ISTITUTO DI ANALISI DEI SISTEMI ED INFORMATICA
"Antonio Ruberti"
CONSIGLIO NAZIONALE DELLE RICERCHE

E. Weitschek, G. Fiscon, V. Fustaino, G. Felici,
P. Bertolazzi

ANALYSIS OF MICROARRAY AND RNA-SEQUENCING GENE EXPRESSION PROFILES THROUGH CLUSTERING AND CLASSIFICATION TECHNIQUES

R. 14-11 2014

Emanuel Weitschek – Istituto di Analisi dei Sistemi ed Informatica del CNR, via dei Taurini 19 - 00185 Roma, Italy. Email: emanuel.weitschek@iasi.cnr.it.

Giulia Fiscon – Istituto di Analisi dei Sistemi ed Informatica del CNR, via dei Taurini 19 - 00185 Roma, Italy. Email: giulia.fiscon@iasi.cnr.it.

Valentina Fustaino – Istituto di Analisi dei Sistemi ed Informatica del CNR, via dei Taurini 19 - 00185 Roma, Italy. Email: valentina.fustaino@iasi.cnr.it.

Giovanni Felici – Istituto di Analisi dei Sistemi ed Informatica del CNR, via dei Taurini 19 - 00185 Roma, Italy. Email: felici@iasi.cnr.it.

Paola Bertolazzi – Istituto di Analisi dei Sistemi ed Informatica del CNR, via dei Taurini 19 - 00185 Roma, Italy. Email: bertola@iasi.cnr.it.

ISSN: 1128–3378

Collana dei Rapporti dell'Istituto di Analisi dei Sistemi ed Informatica "Antonio Ruberti", CNR

viale Manzoni 30, 00185 ROMA, Italy

tel. ++39-06-77161
fax ++39-06-7716461
email: iasi@iasi.cnr.it
URL: http://www.iasi.cnr.it

Abstract

A large amount of gene expression data is available to bioinformaticians and biological scientists thanks to the great advances in microarray technology and in next generation sequencing techniques, e.g., RNA-Seq. Several biological databases and repositories containing raw and normalized gene expression profiles are accessible with up to date online services. The analysis of gene expression profiles from microarray/RNA-Seq experimental samples demands new efficient methods from statistics and computer science. In this report, two main types of gene expression data analysis are taken into account:

1. genes clustering;
2. experiments classification.

Genes clustering is the detection of gene groups that present similar patterns. Indeed, several clustering methods can be applied to group similar genes in the gene expression experiments. The aim of experiments classification is to distinguish between two or more classes to which the different samples belong (e.g., different cell types or diseased *vs* healthy samples). This work first provides a general introduction of microarray and RNA-Seq technologies. Then, gene expression profiles are investigated by means of pattern recognition methods with data mining techniques such as classification and clustering. Additionally, the integrated software packages Gene Pattern, Gene Expression Logic Analyzer (GELA), TM4 software suite, and other common analysis tools are illustrated. As gene expression profiles pattern discovery and experiment classification, the software packages are tested on three real case studies:

1. Alzheimer's diseased (AD) *vs* healthy mice;
2. Multiple Sclerosis samples;
3. Psoriasis tissues.

The performed experiments and the described techniques provide an effective overview to the field of gene expression profiles classification and clustering through pattern analysis.

1. Introduction

Due to the advances of microarray technology, but especially of the *Next Generation Sequencing* (NGS) techniques, several biological databases and repositories contain raw and normalized gene expression profiles, which are accessible with up to date online services. The analysis of gene expression profiles from microarray/RNA-Seq experimental samples demands new efficient methods from statistics and computer science.
In this work, two main types of gene expression data analysis are taken into consideration: *gene clustering* and *experiments classification*. Gene clustering is the detection of gene groups that present similar patterns. Indeed, several clustering methods can be applied to group similar genes in the gene expression experiments. On the other hand, experiments classification aims to distinguish among two or more classes to which the different samples belong (e.g., different cell types or diseased *versus* healthy samples).
This work provides first a general introduction to microarray and RNA-Seq technologies. Then, gene expression profiles are investigated by means of pattern recognition methods with data mining techniques such as classification and clustering. Additionally, the integrated software packages *GenePattern*, *Gene Expression Logic Analyzer* (GELA) [55], TM4 software suite [41], and other common analysis tools are illustrated. For gene expression profiles pattern discovery and experiment classification, the software packages are tested on four real case studies:

1. *Alzheimer's Disease* (AD) *versus* healthy mice;
2. *Multiple sclerosis* samples;
3. *Psoriasis* tissues;
4. *Breast cancer* patients.

The performed experiments and the described techniques provide an effective overview to the field of gene expression profiles classification and clustering through pattern analysis.

The rest of the work is organized as follows. In section 2, we introduce the *Transcriptome Analysis*, highlighting the widespread approaches to handle it. In section 3 and section 4, we provide an overview of the *microarray* and *RNA-Seq* technologies, respectively, including the description of the technique with its applications and the corresponding data analysis work-flow. Then, in section 5, we provide the readers with a comparative analysis between microarray and RNA-Seq technologies, by highlighting the advantages and drawbacks of each one. In section 6, we deepen the analysis of gene expression profiles data obtained from Microarray and RNA-Seq technologies, explaining how to deal with them, describing the widespread normalization techniques that have to be applied to these raw data, and providing an overview of gene clustering, experiments classification methods, and software suites that can be used to analyze them. Furthermore, in section 7, we present the application of the whole described analysis process to four real case studies (i.e., Alzheimer's disease mice, multiple sclerosis samples, psoriasis tissues, and breast cancer patients). Finally, in section 8, we draw a summary of the work.

2. Transcriptome Analysis

A *RNA molecule* is a complementary single-stranded copy of a double stranded DNA molecule, obtained after the process of transcription. It provides a critical contribution in the coding, non-coding, expression, and regulation of genes. The *transcriptome* is the set and quantity of RNA transcripts that characterize a cell in a certain development stage. Its deep analysis is critical either for detecting genes activity and quantifying their expression levels or for identifying functional elements, aiming to understand cellular development stages and mostly pathological conditions.
Several methods have been developed to address the transcriptome analysis focusing on his quantification. They can be mainly divided into two groups:

- *Hybridization-based* methods

4.

- *Sequence-based* methods

The first ones include the widespread *microarray* based methods, that exploit the hybridization techniques (i.e., based on the nucleotides property to pair with their complementary fixed on a specific support) to catch information about gene expression levels with a high-throughput and low costs (unless when investigating large genomes), showing on the other hand a high background noise, a limited dynamic detection range and the inability to identify yet unknown transcripts. Contrarily to the first ones, the second methods provide directly the desired sequence. The Sanger sequencing [42] was the initial widespread sequencing method. Recently, among the sequence-based approaches the *RNA sequencing* (RNA-Seq) [53, 34] stands out with lower time and cost. RNA-Seq performs a mapping and a transcriptome quantification relying on the novel *Next Generation Sequencing* (NGS) techniques (i.e., massively parallel sequencing). The latter guarantee an unequal high-throughput and quantitative deep sequencing that outperforms the other transcriptome analysis techniques according to several point of views, exhaustively explained in section 4.

