Data Analysis and Bioinformatics

Vito Di Gesù

C.I.T.C., Università di Palermo, Italy

Abstract. Data analysis methods and techniques are revisited in the case of biological data sets. Particular emphasis is given to clustering and mining issues. Clustering is still a subject of active research in several fields such as statistics, pattern recognition, and machine learning. Data mining adds to clustering the complications of very large data-sets with many attributes of different types. And this is a typical situation in biology. Some cases studies are also described.

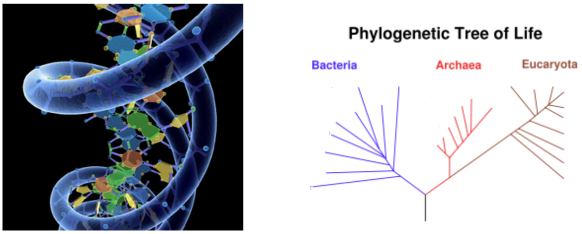
Keywords: Clustering, data mining, bio-informatics, Kernel methods, Hidden Markov Models, Multi-Layers Model.

1 Introduction

Bio-informatics is a new discipline devoted to the solution of biological problems, usually on the molecular level, by the use of techniques including applied mathematics, statistics, computer science, and artificial intelligence. Major research efforts regard sequence alignment [1], gene finding [2], genome assembly, protein structure alignment [3] and prediction [4], prediction of gene expression, protein-protein interactions, and the modeling of evolution [5].

Mining in structured data is particularly relevant for bio-informatics applications, since the majority of biological data is not kept in databases consisting of a single, flat table [6]. In fact, bio-informatics databases, BDB, are structured and linked *objects*, connected by relations representing a rich internal structure. Examples of BDB are databases of proteins [7], of small molecules [8], of metabolic and regulatory networks [9]. Moreover, biological data representations are structured and heterogeneous; they consist of large sequences (e.g. 10^6 gene sequences), $2D$ large structures (e.g. $10^5 \sim 10^6$ spots on DNA chips), $3D$ structures (e.d. DNA phosphate model, Figure 1a), graphs, networks, expression profiles, and phylogenetic trees (Figure 1b). Several issues are dealing with mining biological data, among them there are *kernel methods* for classification of microarray time series data [10]. This classification of gene expression time series has many potential applications in medicine and pharmacogenomics, such as disease diagnosis, drug response prediction or disease outcome prognosis, contributing to individualized medical treatment. Graph kernels representations of proteins have been designed to retrieve structure and bio-chemical information and protein function prediction. Feature graphs are considered to represent potential docking sites and retrieve activity maps $3D$ protein databases.

A. Ghosh, R.K. De, and S.K. Pal (Eds.): PReMI 2007, LNCS 4815, pp. 373–388, 2007.



(a) (b)

Fig. 1. (a) 3D structure of the DNA phosphate model; (b) an example of phylogenetic tree

Concept of similarity play a relevant role in search both $2D$ and $3D$ shape matching in bio-molecular databases. For example, similar $3D$ shape can be retrieved by using a similarity model based on $3D$ shape histograms, $3D$ surface segments, and parametric surface functions including paraboloid and trigonometric polynomials that approximate surface segments.

Finally, methods for finding all subspaces of high-dimensional data containing density-based clusters are necessary because finding clusters in high-dimensional data is usually futile. Moreover, high-dimensional data may be clustered differently in varying subspaces of the feature space. Subspace clustering aims at finding all subspaces of high-dimensional data in which clusters exist.

Specific topics include: preprocessing tasks such as data cleaning and data integration as applied to biological data; classification and clustering techniques for microarrays; comparison of RNA structures based on string properties and energetics; discovery of the sequence characteristics of different parts of the genome; mining of haplotype to find disease markers; sequencing of events leading to the folding of a protein; inference of the subcellular location of protein activity; classification of chemical compounds based on structure; special purpose metrics and index structures for phylogenetic applications; query languages for protein searching based on the shape of proteins, and very fast indexing schemes for sequences and pathways.

The paper is structured as follows: Section 2 outlines both the descriptive and the predictive mining in databases; Section 3 reviews recent clustering algorithms for biological data; in Section 4 two cases studies are described; Section 5 provides final remarks and new perspectives in mining biological data.

2 Mining in Biological Database

Data mining techniques are classified in *descriptive* and *predictive*. In descriptive mining, local structures are searched to discover pattern embedded in data. In predictive mining, models are designed to make predictions for new, unseen cases.

2.1 Descriptive Data Mining

The problem of finding patterns of interest in a data-set is a typical pattern recognition problem, that depends on the nature of problem. For example, the problem of finding frequent item-sets in coregulated genes by estimating number of motif instances has been considered in [11]. Authors ground their analysis on TOUCAN system [12] and Hidden Markov Model inference nets [13]. The prediction of Cis-Regulatory Elements is analyzed in [14] combining different algorithms (Clover, Cluster-Buster, sequence identity, and ITB-algorithm).

Search technique have been developed to solve pattern matching problems in other domains, such as approximate string matching on large DNA sequences [15,16]. These methods include star alignments and tree alignments, which are usually based on dynamic programming. In [17] a polynomial-time dynamic programming algorithm for solving the maximum common subtree of two trees is considered to implement an accurate and efficient tool for finding and aligning maximally matching glycan trees (see Figure 2). For a review on trees matching see [18].

