been a Full Professor since 2000. From 1991 to 1992, he was the Chief of Staff of the Belgian Federal Minister of Science W. Demeester-DeMeyer and, later, of the Belgian Prime Minister W. Martens. From 1994 to 1999, he was the main advisor on science and technology policy to the Flemish Minister-President L. Van den Brande. His research interests include numerical linear algebra, system identification, control theory, and signal processing. He has more than 200 papers in international journals and conference proceedings.

Dr. De Moor received the Leybold-Heraeus Prize in 1986, the Leslie Fox Prize in 1989, the Guillemin-Cauer Best Paper Award of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS in 1990, and the biannual Siemens prize in 1994. He became a Laureate of the Belgian Royal Academy of Sciences in 1992. He is a member of several boards of administrators of (inter)national scientific, cultural, and commercial organizations, including the Belgian Institute for Control and Automation, the Flemish Interuniversity Institute for Biotechnology, and the spin-off companies ISMC NV and Data4s NV.