3. Microarrays

The *microarray technology* (also known as biochips, DNA chips, gene chips, or high-density arrays) is a high-resolution method to detect the expression level of a large set of genes with a unique parallel experiment [44, 6]. Specifically, a microarray is a semiconductor device composed of a grid of multiple rows and columns. A cell of the array is associated to a probe DNA sequence, hybridized by Watson-Crick complementarity [9] to the DNA of a target gene (e.g., the mRNA sequences). The mRNA sequences contain the necessary information for the amino acids to form a given protein. The microarray experimental process is composed of the following steps. First, the mRNA sequences are amplified. Next, the mRNA sequences are fluorescently tagged. Then, the mRNA sequences are poured on the array. Next, the array is so hybridized. Finally, the array is scanned with a laser that measures the quantity of fluorescent light in each cell. This measure is the expression level of the current gene.

3.1. Applications

Microarrays have several applications ranging from the genomic and transcriptomic areas, including pharmacogenomics, drug discovery, diagnostics, and forensic purposes [23]. The widespread transcriptomic application of this technology is the measure of gene expression in different sample conditions, also called *Gene Expression Profiling* (GEP) [45]. Gene expression microarrays include case-control studies [1], body maps, tumors profiling, and outcomes prediction. Genomics microarrays are used to screen the sample for mutations, *Single Nucleotide Polymorphisms* (SNPs), *Short Tandem Repeats* (STRs), comparative genome hybridizations, genotyping, resequencing, and pathogen characterization [48].

3.2. Microarray technology

A microarray consists of a rectangular support generally constructed on glass, silicon, or plastics substrates in a 1-2 cm^2 area [22]. It is organized in grids where many thousands of different probes are immobilized in fixed locations. Each location is called *spot* and contains a pool of probes. Probes, usually synthetic oligonucleotides or larger DNA/*complementary* DNA (cDNA) fragments, are nucleic acid sequences that hybridize the labelled targets via Watson-Crick duplex formation [9]. The targets are then held into the spots. Each spot can recognize a specific target that may be either a chromosomal region for the genomic microarray, or part of a mRNA in the case of gene expression microarrays. A scanner detects the single spot fluorescence signals and converts them into digital quantifiable signals.
Microarray devices and systems are commercialized by several companies specialized on manufacturing

[1] *Case-control studies* are defined as specific studies that aim to identify subjects by outcome status at the outset of the investigation, e.g., whether the subject is diagnosed with a disease. Subjects with such an outcome are categorized as *cases*. Once outcome status is identified, controls (i.e., subjects without the outcome but from the same source population) are selected [46].

technology [48]. There are two main platforms that differ in the probe manufacturing: robotic deposition and in situ synthesis on a solid substrate. The following subsections are focused on *in situ* synthesized oligonucleotide microarray, which is the microarray technology of *GeneChip* by *Affymetrix* (affymetrix.com), the most widespread company specialized on gene expression microarray technology.

3.3. Microarray work-flow

A gene expression microarray experiment is usually composed of four laboratory steps [47]:

1. *Sample preparation and labeling*:
   the first performed steps on the biological sample are the RNA extraction, purification, reverse transcription, and labeling.

2. *Hybridization*:
   probes and targets form a double hybrid strand according to the Watson-Crick base complementarity [9].

3. *Washing*:
   the excess material is removed from the array to ensure the accuracy of the experiment by reducing not specific hybridizations. The targets remain bounded on the array.

4. *Imaging acquisition*:
   a scanner excites the fluorophores and measures the fluorescence intensity providing a color image representation as output.

5. *Data extraction*:
   specific software converts the image in numeric values by quantifying the fluorescence intensity.

For further details on the microarray experiment protocol the reader may refer to [44, 6].

4. RNA-Seq

RNA-Seq [53, 34] is a family of methods which takes advantage from NGS technologies for detecting the transcriptome expression levels with flexibility, low cost, and high reproducibility. RNA-Seq allows performing more measurements and consequently a quantitative analysis of the expression levels. Differently from the traditional sequence-based approaches, RNA-Seq yields quantitative measures instead of a qualitative and normalized description of the considered sequence. Unlike to tag-based techniques, it provides different sequences for each transcript with a single-base resolution. Indeed, RNA-Seq uses directly short fragments of the DNA molecule without any other insertion as input data and provides as output discrete measures, suitable to be simply managed. Furthermore, it is suited for two different tasks: (i) transcriptome reconstruction and analysis; (ii) quantification of the expression levels.
The RNA sequencing technique is characterized by the three following main phases that constitute the RNA-Seq work-flow:

- Sample preparation through the building of the fragments library to be sequenced;

- Sequencing;

- Sequenced reads analysis and expression levels quantification.

Even if the RNA-Seq work-flow should be characterized by the RNA extraction and its direct sequencing, actually the RNA-sequencing techniques require a pre-processing phase in order to make the sample of RNA suitable to the following part of the experimental protocol. The whole detailed procedure is sketched in Figure 1 and the three leading phases are thoroughly explained as follows.

- *Sample Preparation and Library of Fragments Building*:
  Firstly, the input RNA sequence is usually purified by the poly-A binding (step 1 and step 2 of Figure 1), in order to minimize the copious ribosomial RNA transcripts. A sequence random-size fragmentation (step 3 of Figure 1) reduces the RNA sequence into shorter segments (∼ 200-300 bases), because the next steps require smaller strings of RNA. The obtained fragments can be turned into cDNA by the reverse transcription, which performs firstly a complementary copy of the RNA strand and secondly, using as primer the remaining (partially degraded) RNA fragments, yields as output the corresponding double-stranded cDNA (step 4 of Figure 1). The step 4 is necessary because the sequencers are unable to perform directly a sequencing of a RNA molecule. The cDNAs obtained by the step 4 are amplified and the adapters are attached to one or both ends (step 5 of Figure 1) in order to build up a suitable library of fragments to be sequenced.

- *Sequencing*:
  The nucleotide sequence of short cDNA segments has to be turned into a quantified signal. A sequencer processes the amplified fragments of input cDNA (called *reads*) and identifies the successive order of the nucleotides that make up the analyzed sequence (step 6 of Figure 1). Finally, the sequencer provides the sequenced reads as output. Nowadays, widespread sequencers are those ones from Illumina and Roche.