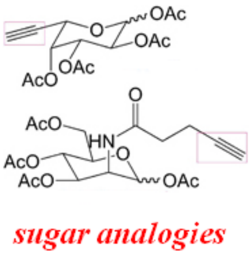
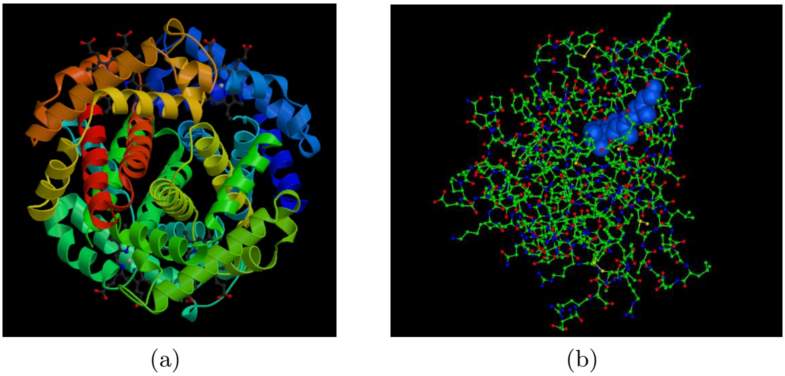


Fig. 2. (a) An example of biological tree-structure

Structure similarity of two proteins from the matching of pairs of secondary structure elements. In [19] the matching is performed using a fast bipartite graph-matching algorithm that avoids the computational complexity of searching for the full subgraph isomorphism between the two sets of interactions. More information on graph matching in biology can be found in [20].

Matching algorithms are interesting to find all sub-patterns occurring with a minimum frequency in a database of patterns (strings, trees, graphs). The problem can be extended in finding all patterns with a minimum frequency in one data-set and a maximum frequency in another. This is a question relevant for the analysis of differentially expressed genes with applications to protein structure folding prediction and drug discovering, both of them are characterized by 3D structures (see Figures 3a,b).



(a) (b)

Fig. 3. (a) 3D structure of a protein molecule; (b) an example of binding drug molecule into a protein molecule

2.2 Predictive Mining

Data Mining is an analytic process designed to explore a large amount of data to find consistent patterns and/or systematic relationships between variables, and then to validate the findings by applying the detected patterns to new subsets of data. *Predictive Data Mining* (*PDM*) is usually applied to identify a statistical models that can be used to predict some response of interest [21]. For example, a *PDM* may be more exploratory in nature to identify cluster or to aggregate or amalgamate the information in very large data sets into useful and manageable chunks. *PDM* techniques can be very useful in the case of inductive databases. The process of data mining consists of three stages:

1) *The initial exploration* usually starts with data preprocessing followed by data transformations to select subsets of records. In case of data sets with large numbers of variables (”fields”), preliminary feature selection operations are performed to lower the number of variables to a manageable range. Depending on the nature of the analytic problem, data mining may involve a simple choice of straightforward predictors for a regression model, and a wide variety of graphical and statistical methods in order to identify the most relevant variables and determine the complexity and/or the general nature of models that can be taken into account in the next stage.

2) *Model building and validation* considers and evaluates various models to chose the best one based on their predictive performance. This may be a very elaborate process. There are a variety of techniques developed to achieve this goal - many of which are based on the so-called ”competitive evaluation of models”, that is, applying different models to the same data set and then comparing their performance to choose the best. These techniques - which are often considered the core of predictive data mining - include: Bagging (Voting, Averaging), Boosting, Stacking (Stacked Generalizations), and Meta-Learning.

3) *Deployment* uses the model selected as best and applies it to new data in order to generate predictions or estimates of the expected outcome (i.e., the application of the model to new data in order to generate predictions).

Traditionally *PDM* has been applied to business or market related. Recently, *PDM*, due to the enormous growing of biological repositories of gene expression data generated by DNA microarray experiments, is providing a new tool for both medical diagnosis and genomic studies. In [22] a systematic approach for learning and extracting rule-based knowledge from gene expression data is presented. A class of predictive self-organizing network, known as *Adaptive Resonance Associative Map* (ARAM), is used for modeling gene expression data, whose learned knowledge can be transformed into a set of symbolic *IF-THEN* rules for interpretation.

3 Survey on Clustering Methods in Bioinformatics

Clustering is the process of grouping data objects into a set of disjoint classes so that objects within a class have high *similarity* to each other, while objects in separate classes are more *dissimilar*. Clustering is part of exploratory data analysis, where *rules* are *eventually* found as a *creative* induction scheme that implies the need for experimental and theoretical models validations.

Currently, typical microarray experiments may contain 10^6 genes. One of the characteristics of gene expression data is that it is meaningful to cluster both genes and samples [23,24]. Here, genes are treated as elements, while samples are features. On the other hand, samples can be partitioned into homogeneous groups that may correspond to some particular macroscopic phenotype. The distinction of gene-based clustering and sample-based clustering is grounded on different characteristics of clustering tasks for gene expression data. Some clustering algorithms, such as K-means and hierarchical approaches, can be used both to group genes and to partition samples. In the following a list of most used clustering techniques to analyze biological data is listed.

K-means [25] is a partition-based clustering method. Given a pre-specified number K, K-means partitions the data set into K disjoint clusters such that the sum of the squared distances of elements from their cluster centers is minimized.