- *Reads Analysis and Expression Levels Quantification*:
  In order to quantify the read referred to each gene, the sequenced reads are aligned and hence mapped in the reference genome (step 7 of Figure 1). The mapping aims to look for a unique region where the reads and the reference genome are identical. Starting from the genome, the transcriptome can be reconstructed by removing the intronic regions. In order to estimate the expression levels of each gene, a quantification of all reads, which correspond either to the same exon or transcript or gene is performed. The reads are summarized in a *table of counts*, where a *count* is the real output of RNA-Seq, representing the measure of the expression level for a specific gene (step 8 of Figure 1). Afterwards, there are two issues to be addressed: how to deal with the overlapping reads and how to take into account the RNA isoforms [2]. In fact, RNA-Seq is able to quantify even the expression levels of isoform transcripts. The first issue can be solved assigning to each gene only the reads that are exactly mapped in such a gene, while those that overlap more than one gene are not counted. The second issue can be handled with a scheme that counts the mapped reads shared with all isoforms of a gene. Lastly, step 9 of Figure 1 performs the measure normalization required to have comparable measurements.

5. Benefits and drawbacks of RNA-Seq and microarray technologies

In this subsection, the main *advantages* as well as the identified *drawbacks* of the RNA-Seq and microarray technologies are highlighted. As mentioned in the previous section, RNA-Seq shows several advantages, which make it the mainstream tool for deeply profiling the transcriptome and for quantifying the expression levels with an high accuracy.
The supremacy of RNA-Seq, summarized in Figure 2, is due to several features explained as follows:

- it is able to detect isoforms, allelic expression, complex transcriptomes, and currently unknown transcripts (unlike the hybridization-based approaches);
- it is characterized by a single-base resolution when determining the transcript boundaries, thanks to which several aspects of the nowadays genome annotation can be addressed;
- high-throughput quantitative measures are feasible and can be easily reproduced;
- it guarantees low background signal, lack for an upper limit for the quantification, and a large dynamic range of expression levels over which transcripts can be profiled;

[2] The different gene transcripts generated from a single gene.

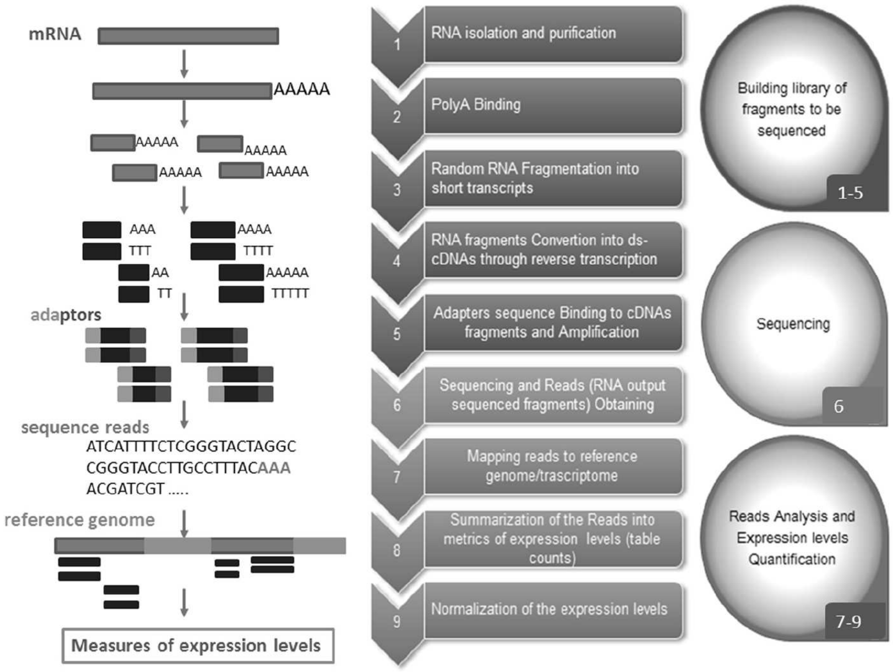
mRNA
AAAAA
AAAAA
AAAAA
AAAAA
AAAAA
AAA
TTT
AAAA
TTTT
AA
TT
AAAAA
TTTTT
adaptors
sequence reads
ATCATTTTCTCGGGTACTAGGC
CGGGTACCTTGCCTTTACAAA
ACGATCGT .....
reference genome
Measures of expression levels
1
RNA isolation and purification
2
PolyA Binding
3
Random RNA Fragmentation into short transcripts
4
RNA fragments Convertion into ds-cDNAs through reverse transcription
5
Adapters sequence Binding to cDNAs fragments and Amplification
6
Sequencing and Reads (RNA output sequenced fragments) Obtaining
7
Mapping reads to reference genome/trascriptome
8
Summarization of the Reads into metrics of expression levels (table counts)
9
Normalization of the expression levels
Building library of fragments to be sequenced
1-5
Sequencing
6
Reads Analysis and Expression levels Quantification
7-9

Figure 1: Overview of the RNA-Seq work-flow

- it presents high sensitivity and accuracy to quantify the expression levels;
- it requires a low amount of RNA (unlike microarray or other sequencing methods);
- the costs for mapping transcriptomes are relative low.

On the other hand, there are some drawbacks to be underlined:

- RNA-Seq data are complex to be analyzed and sensitive to bias;
- the cost of the sequencing platform can be too high for certain studies and laboratories;
- the requirement of an up-to-date high level IT infrastructure to store and process the large amount of data (unlike the microarrays).

Following benefits can be identified in microarray technology (summarized in Figure 3:

- the costs of a microarray experiment are low, data size is relatively small, and short time is required to perform it;
- the measures are quantitative and obtained with an high-throughput technology;
- the microarray technologies are widespread, and hence several methods are available for analyzing microarray gene expression profiles data;

The drawbacks of the microarray technology can be summarized in:

- a background signal correction of the gene expression profiles is necessary;
- the quantification of the gene expression intensity signals has upper limits;
- the conservation of the gene chips demands high caution and care;

8.

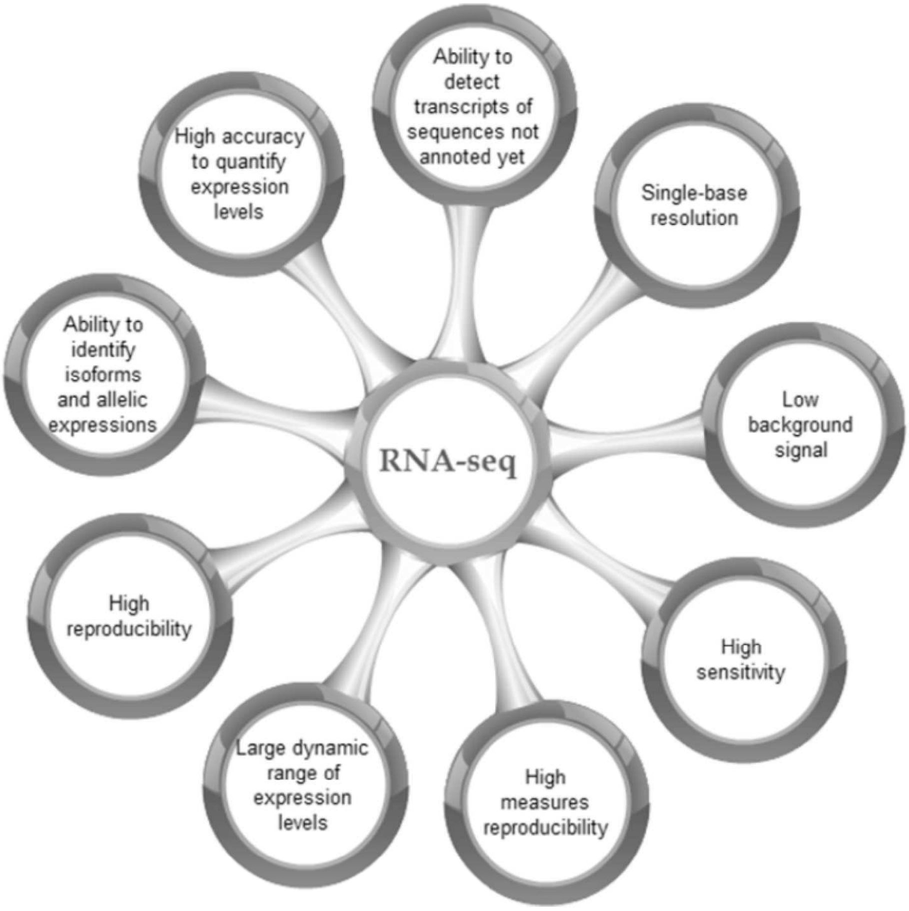
Ability to detect transcripts of sequences not annoted yet
Single-base resolution
Low background signal
High sensitivity
High measures reproducibility
Large dynamic range of expression levels
High reproducibility
Ability to identify isoforms and allelic expressions
High accuracy to quantify expression levels
RNA-seq

Figure 2: RNA-Seq benefits

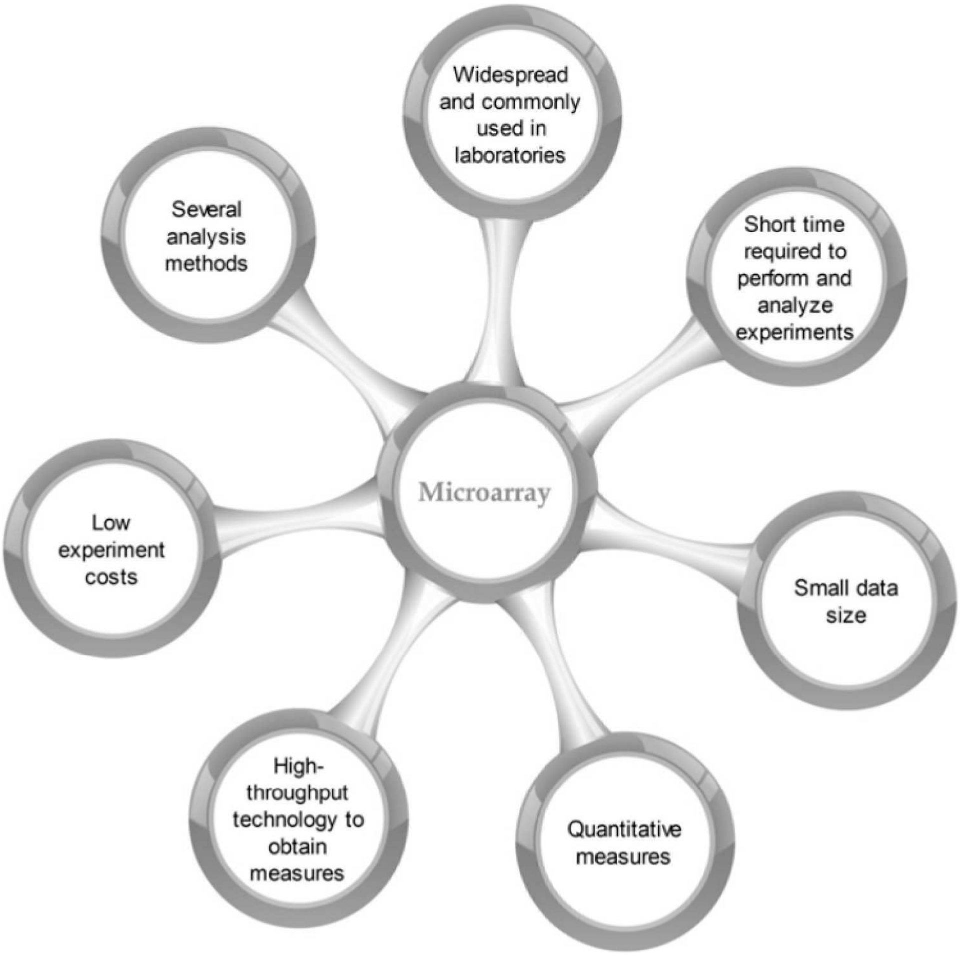
Microarray
Widespread and commonly used in laboratories
Short time required to perform and analyze experiments
Small data size
Quantitative measures
High-throughput technology to obtain measures
Low experiment costs
Several analysis methods

Figure 3: Microarray benefits

- the gene expression intensity signals may present high background noise;
- microarrays have a limited dynamic detection range of gene expression profiles and are unable to identify currently unknown transcripts.

Table 1 provides the reader with a summary of the comparison between microarrays and RNA-Seq benefits and drawbacks at a glance.

Table 1: Microarrays *versus* RNA-Seq

	Microarrays	**RNA-Seq**
High-throughput	√	√
Quantitative measures	√	√
High measure reproducibility		√
Unknown transcript identification		√
Isoform and allelic expression detection		√
Single-base resolution		√
Low background signal		√
Unlimited quantification		√
Low amount of RNA		√
Widespread	√	
Complexity		√
Handling of data	√	
Low cost	√	

6. Gene expression profiles analysis

After an adequate experimental set-up and execution, either with microarrays technology or with RNA-Seq techniques, the obtained raw data has to be processed and analyzed with effective and efficient statistical, mathematical, and computer science methods. In this section, the whole gene expression profile analysis process is described, focusing on two particular types of analysis: *genes clustering* and *experiments classification*.