K-means has been applied on gene expression data [26], finding clusters that contain a significant portion of genes with similar functions. Moreover, upstream sequences of DNA-genes within the same cluster allowed to extract 18 motifs, which are promising candidates for novel cis-regulatory elements. The K-means algorithm has some drawbacks (setting of number of clusters, it produces a large number of outliers). To overcome them, several algorithms have been proposed. For example, the K-medoids algorithm uses an element closest to the center of a cluster as the representative (medoid) such that the total distance between the K selected medoids and the other elements is minimized. This algorithm is more robust to the outliers than K-means. Another group of algorithms use some thresholds to control the coherence of clusters. For example, the maximal similarity between two separate cluster centroids and the minimal similarity between an element and its cluster centroid, [27]. In [28], clusters are constrained to have a diameter no larger than d (compact clusters); in [29] a more efficient algorithm (*Adapt_Cluster*) is proposed. Here, an element will be assigned to a given cluster if the assignment has a *higher probability* than a given threshold. It turns out that only clusters with qualified coherence from the data set are extracted. Therefore, users do not need to input the number, K of clusters.

In any case, K-means algorithm and its derivatives require either the number of clusters or some coherence threshold. The clustering process is like a *black box*. Therefore, they are not flexible to the local structures of the data set, and can hardly support interactive exploration for coherent expression patterns.

SOM (Self-Organizing Maps) were developed on the basis of a single layered neural network [30]. Elements, usually of high dimensionality, are mapped onto a set of neurons organized with low dimensional structures, e.g., a two dimensional $p \times q$ grid. Each neuron is associated with a reference vector, and each element is mapped to the neuron with the closest reference vector. During the clustering process, each data object acts as a training sample that directs the movement of the reference vectors towards the denser areas of the input vector space, so that those reference vectors are trained to fit the distributions of the input data set. When the training is complete, clusters are identified by mapping all data points to the output neurons.

One of the remarkable features of SOM is that it allows one to impose partial structure on the clusters, and arranges similar patterns as neighbors in the output neuron map. This feature facilitates an easy visualization and interpretation of clusters, partly supporting the explorative analysis of gene expression patterns. However, similar to the K-means algorithm, SOM requires the number of clusters, which is typically unknown in advance for gene expression data.

In [31] the SOM algorithm is applied to study hematopoietic differentiation. The expression patterns of 1,036 human genes are mapped to a 6 × 4 SOM. The SOM organizes genes into biologically relevant clusters that suggest novel hypotheses about hematopoietic differentiation and this provides interesting insights into the mechanism of differentiation.

Recently, the Department of Information Technology (National University of Ireland) has developed the system SOMBRERO (Self-Organizing Map for Biological Regulatory Element Recognition and Ordering). SOMBRERO finds regulatory binding sites by using SOM to find over-represented motifs in a set of DNA sequences [32,33]. It includes prior knowledge in the initialization phase that significantly improves accuracy when known motifs are present in the input data, while accuracy is not negatively affected for the discovery of novel motifs.

MBA (Model Based Algorithms) [34] provides a statistical framework to model the cluster structure of gene expression data. The data set is assumed to come from a mixture of underlying probability distributions, with each component corresponding to a different cluster. The goal is to estimate the parameters $\Theta = \{\theta_i | 1 \leq i \leq k\}$ and $\Gamma = \{\gamma_r^i | 1 \leq i \leq k; 1 \leq r \leq n\}$ that maximize the likelihood $L_{mix}(\Theta, \Gamma) = \sum_{i=1}^{k} \gamma_r^i f_i(x_r, \theta_i)$, where n is the number of elements, k is the number of components, x_r is a data object (i.e., a gene expression profile),

$f_i(x_r, \theta_i)$ is the density function of x_r in component C_i with some unknown set of parameters θ_i; γ_r^i represents the probability that x_r belongs to C_i.

An important advantage of MBA approaches is that they provide an estimated probability that an elements belongs to a given cluster. The probabilistic feature of MBA is particularly suitable for gene expression data because it is typical for a gene to participate multiple cellular processes, so that a single gene may have a high correlation with two different clusters. Moreover, MBA does not need to define a distance (or similarity) between two gene profiles. Instead, the measure of coherence is inherently embedded in the statistical framework.

However, MBA assumes that the data set fits a specific distribution. This may not be true and there is currently no well-established general model for gene expression data. Several MBA approaches claim a multivariate Gaussian distribution. Although the Gaussian model works well for gene-sample data where the expression levels of genes are measured under a collection of samples, it may not be effective for time-series data (the expression levels of genes are monitored during a continuous series of time points).

To better describe the gene expression dynamics in time-series data, several new models have been introduced. For example, each gene expression profile can be modeled as a cubic spline so that each time point influences the overall smooth expression curve [35]. In addition, time-series may follow an autoregressive model, where the value of the series at time t is a linear function of the values at several previous time points [36].

Time series data are often treated with *Hidden Markov model* (HMM) as an extension of a Markov model, in which a state has a probability of emitting some output. Formally, an HMM is a finite state machine with probabilities for each transition, that is, a probability of the next state is given by the current state. The states are not directly observable; instead, each state produces one of the observable outputs with a certain probability.

HMM's can be used to represent the alignment of multiple sequences or sequence segments by attempting to capture common patterns of residue conversion. They are widely used in the analysis of biological sequences to take in account for the dependencies in time-series bio-data [37].

GBA(Graph Based Algorithm) models a gene expression data set as a undirected weighted graph $G(V, E, W)$, where each gene is represented by a vertex $v \in V$, an arc $(x, y) \in E$ connects a pair of genes $x, y \in V$ with a weight, $W(x, y)$, based on the similarity between the expression patterns of x and y. The similarity is often normalized to $[0, 1]$, where 0 imply the non existence of an arc, and 1 the perfect fit of two genes. The problem of clustering a set of genes is then isomorph to some classical graph-theoretical problems, such as searching for the *minimum cut* [38], the *minimum spanning tree* [39], or the *maximum cliques* [23] in graph G. Other algorithms recursively split G into a set of *Highly Connected Components* (HCC) along the minimum cut, and each HCC is considered as a cluster. For example, the algorithm CLICK (CLuster Identification via Connectivity Kernels) sets up a statistic framework to measure the coherence within a subset of genes and determine the criterion to stop the recursive splitting process.

The graph-based algorithms stem from some classical graph theoretical problems. Although with solid mathematical ground, they may not be suitable for gene expression data without adaption. For example, in gene expression data, groups of co-expressed genes may be highly connected by a large amount of *intermediate* genes. In this case, the approaches based on minimum spanning tree and minimum cut may lead to clusters including genes with incoherent profiles but highly connected by a series of *intermediate* genes [40].

HA's (Hierarchical Algorithms) fall in two categories:

Agglomerative (i.e., bottom-up approach) that initially regards each data object as an individual cluster. Agglomerative approaches merge, at each step, the closest pair of clusters until all the groups are merged into one cluster.

Divisive (i.e., top-down approach) that starts with one cluster containing all the data objects. Divisive approaches iteratively split clusters until each cluster contains only one data object or certain stop criterion is met. For divisive approaches, the essential problem is to decide how to split clusters at each step.

An example of agglomerative hierarchical clustering is proposed in [41]; it combines tree-structured vector quantization and partitive K-means clustering. This hybrid technique reveals clinically relevant clusters in large publicly available data sets. The system is less sensitive to data preprocessing and data normalization. Moreover, results obtained have strong similarities with those obtained by self-organizing maps.

A clique graph is an undirected graph that is the union of disjoint complete graphs. In [23] the idea of a corrupted clique graph data model is introduced. Clustering a dataset is equivalent to identifying the original clique graph from the corrupted version with as few errors as possible. CAST Algorithm is an example of graph theoretic approach that relies on the concept of a clique graph and uses a divisive clustering approach. Thus, the model assumes that there is a *true biological partition of the genes into disjoint clusters bases on the functionality of the genes*. In [42] an enhanced version of CAST, called E-CAST, is described. The main difference with CAST is the use of a dynamic threshold is introduced. The threshold value is computed at the beginning of each new cluster.

PBCA (Pattern-based Clustering Algorithms) have been proposed to capture coherence exhibited by a subset of genes on a subset of attributes. This approach takes in account the fact that in molecular biology any cellular process may take place only in a subset of the attributes (samples or time points). For example, in [43] the concept of *bicluster* to measure the coherence between genes and attributes is introduced. *Biclustering* was first introduced in [44,45] , it finds a partition of the vectors and a subset of the dimensions such that the projections along those directions of the vectors in each cluster are close to one another. Then the problem requires to cluster vectors and dimensions simultaneously, thus the name biclustering.

The complexity of the biclustering problem depends on the exact problem formulation, and particularly on the merit function used to evaluate the quality of a given bicluster. The exact solution of biclustering is NP-complete so that clever heuristics are considered to solve it with small lossy of information. For example, in [46] a stochastic algorithm based on Simulated Annealing [47] is presented and validated on a variety of data-sets showing that Simulated Annealing find significant biclusters in many cases. Evolutionary algorithms have been used in [48] to implement biclustering clustering of gene expression on yeast and lymphoma data-sets.

A multi-objective evolutionary clustering for gene expression data is described in [49]. Here, a set of solutions, which are all optimal and involving trade-offs between conflicting objectives, are considered. Unlike single-objective optimization problems, the multiple-objective approach tries to optimize $m \geq 2$ conflicting solutions evaluated by fitness functions. Validation was carried out on microarray data consisting of a benchmark gene expression dataset, viz., Yeast.

ECA (Evolutive Clustering Algorithms) have been recently proposed to analyze biological data to overcome both the computational complexity of greedy algorithms and to improve the space solution scan.

In [50] the *GenClust* (Genetic Clustering) algorithm has been introduced for clustering of gene expression data. GenClust has two key features: (a) a novel coding of the search space that is simple, compact and easy to update; (b) it can be naturally used in conjunction with *data driven* internal validation methods.

In [51] a new classifier, based on fuzzy-integration schemes, is introduced. Schemes are controlled by a genetic optimization procedure. Two versions of integration are proposed and validated by experiments on real data representing: (a) biological cellsBreast cancer databases from the University of Wisconsin and Waveform ((ftp://ftp.ics.uci.edu/pub/machine-learning-databases)); (b) Urine analysis cells database kindly provided by IRIS Diagnostic, CA, USA. Comparison with feed-forward neural network and Support Vector Machine classifiers have been considered for comparison. Results show the good performance and robustness of the integrated classifier.