6.1. Data definition

In general, a gene expression profiles data set is a collection of experiments organized in records. Each experiment has a fixed number of gene expression profiles, which can be discrete or continuous. An experiment is described by a set of gene expression profiles represented as a multidimensional vector. The data set can be stored in a matrix, where each row i represents a gene ($i = 1 \dots k$), each column j an experimental sample (Exp_j, with $j = 1 \dots n$), and each cell (i, j) the gene expression value ($expr_{(i,j)}$), as shown in Table 2. The matrix may present two heading lines containing the experimental sample identifiers and the class labels. In a case-control study, the class labels can be either *case* or *control*. In particular, we define *case* samples as subjects that are diagnosed with a disease; while we define *control*

Table 2: Gene expression profile data set

Experiment	Exp_1	$\cdots$	Exp_m	Exp_{m+1}	$\cdots$	Exp_n
Class	$Control$	$\cdots$	$Control$	$Case$	$\cdots$	$Case$
$gene_1$	$expr_{(1,1)}$	$\cdots$	$expr_{(1,m)}$	$expr_{(1,m+1)}$	$\cdots$	$expr_{(1,n)}$
$gene_2$	$expr_{(2,1)}$	$\cdots$	$expr_{(2,m)}$	$expr_{(2,m+1)}$	$\cdots$	$expr_{(2,n)}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$gene_k$	$expr_{(k,1)}$	$\cdots$	$expr_{(k,m)}$	$expr_{(k,m+1)}$	$\cdots$	$expr_{(k,n)}$

samples as healthy subjects without the disease, but extracted from the same source population of the cases. Each cell of the matrix (expr) contains the gene expression value.

6.2. Data Analysis

A typical gene expression profiles analysis, whose aim is to cluster similar genes and to classify the experimental samples, is composed of following steps:

1. *Normalization and background correction*;
2. *Genes clustering*;
3. *Experiments classification.*

Normalization is a preprocessing step that is required to transform the raw data into reliable and comparable gene expression measurements.

Genes clustering is the detection of gene groups that present similar patterns [40]. In [26] and [56] different clustering methods, that can be applied to group similar genes in microarray experiments, are described. In [3] the authors introduce the biclustering technique: here the genes and the experimental samples are clustered simultaneously in order to find groups of genes that are related to a particular class (e.g., in a case-control study); ideally two biclusters should appear, one containing the case samples and the associated genes and the other the control samples and the associated genes.

In *experiments classification*, the aim is to separate either the samples in classes (e.g., case *versus* control) or different cell types [27]. The experimental sample separation should be performed with classification models composed of human interpretable patterns, e.g., logic formulas ("if-then rules").

6.3. Normalization and background correction

In this subsection, we explain the normalization preprocessing step, highlighting the widespread techniques to normalize the gene expression profiles data provided by microarray, as well as RNA-Seq technologies.

6.3.1. Microarray normalization methods

Microarrays are subject to different sources of variation and noise that make data not directly comparable and that originate from array manufacturing process, sample preparation and labeling, hybridization, and quantification of spot intensities. Normalization methods aim to handle these systematic errors and bias introduced by the experimental platforms. Therefore, normalization is a basic prerequisite for microarray data analysis and for the quantitative comparisons of two or more arrays. Every normalization method includes a specific background correction step that is used to remove the contribution of unspecific background signals from the spot signal intensities. Furthermore, each method computes a measure of the expression level from the probe intensities that represent the amount of the corresponding mRNA [15].

The most commonly used normalization methods are *Microarray Analysis Suite* 5.0 (MAS 5.0) [1], *dChip* [32], and *Robust Multi-array Average* (RMA) [25].

In MAS 5.0, the normalization step operates on the gene-level intensities and applies a simple linear scaling. The intensities between two or more arrays are mathematically represented on a straight line with a zero y-intercept. The slope of the line is multiplied with a scaling factor to let the mean of the experiment chip be equal to the one of the baseline chip (reference).

Also in *dChip* the normalization is applied on the gene-level intensities, but the intensities are represented on non-linear smooth curves. A rank invariant set is used to force a given number of non-differentially expressed genes to have equal values among the data sets [43]. Otherwise, RMA employs a probe-level quantile normalization in multiple arrays by forcing the quantile values of the probe intensity distribution for each array to be equal [4]. Microarray normalization methods are available at the open source software for bioinformatics R/Bioconductor (www.bioconductor.org) [20], *GenePattern* [39] and in *Microarray Data Analysis System* (MIDAS) from the TM4 software suite [41].

6.3.2. RNA-Seq normalization methods

As mentioned in section 4, RNA-Seq returns the total number of reads aligned to each gene (called *counts*), which hence represent the mapping into the reference transcriptome. The counts are stored in a matrix (Table 3), whose rows and columns contain the genes and the sequenced samples, respectively. Starting from the raw output, a normalization step is required in order to remove the inherent bias of the sequencing (e.g., the length of RNA transcripts and the depth of sequencing for a sample). Indeed, the metrics of normalization make the measurements directly comparable and the expression levels estimation worthwhile. First of all, the number of reads mapped to a gene can be conditioned by the length of the reference transcripts: trivially, shorter transcripts are more likely to have less mapped reads than longer ones. Moreover, because of the sequencing depth, differences can occur that may make the comparisons between two samples inconsistent.

Table 3: Table of RNA-Seq counts

Experiment	Exp_1	$\cdots$	Exp_m	Exp_{m+1}	$\cdots$	Exp_n
Class	$Control$	$\cdots$	$Control$	$Case$	$\cdots$	$Case$
$gene_1$	$count_{(1,1)}$	$\cdots$	$count_{(1,m)}$	$count_{(1,m+1)}$	$\cdots$	$count_{(1,n)}$
$gene_2$	$count_{(2,1)}$	$\cdots$	$count_{(2,m)}$	$count_{(2,m+1)}$	$\cdots$	$count_{(2,n)}$
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$gene_k$	$count_{(k,1)}$	$\cdots$	$count_{(k,m)}$	$count_{(k,m+1)}$	$\cdots$	$count_{(k,n)}$

Among the normalization methods, we focus on the two widespread techniques based on the *Reads Per Kilobase per Million mapped* (RPKM) [35] and the *Fragments Per Kilobase per Million mapped* (FPKM) [51] values. The RPKM normalizes the read counts according to their length and the counts sum for each sample. Specifically, let R be the count of mapped reads into the genes exon, N_r the total number of mapped reads, and L the length of all exons of such a gene (in terms of number of base-pairs), then the value is calculated with the following formula:

$$RPKM = \frac{R}{N_r \cdot L} \cdot 10^9 \tag{1}$$

The FPKM computes the expected fragments (pairs of reads) per kilobase of the transcript per million sequenced fragments. Specifically, let N_f be the count of mapped fragments, F the total number of the mapped fragments, and L the length of all exons of such a gene (in terms of number of base-pairs), then the value is calculated with the following formula:

$$FPKM = \frac{F}{N_f \cdot L} \cdot 10^9 \tag{2}$$

The RPKM value is really close to the FPKM value, indeed, if all mapped reads are paired, the two values will be coincident. However, the latter is able to handle a higher number of reads from single molecules. Finally, it should be underlined that a normalization of an RNA-Seq experiment does not imply an alteration of the raw data, contrarily to the normalization techniques of microarray data. The major RNA-Seq software analysis tools, like MEV [24] from TM4 software suite, include automatic conversion from raw data counts in RPKM and FPKM normalized values.