In [52] an incremental Genetic K-means Algorithm (IGKA) is presented. IGKA is an extension of previously proposed genetic algorithm to improve the computation of K-means algorithm. The main idea of IGKA is to calculate the objective value *Total Within-Cluster Variation* and to cluster centroids incrementally whenever the mutation probability is small. IGKA always converges to the global optimum. Experiments indicate that IGKA algorithm has a better time performance when the mutation probability decreases to some point.

4 Case Studies

4.1 An Example of Evolutive Algorithm: GenClust

GenClust [50] is one of the most recent evolutive clustering algorithm, that can be seen as a genetic variant of ISODATA and it is an incarnation of the technique devised in [53] for clustering based on Genetic Algorithms. The main difference between Genetic algorithms for clustering already present in the literature [54] and *GenClust* consists in the generated solution space. GenClust codes a solution (label) for each element instead of coding the whole partition of the data set.

Such much simpler coding technique allows a very efficient update of the state of the algorithm and also guarantees a more efficient search of the solution space.

The general idea behind *GenClust* is quite simple. The algorithm proceeds in stages; at each stage, t, a partition $\mathcal{P}_t$ of $X \subset \mathbb{R}^d$ into K classes $C_1, C_2, \cdots, C_K$ is generated. The initial partition, $\mathcal{P}_0$, is obtained by a random assignment of elements to classes or by the computation of the partition through another clustering algorithm. Based on $\mathcal{P}_t$ and using genetic operators (cross-over and mutation) and a suitable fitness function, the algorithm computes $\mathcal{P}_{t+1}$. Note that, there is no guarantee that the new partition is such that $VAR(\mathcal{P}_{t+1}) \leq VAR(\mathcal{P}_t)$. Where, $VAR(\mathcal{P})$ denoted the internal partition variance.

Each element $x \in X$ is coded via a 32 bit string α_x (referred to as *chromosome*). The chromosome encodes the class that x belongs to in a partition using the 8 least significant bits. We refer to it as the label λ_x. The remaining 24 bits give the position of x within its cluster, referred to as pos_x. The chosen coding is compact and easy to handle and allows to represents up to 256 classes and data sets of size up to 1.6793.604 elements. These values are adequate for real applications. The genetic operators of one point crossover and mutation are applied to each chromosome with probability 0.9 and 0.1, respectively. The fitness function of individual (x, λ) in partition $\mathcal{P}$ is given by:

$$f((x,\lambda)) = \sqrt{\frac{1}{d}\sum_{j=1}^{d}\frac{(x_j - \mu_j^{\lambda})^2}{\max(x_j, \mu_j^{\lambda}))^2}} \tag{1}$$

where, μ^{λ} is centroid of the cluster λ in $\mathcal{P}$.

GenClust has been validated using the FOM methodology, conceived for gene expression data [58]. GenClust has been validated on several set of biological data; among them the Rat Central Nervous System data set [59], Yeast Cell Cycle [60], Reduced Yeast Cell Cycle [61], and Peripheral Blood Monocytes [62].

4.2 Analysis of Genes Expression Patterns

Analyzing coherent gene expression patterns is an important task in bioinformatics research and biomedical applications. This issue is important because co-expressed genes may belong to the same or similar functional categories and indicate co-regulated families, while coherent patterns may characterize important cellular processes and suggest the regulating mechanism in the cells. Examples of co-expressed gene groups are shown in Figure 4. adapted or proposed to identify clusters of co-expressed genes and recognize coherent expression patterns as the centroids of the clusters. However, the interpretation of co-expressed genes and coherent patterns mainly depends on the knowledge domain, which presents several challenges for coherent pattern mining. In such cases, the design of interactive clustering systems may be useful. An examples of interactive exploration system is GeneX (Gene eXplorer) for mining coherent expression patterns [63]. GeneX is composed of a preprocessing module to perform to estimate missing values, logarithmic transformation and standardization of each

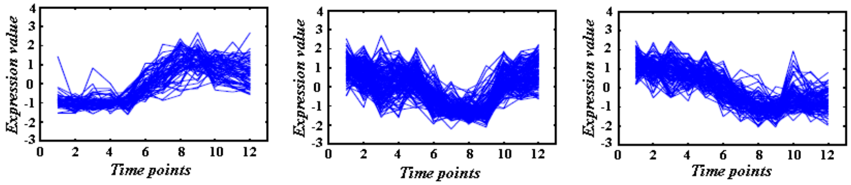
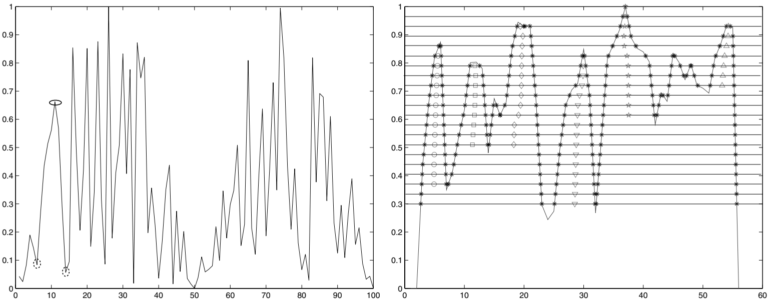


Fig. 4. Examples of co-expressed gene groups.

gene expression profile. A pattern manager module allows users to explore coherent patterns in the data set and save/load the coherent patterns. A working zone integrates parallel coordinates, coherent pattern index graphs (pulses of coherent pattern index graphs indicate the potential existence of a coherent pattern), and tree views. For example, users can select a node in the tree view, then the working zone will display the corresponding expression profiles and coherent pattern index graph.

4.3 Study of Proteins Sequences

The analysis of stochastic signals aims to both extract *significant* patterns from noisy background and to study their spatial relations (periodicity, long term variation, burst, etc.). Examples of such kind of data are protein-sequences in molecular biology where protein folding are studied [64] and the positioning of nucleosomes along chromatin [55]. The analysis carried out in both cases has been tackled by using probabilistic networks (e.g., Hidden Markov Models [13], Bayesian networks,...). However, probabilistic networks may suffer of high computational complexity, and results can be biased from locality that depends on the memory steps they use [56]. In [57] a Multi-Layers Model (MLM) is proposed that is computational efficient, providing a better structural view of the input data. The MLM consists in the generation of several sub-samples from the input signal. For example, in the case of input signal fragment, representing the *Saccharomyces cerevisiae* microarray data, each value in the x axes represents a spot on the microarray and its intensity is the log ratio Green/Red (see Figure 5a). The problem is the identification of particular patterns in the DNA called *nucleosome* and *linker* regions. Nucleosomes correspond to peaks of about 140 base pairs long, or six to eight microarray spots (black circle in Figure 5a), surrounded by lower ratio values corresponding to linker regions (marked by dashed circles). The multi-layer view is obtained by intersecting the signal with horizontal lines, each one representing a threshold value t_k (see Figure 5b). The persistence of the signal at increasing threshold values together with its width and power is considered to perform the discrimination of linkers from nucleosomes. From the biological point of view, the accurate positioning of nucleosomes provides useful information regarding the regulation of gene expression



(a) (b)

Fig. 5. An example of analysis by *MLM*: (a) Input Signal; (b) Pattern identification and extraction

in eukaryotic cells. In fact, how eukaryotic DNA is packaged into a highly compact and dynamic structure called chromatin may provide information about a variety of diseases, including cancer.

5 Final Remarks and Perspectives

A survey of current data analysis methods in bioinformatics has been provided. Main emphasis has been given to clustering techniques because of their impact in many biological applications, as the mining of biological data. The topic is so wide that several aspects have been omitted or not fully developed. The aim was to introduce main ideas and to stimulate new research directions. Challenges with bioinformatics are the need to deal with interdisciplinary directions, the difficulties in the validation of development data analysis methods, and, more important to be addressing important biological problems.

References

1. Brudno, M., Malde, S., Poliakov, A.: Glocal alignment: finding rearrangements during alignment. Bioinformatics 19(1), 54–62 (2003)
2. Rogic, S.: The role of pre-mRNA secondary structure in gene splicing in Saccharomyces cerevisiae, PhD Dissertation, University of British Columbia (2006)
3. Bourne, P.E., Shindyalov, I.N.: Structure Comparison and Alignment. In: Bourne, P.E., Weissig, H. (eds.) Structural Bioinformatics, Wiley-Liss, Hoboken, NJ (2003)
4. Zhang, Y., Skolnick, J.: The protein structure prediction problem could be solved using the current PDB library. Proc. Natl. Acad. Sci. USA 102(4), 1029–1034 (2005)
5. Gould, S.J.: The Structure of Evolutionary Theory. Belknap Press (2002)
6. Matsuda, T., Motoda, H., Yoshida, T., Washio, T.: Mining Patterns from Structured Data by Beam-wise Graph-Based Induction. In: Lange, S., Satoh, K., Smith, C.H. (eds.) DS 2002. LNCS, vol. 2534, pp. 422–429. Springer, Heidelberg (2002)