6.4. Genes clustering

After the normalization procedure, the real analysis process begins with *gene clustering*. A large number of genes (ten to forty thousands) are present in every sample of gene expression profiles data, and the extraction of a subset of representative genes, which are able to characterize the data, is a crucial analysis step. Gene cluster analysis is a technique whose aim is to partition the genes into groups, based on a distance function, such that similar or related genes are in same clusters, and different or unrelated genes are in distinct ones [16]. Cluster analysis groups the genes by analyzing their expression profiles, without considering any class label - if available. For this reason, it is also called *unsupervised learning*: it creates an implicit labeling of genes that is given by the clusters and is derived only from their expression profiles. For defining the genes partition, a *distance measure* (or distance metric) between them has to be defined. The measure has to maximize the similarity of the gene expression in the same clusters and the dissimilarity of the gene expression in different ones. Various metrics and distance functions can be adopted, the widespread ones are: *Minkowski distance* [8], that of order 1 is called *Manhattan distance* [49] and that of order 2 is called *Euclidean distance* [12], [11]; *Mahalanobis distance* [50]; and *Correlation distance* [31]. The adoption of a measure of the similarity of the genes over the samples is already an important tool of analysis; in fact, we can just examine one gene and verify what are those that exhibit a similar behavior. In this context, the correlation distances are of particular relevance (the most used is Pearson's correlation)that are designed to measure the relative variations of the genes over the sample and do not take into account the scale of the expression values.

Clustering is indeed a more sophisticated analysis that can be performed, but it is still based on the concept of similarity between genes as measured by a distance function. Clustering algorithms can be divided in two main groups: *partition algorithms* and *hierarchical algorithms*. In the partition algorithms, the objects are divided into a given number of clusters. In the hierarchical algorithms, the objects are grouped in clusters starting from a initial clusters (that contains all the objects) or vice versa. The most important partition clustering algorithm is *K-means* [33]. Widely used hierarchical clustering algorithms are *AGglomerative NESting* (AGNES) [29] and *DIvisive ANAlysis* (DIANA) [29]. A new type of clustering algorithm, particularly designed for the gene expression profiles, is presented in the software GELA [54],[55], where the clusters are selected by a discretization procedure [3].

In [3] a biclustering method is described, where the genes and the experimental samples are clustered simultaneously. This is a very important approach, as the clusters may contain the genes associated to experiments of a particular class (e.g., a cluster may group together diseased experimental samples and the corresponding involved genes).

After the clusters have been computed, a cluster validation step can be performed: in this step the clusters are validated with statistical measures, like entropy, purity, precision, recall, and correlation. According to these measures the clustering algorithm may be adjusted or fine tuned.

Finally, the results have to be presented to the user and knowledge has to be transferred with a graphical interface and cluster lists of similar genes.

Another important step that can be taken into account is the reduction of the dimensions of the space where the data lies. There are several methods that may perform this step based on the projection of the data point onto a subspace of smaller dimension, either with respect to the number of genes or to the number of samples; the most popular among such methods is the *Principal Component Analysis* (PCA) [28]. PCA identifies a projection onto a small number of *components*, or *factors*, that are obtained as linear combination of the original ones. For example, if PCA is applied on the genes, the sample points are represented in a new space where each coordinate is obtained as a linear combination of the original genes (symmetrically, it can be applied to the samples). Such a projection is characterized by the fact that the covariance matrix is preserved at best by few coordinates, and hence it is possible to visualize the points in a small space (2-3 dimensions) loosing very little part of the information contained in the data.

6.5. Experiments classification

Classification is the action of assigning an unknown object into a predefined class after examining its characteristics [16]. The experiments classification aims to differentiate the classes present in the gene expression profiles data set. An experimental sample is usually associated with a particular state or class (e.g., control *versus* case). For classifying the samples, *supervised machine learning* techniques can be used. In supervised learning, the class label of the samples is assigned or known, a classification function

[3] Discretization is the conversion of numeric (continuous) values into discrete ones

or model is learned from a part of the data set (the training set) and applied for verification to the rest of the data set (the test set) for verifying the classification accuracy. A classification method is a systematic approach for building a classification model from the training data set [49], characterized by a learning algorithm that computes a model for representing the relationship between the class and the attributes set of the records. This model should fit to the input data and to new data objects belonging to the same classes of the input data. Moreover, the classification model should contain common patterns for each class and be interpretable by humans.

A general approach for solving a classification problem consists of the following steps [49]: split the data in training set and test set; perform training; compute the classification model; verify the performances of the model on the training and on the test set; and evaluate.

The most popular classification methods that can be applied to gene expression profiles are: *C4.5 Classification Tree* (C4.5) [37, 36], *Support Vector Machines* (SVM) [52, 10], *Random Forest* (RF) [14], *Nearest Neighbour* [13], and *Logic Data Mining* (logic classification formulas or rule-based classifiers) [17].

In a *classification tree*, each node is associated to a predicate that represents the attributes of the objects in the data set; the values of this predicates are then used to iteratively generate new nodes ("growing the tree"). The special class attributes are represented in the tree by the leaves. The most used tree decision classifiers, such as C4.5 [36], rely on rules that create new nodes with the local objective of minimizing the class entropy. The classification trees models can be easily transformed in "if-then rules", which are easily interpretable by humans.

Similar results can be obtained by a class of methods commonly referred to as *Logic Data Mining*, or rule-based classification, where the classifier uses logic propositional formulas in disjunctive or conjunctive normal form ("if-then rules") for classifying the given records. Examples of methods for computing logic classification formulas are RIPPER [7], LSQUARE [17], GELA [54],[55], LAD [5], RIDOR [19], and PART [18]. The major strength of classification formulas is the expressiveness of the models, that are very easy to interpret, e.g., "if gene Nudt19 $<$ 0.76 then the sample is diseased".

As a success of experiment classification, the work in [2] is cited where the authors are able to distinguish the Alzheimer's diseased *versus* the control microarray experimental samples using only few genes and individuating logic formulas in the form of "if-then" classification rules.

SVM map the data in n dimensional vectors, and try to construct a separating hyperplane which maximizes the margin (defined as the minimum distance between the hyperplane and the closest point in each class) between the data. Their strength relies in the ability to impose very complex non linear transformation on the data, lifting the data in a new space where linear separation is easier to achieve. They normally obtain very good classification performances on numeric data, but the main drawback is that the classification model is not easily interpretable.

RF is a classification method that operates by constructing several classification trees, by selecting randomly the features (i.e., genes), and by using a special model evaluation named *bagging*.

The *K-Nearest Neighbour* (KNN) algorithm is a classifier based on the closest training data in the feature space [13]. Given a training set of objects whose class is known, a new input object is classified by looking at its k closest neighbours of the training set (e.g., using the Euclidean distance). The main drawback of this approach is that no model is computed to classify the data.

Furthermore, when evaluating a classifier, the *cross validation* data sampling technique for supervised learning is recommended. Cross validation is a standard sampling technique that splits the dataset in a random way in k disjoint sets, the classification procedure is run k times with different sets. At a generic run k, the k subset is used as test set and the remaining $k-1$ sets are merged and used as training set for building the model. Everyone of the k sets contains a random distribution of the data. The cross validation sampling procedure builds k models and each of these models is validated with a different set of data. Classification statistics are computed for every model and the average of those represents an accurate estimation of the data mining procedure performance.

6.6. Software tools for gene expression profiles analysis

In this subsection, we present four state of the art software packages that can be used to perform the previously described gene expression profile analysis steps.

6.6.1. TM4 software suite

TM4 software suite [41] is a collection of different open source applications that can be used in the major steps of the gene expression profile analysis pipeline. TM4 comprises four analysis tools, *MicroArray DAta Manager* (MADAM), *Spotfinder*, MIDAS, and *Multi-Experiment Viewer* (MEV).
MADAM is the software tool dedicated to the data management of gene expression profiles. It is able to guide the user in creating a relational database to store the expression values and to keep track of the experimental and analysis work-flow.
Spotfinder is the tool designed for the analysis of microarray images and the quantification of gene expressions. It processes the image files that are produced by the microarray scanners and computes the gene expression profile intensities.
After that, the intensity values can be processed and analyzed, but a normalization step is necessary, as explained in section 6.2. MIDAS is the tool that is dedicated to this task, integrating several normalization methods.
MEV is a specific freely available software package able to perform the real gene expression analysis: it provides a collection of clustering, classification, statistical, and visualization methods. MEV can process even normalized data of RNA-Seq [24].
The main advantages of TM4 software suite are its *ad-hoc* analysis tools with friendly user interfaces and with several integrated analysis methods.

6.6.2. GenePattern

GenePattern is a user-friendly platform for computational analysis and visualization of genomic data freely available at the Broad Institute website (broadinstitute.org/cancer/software/genepattern/) [39]. Its comprehensive environment offers the possibility to perform the entire pipeline of gene expression data analysis from raw data pre-processing, gene selection, and gene clustering to pathway detection [30]. By means of *GenePattern*, the whole pipeline of data analysis can be recorded and re-run also with different data or parameters. Gene expression data can be given as input in *Affymetrix GeneChip* or tab-delimited text formats. Gene expression data from any source can be easily transformed into a tab separated file. *GenePattern* has an extensible structure that allows to share analysis modules and quickly integrate new methods with existing tools, supporting the steady growing repository of new computational methods. It is also possible to access to *GenePattern* from the *R* (www.r-project.org) [38], MATLAB (The MathWorks Inc., Natick, MA), or Java programming languages (java.com).

6.6.3. WEKA

Waikato Environment for Knowledge Analysis (WEKA) [21] is a Java software tool that comprises a collection of open source data analysis methods, including classification and clustering algorithms. Two main software containers -"weka.classifier" and "weka.cluster" - provide implementations of established classification and clustering algorithms, as the ones described in section 6.5. WEKA strengths are the friendly user interface, the amount of available algorithms, and the possibility of performing several experiments and comparisons in an integrated software suite. As a general tool for machine learning, WEKA uses an own file format (arff), so a format conversion of the gene expression profiles data must be performed.

6.6.4. GELA

GELA [54], [55] is a clustering and rule-based classification software, particularly engineered for gene expression analysis. The aims of GELA are to cluster gene expression profiles in order to discover similar genes and to classify the experimental samples. GELA converts the numeric gene expression profiles into discrete by defining intervals, implements a method named *Discrete Cluster Analysis* (DCA) based on the discretization step to cluster the genes, uses an integer programming method for selecting the characteristic genes for each class, adopts the lsquare method for computing the logic classification formulas ("if-then rules"), and a special weighted sample classification procedure. GELA also integrates standard statistical methods for gene expression profiles data analysis: PCA to group similar genes and experiments, and Pearson Correlation Analysis to find a list of correlated genes to a selected gene. It is available at dmb.iasi.cnr.it/gela.php. GELA also supports the classification and clustering of RNA-Seq data.

7. Real case studies

To provide the reader with an example of the analysis process described in the previous sections, we describe in the following some results obtained in our research. In [2], GELA was applied in a microarray case-control study: early (1–3 months) and late stage (6–15 months) experimental samples of *Alzheimer's disease versus* healthy mice had to be distinguished by analyzing their gene expression profiles. 119 experimental samples and 16,515 gene expression profiles, provided by the *European Brain Research Institute* (EBRI), have been analyzed in order distinguish the experimental samples with a clear human interpretable classification model and to detect the genes whose expression or co-expression strongly characterizes the Alzheimer's disease. Firstly, a MAS 5.0 normalization step was performed using the standard Affymetrix Expression Console software (ver 1.2). Then, a clustering of the genes with the DCA individuated each other related gene groups and shrank the whole data set to 3656 genes for 1–3 months and to 3615 for 6–15 months. Finally, the classification model was computed: a small number of classification formulas that are able to distinguish diseased from control mice were extracted from GELA and a small number of genes capable to effectively separate control and diseased mice samples was identified. The logic separating formulas for 1–3 and 6–15 months are reported in Table 4 and Table 5, respectively. Each clause of such tables is able alone to separate the two different classes. A 30-fold cross validation and a percentage split sampling technique (10% test and 90% training) were used to validate the logic formulas, obtaining 99% of accuracy (percentage of the correctly classified samples). The same classification analysis was performed with the WEKA [21] software for comparing the results obtained with GELA. A subset of the WEKA classification algorithms, C4.5 Decision Tree algorithm, RF, SVM, and KNN, were used on the Alzheimer's data set. Different parameter settings of the algorithms were used, the best resulting accuracy are reported in Table 6. All the algorithms were run by using a 30-fold cross validation sampling procedure. From the results, it can be seen that all methods (except KNN) perform at a comparable level; GELA and SVM have excellent accuracies, but SVM produces classification models whose interpretation is very difficult for human beings. On the other hand, GELA and C4.5 are able to extract meaningful and compact models (i.e., classification formulas and trees, re-

Table 4: Logic formulas in early stage

Alzheimer's Disease	(Nudt19 $<$ 0.76) OR
	(Arl16 $\geq$ 1.31) OR
	(Aph1b $\geq$ 0.47) OR
	(Slc15a2 $\geq$ 0.55) OR
	(Agpat5 $\geq$ 0.73) OR
	(Sox2ot $<$ 0.58 OR Sox2ot $\geq$ 1.53) OR
	(2210015D19Rik $\geq$ 0.86) OR
	(Wdfy1 $\geq$ 1.37)
Control	(Nudt19 $\geq$ 0.76) OR
	(Arl16 $<$ 1.31) OR
	(Aph1b $<$ 0.47) OR
	(Slc15a2 $<$ 0.55) OR
	(Agpat5 $<$ 0.73) OR
	(0.58 $\geq$ Sox2ot AND Sox2ot $<$ 1.53) OR
	(2210015D19Rik $<$ 0.86) OR
	(Wdfy1 $<$ 1.37)