7. Schaffer, A.A., Aravind, L., Madden, T.L., Shavirin, S., Spouge, J.L., Wolf, Y.I., Koonin, E.V., Altschul, S.F.: Improving the accuracy of PSI-BLAST protein database searches with composition-based statistics and other refinements. Nucleic Acids Res. 29(14), 2994–3005 (2001)
8. `http://www.netsci.org/Resources/Web/small.html`
9. Karp, P.D., Riley, M., Saier, M., Paulsen, I.T., Paley, S.M., Pellegrini-Toole, A.: The EcoCyc and MetaCyc databases. Nucleic Acids Research 28, 56–59 (2000)
10. Vert, J.-P.: Support Vector Machine Prediction of Signal Peptide Cleavage Site Using a New Class of Kernels for Strings. In: Proceedings of the Pacific Symposium on Biocomputing, vol. 7, pp. 649–660 (2002)
11. Aerts, S., Thijs, G., Coessens, B., Staes, M., Moreau, Y., De Moor, B.: Toucan: deciphering the cis-regulatory logic of coregulated genes. Nucleic Acids Research 31(6), 1753–1764 (2003)
12. `http://homes.esat.kuleuven.be/~saerts/software/toucan.php`
13. Cappé, O., Moulines, E., Rydén, T.: Inference in Hidden Markov Models. Springer, Heidelberg (2005)
14. Kielbasa, S.M., Blüthgen, N., Sers, C., Schäfer, R., Herze, H.: Prediction of Cis-Regulatory Elements of Coregulated Genes Szymon. Genome Informatics 15(1), 117–124 (2004)
15. Cheng Cheung, L.-L., Siu-Ming Yiu, D.W.: Approximate string matching in DNA sequences. In: Proceedings DASFAA 2003, pp. 303–310 (2003)
16. Myers, G.: A fast bit-vector algorithm for approximate string matching based on dynamic programming. Journal of the ACM 46(3), 395–415 (1999)
17. Aoki, K.F., Yamaguchi, A., Okuno, Y.: Effcient Tree-Matching Methods for Accurate Carbohydrate Database Queries. Genome Informatics 14, 134–143 (2003)
18. Gusfield, D.: Algorithms on Strings, Trees and Sequences: Computer Science and Computational Biology, The Press Syndacate of the University of Cambridge, UK (1999)
19. Taylor, W.R.: Protein Structure Comparison Using Bipartite Graph Matching and Its Application to Protein Structure Classification. Molecular & Cellular Proteomics 1(4), 334–339 (2002)
20. Yang, Q., Sze, S.-H.: Path Matching and Graph Matching in Biological Networks. Journal of Computational Biology 14(1), 56–67 (2007)
21. Sholom, M.W., Indurkhya, N.: Predictive Data-Mining: A Practical Guide. Morgan Kaufmann, San Francisco (1998)
22. Tana, A.H., Panb, H.: Predictive neural networks for gene expression data analysis. Neural Networks 18, 297–306 (2005)
23. Ben-Dor, A., Shamir, R., Yakhini, Z.: Clustering gene expression patterns. Journal of Computational Biology 6(3/4), 281–297 (1999)
24. Eisen, M.B., Spellman, P.T., Brown, P.O., Botstein, D.: Cluster analysis and display of genome-wide expression patterns. Proc. Natl. Acad. Sci. USA 95(25), 14863–14868 (1998)
25. MacQueen, J.B.: Some Methods for classification and Analysis of Multivariate Observations. In: Proceedings of 5-th Berkeley Symposium on Mathematical Statistics and Probability, Berkeley, vol. 1, pp. 281–297. University of California Press (1967)
26. Tavazoie, S., Hughes, J.D., Campbell, M.J., Cho, R.J., Church, G.H.: Systematic determination of genetic network architecture. Nature Genet. 22(3), 281–285 (1999)
27. Herwig, R., Poustka, A.J., Muller, C., Bull, C., Lehrach, H., O'Brien, J.: Large-Scale Clustering of cDNA Fingerprinting Data. Genome Research 9(11), 1093–1105 (1999)

28. Heyer, L.J., Kruglyak, S., Yooseph, S.: Exploring expression data: identification and analysis of coexpressed genes. Genome Research 9(11), 1106–1115 (1999)
29. De Smet, F., Mathys, J., Marchal, K., Thijs, G., De Moor, B., Moreau, Y.: Adaptive quality-based clustering of gene expression profiles. Bioinformatics 18, 735–746 (2002)
30. Kohonen, T.: Self-Organization and Associative Memory. Springer, Berlin (1984)
31. Tamayo, P., Slonim, D., Mesirov, J., Zhu, Q., Kitareewan, S., Dmitrovsky, E., Lander, E.S., Golub, T.R.: Interpreting patterns of gene expression with self-organizing maps: Methods and application to hematopoietic differentiation. Proc. Natl. Acad. Sci. USA 96(6), 2907–2912 (1999)
32. `http://bioinf.nuigalway.ie/sombrero/`
33. Mahony, S., Golden, A., Smith, T.J., Benos, P.V.: Improved detection of DNA motifs using a self-organized clustering of familial binding profiles. Bioinformatics 21(Suppl 1), 283–291 (2005)
34. Yeung, K.Y., Fraley, C., Mura, A., Raftery, A.E., Ruzzo, W.L.: Model-based clustering and data transformations for gene expression data. Bioinformatics 17, 977–987 (2001)
35. Yeang, C.-H., Jaakkola, T.: Time Series Analysis of Gene Expression and Location Data. In: Proceedings of the Third IEEE Symposium on BioInformatics and BioEngineering (BIBE 2003), pp. 1–8 (2003)
36. Ramoni, M.F., Sebastiani, P., Kohane, I.S.: Cluster analysis of gene expression dynamics. Proc. Natl. Acad. Sci. USA 99(14), 9121–9126 (2002)
37. Koski, T.T.: Hidden Markov Models for Bioinformatics. Series: Computational Biology, vol. 2. Springer, Heidelberg (2002)
38. Hartuv, E., Shamir, R.: A clustering algorithm based on graph connectivity. Information Processing Letters 76(4/6), 175–181 (2000)
39. Xu, Y., Olman, V., Xu, D.: Clustering gene expression data using a graph-theoretic approach: an application of minimum spanning trees. Bioinformatics 18, 536–545 (2002)
40. Jiang, D., Pei, J., Zhang, A.: Interactive Exploration of Coherent Patterns in Time-Series Gene Expression Data. In: Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2003), Washington, DC, USA, pp. 24–27 (2003)
41. Sultan, M., Wigle, D.A., Cumbaa, C.A., Marziar, M., Glasgow, J., Tsao, M.S., Jurisca, J.: Binary tree-structured vector quantization approach to clustering and visualizing microarray data. Bioinformatics 18(1), 111–119 (2002)
42. Bellaachia, A., Portnoy, D., Chen, Y., Elkahloun, A.G.: E-CAST: a data mining algorithm for gene expression data. In: Proceedings of the ACM SIGKDD Workshop on Data Mining in Bioinformatics (BIOKDD 2002), pp. 49–54 (2002)
43. Cheng, Y., Church, G.M.: Biclustering of expression data. In: Proceedings of the Eighth International Conference on Intelligent Systems for Molecular Biology (ISMB), vol. 8, pp. 93–103 (2000)
44. Mirkin, B.: Mathematical Classification and Clustering. Kluwer Academic Publishers, Dordrecht (1996)
45. Van Mechelen, I., Bock, H.H., De Boeck, P.: Two-mode clustering methods:a structured overview. Statistical Methods in Medical Research 13(5), 363–394 (2004)
46. Bryan, K., Cunningham, P., Bolshakova, N.: Biclustering of Expression Data Using Simulated Annealing. In: 18th IEEE Symposium on Computer-Baseds Medical Systems (CBMS 2005), pp. 383–388 (2005)

47. Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P.: Optimization by Simulated Annealing. Science 220(4598), 671–680 (1983)

48. Chakraborty, A., Maka, H.: Biclustering of Gene Expression Data Using Genetic Algorithm. In: IEEE Symposium on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB 2005), vol. 14(15), pp. 1–8 (2005)

49. Sushmita, M., Haider, B.: Multi-objective evolutionary biclustering of gene expression data. Pattern Recognition 39(12), 2464–2477 (2006)

50. Di Gesù, V., Giancarlo, R., Lo Bosco, G., Raimondi, A., Scaturro, D.: GenClust: A Genetic Algorithm for Clustering Gene Expression Data. BMC Bioinformatics 6(289) (2005)

51. Di Gesù, V., Lo Bosco, G.: A genetic integrated fuzzy classifier. Pattern Recognition Letters 26(4), 411–420 (2005)

52. Lu, Y., Lu, S., Fotouhi, F., Deng, Y., Brown, S.J.: Incremental genetic K-means algorithm and its application in gene expression data analysis. BMC Bioinformatics 5(172) (2004)

53. Di Gesù, V., Lo Bosco, G.: GenClust: a Genetic Algorithm for Cluster Analysis. In: Proc. ADA III, pp. 12–18 (2004)

54. Jain, A.K., Murty, M.N., Flynn, P.J.: Data Clustering: A Review. ACM Computing Surveys 31(3), 264–323 (1999)

55. Yuan, G.C., Liu, Y.J., Dion, M.F., Slack, M.D., Wu, L.F., Altschuler, S.J., Rando, O.J.: Genome-Scale Identification of Nucleosome Positions in S. cerevisiae. Science 309, 626–630 (2005)

56. Delcher, A.L., Kasif, S., Goldberg, H.R., Hsu, W.H.: Protein secondary structure modelling with probabilistic networks. In: Proc. of Int. Conf. on Intelligent Systems and Molecular Biology, pp. 109–117 (1993)

57. Corona, D., Di Gesù, V., Lo Bosco, G., Pinello, L., Yuan, G.-C.: A new Multi-Layers Method to Analyze Gene Expression. In: Proc. KES 2007. LNCS, Springer, Heidelberg (in press, 2007)

58. Yeung, K.Y., Haynor, D.R., Ruzzo, W.L.: Validating clustering for gene expression data. Bioinformatics 17, 309–318 (2001)

59. Somogyi, R., Wen, X., Ma, W., Barker, J.L.: Developmental kinetic of GLAD family mRNAs parallel neurogenesis in the rat Spinal Cord. Journal Neurosciences 15, 2575–2591 (1995)

60. Spellman, P., Sherlock, G., Zhang, M., et al.: Comprehensive identification of cell cycle regulated genes of the yeast Saccharomyces Cerevisiae by microarray hybridization. Journal of Mol. Biol. Cell 9, 3273–3297 (1998)

61. Cho, R.J., et al.: A genome-wide transcriptional analysis of the mitotic cell cycle. Journal of Molecular Cell 2, 65–73 (1998)

62. Hartuv, E., Schmitt, A., Lange, J., et al.: An Algorithm for Clustering of cDNAs for Gene Expression Analysis Using Short Oligonucleotide Fingerprints. Journal Genomics 66, 249–256 (2000)

63. Jiang, D., Pei, J., Zhang, A.: Towards Interactive Exploration of Gene Expression Patterns. SIGKDD Explorations 5(2), 79–90 (2003)

64. Delcher, A.L., Kasif, S., Goldberg, H.R., Hsu, W.H.: Protein secondary structure modelling with probabilistic networks. In: Proc. of Int. Conf. on Intelligent Systems and Molecular Biology, pp. 109–117 (1993)

65. Yuan, G.C., Liu, Y.J., Dion, M.F., Slack, M.D., Wu, L.F., Altschuler, S.J., Rando, O.J.: Genome-Scale Identification of Nucleosome Positions in S. cerevisiae. Science 309, 626–630 (2005)
66. Delcher, A.L., Kasif, S., Goldberg, H.R., Hsu, W.H.: Protein secondary structure modelling with probabilistic networks. In: Proc. of Int. Conf. on Intelligent Systems and Molecular Biology, pp. 109–117 (1993)
67. Corona, D., Di Gesù, V., Lo Bosco, G., Pinello, L., Yuan, G.-C.: A new Multi-Layers Method to Analyze Gene Expression. In: Proc. KES 2007 11th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems. LNCS, Springer, Heidelberg (in press, 2007)