Table 5: Logic formulas in late stage

Alzheimer's Disease	(Slc15a2 $\geq$ 0.62) OR (Agpat5 $<$ 0.26 OR Agpat5 $\geq$ 0.55) OR (Sox2ot $\geq$ 1.78) OR (2210015D19Rik $\geq$ 0.82) OR (Wdfy1 $<$ 0.75 OR Wdfy1 $\geq$ 1.29) OR (D14Ertd449e $<$ 0.33 OR D14Ertd449e $\geq$ 0.52) OR (Tia1 $<$ 0.17 OR Tia1 $\geq$ 0.49) OR (Txnl4 $<$ 0.74) OR (1810014B01Rik $<$ 0.71 OR 1810014B01Rik $\geq$ 1.17) OR (Snhg3 $<$ 0.16 OR Snhg3 $\geq$ 0.35) OR [(1.12 $\geq$ Actl6a AND Actl6a $<$ 1.42) OR Actl6a $\geq$ 1.48] OR (Rnf25 $<$ 0.67 OR Rnf25 $\geq$ 1.26)
Control	(Slc15a2 $<$ 0.62) OR (0.26 $\geq$ Agpat5 AND Agpat5 $<$ 0.55) OR (Sox2ot $<$ 1.78) OR (2210015D19Rik $<$ 0.82) OR (0.75 $\geq$ Wdfy1 AND Wdfy1 $<$ 1.29) OR (0.33 $\geq$ D14Ertd449e AND D14Ertd449e $<$ 0.52) OR (0.17 $\geq$ Tia1 AND Tia1 $<$ 0.49) OR (Txnl4 $\geq$ 0.74) OR (0.71 $\geq$ 1810014B01Rik AND 1810014B01Rik $<$ 1.17) OR (0.16 $\geq$ Snhg3 AND Snhg3 $<$ 0.35) OR [(0.81 $<$ Actl6a AND Actl6a $<$ 1.12) OR (1.42 $<$ Actl6a AND Actl6a $<$ 1.48)] OR (0.67 $\geq$ Rnf25 AND Rnf25 $<$ 1.26)

Table 6: Classification accuracy [%] on Alzheimer data sets

method	setting	early stage	late stage	model
GELA	no settings	100.0	100.0	yes
SVM	*polykernel*=2	96.66	100.0	no
RF	*trees*=100	96.66	94.91	no
C4.5	*unpruned*, *minobj*=2	98.33	98.30	yes
KNN	*k*=2	70.00	86.44	no

spectively).

Other tests have been performed on data sets downloaded from public repositories *ArrayExpress* and *Gene Expression Omnibus* (GEO): *Psoriasis* and *Multiple Sclerosis Diagnostic*. The Psoriasis data set was composed of 54,613 gene expression profiles of 176 experimental samples (85 control and 91 diseased) and was provided from the National Psoriasis Foundation. The Multiple Sclerosis Diagnostic data set contained 22,215 gene expression profiles of 178 experimental samples (44 control and 134 diseased) and was released from the *National Institute of Neurological Disorders and Stroke* (NINDS). All gene expression profile values were normalized using the standard Affymetrix Expression Console software (ver 1.2) by the MAS5 algorithm. The results are reported in Table 7. In this case, all methods perform at a comparable level. SVM, KNN, and RF produce classification models that are difficult to understand for humans. On the other hand, GELA and C4.5 obtained clear classification models.

Table 7: Classification accuracy [%] on multiple sclerosis and psoriasis data sets

method	**setting**	**MsDiagnostic**	**Psoriasis**	**model**
GELA	no settings	94.94	100.0	yes
SVM	*polykernel*=2	90.45	98.86	no
RF	*trees*=100	91.57	98.86	no
C4.5	*unpruned, minobj*=2	87.08	97.16	yes
KNN	*k*=2	87.64	99.43	no

Finally, we describe the application of GELA for a case-control analysis on a public RNA-Seq data set of breast cancer extracted from *The Cancer Genome Atlas* (TCGA) data portal (http://cancergenome.nih.gov/). GELA has been applied to the *breast cancer* data set where 20531 gene expression profiles of 783 subjects have been analyzed for classifying experimental samples either in control (i.e., healthy subjects) or breast cancer patients. It was able to detect several clusters of similar genes and to provide a clear, compact, and accurate classification model, composed of small "if-then rules", which also allow detecting a small subset of those genes that characterize the breast cancer. The classification model was tested with a 10-fold cross validation sampling and performed with a 98% accuracy. For comparing the results with respect to other supervised machine learning algorithms, a similar analysis was performed with WEKA [21], using C4.5, RF, SVM, and KNN methods. The accuracy values of a 10-fold cross validation scheme are depicted in Table 8. All methods performed with excellent classification rates showing an accuracy that is even greater than 98%. RF, SVM, and KNN are the best performing methods, but GELA and C4.5 have the advantage to provide a human-readable classification model, such as in Figure 4.

Table 8: Classification accuracy [%] on breast cancer data set

Method	**Settings**	**Accuracy**	**Model**
GELA	no settings	98	yes
C4.5	*unpruned, minobj*=2	98	yes
RF	*trees*=100	99	no
SVM	*polykernel*=2	99	no
KNN	*k*=2	99	no

ABCA5|23461<5.99 OR ADRB2|154<2.54 ⇒ breast cancer

Figure 4: An example of GELA model for breast cancer

8. Conclusions

Thanks to the new advances in microarray and RNA-Seq technology, a large quantity of gene expression profiles data is available both at on-line open data repositories and at local laboratories. In this work efficient methods, algorithms, and software for performing an effective gene expression profiles analysis have been described. In particular, the emphasis was placed on two types of analysis: *gene clustering* and *experiment classifications* for which several methods have been presented and described. Afterwards, gene expression profile analysis tools, which integrate these methods, have been illustrated. For providing the reader with a practical example, the software GELA and WEKA were applied to four real case studies. The performed experiments show a complete knowledge discovery process whose aim is to identify the hidden patterns in the different classes of the experimental samples by analyzing their gene expression profiles.

20.

References

[1] Affymetrix, *Affymetrix Microarray Suite User Guide.* Affymetrix, Santa Clara, CA, version 5 edition ed., 2001.

[2] I. Arisi, M. D'Onofrio, R. Brandi, A. Felsani, S. Capsoni, G. Drovandi, G. Felici, E. Weitschek, P. Bertolazzi, and A. Cattaneo, "Gene expression biomarkers in the brain of a mouse model for alzheimer's disease: mining of microarray data by logic classification and feature selection," *Journal of Alzheimer's Disease*, vol. 24, no. 4, pp. 721–738, 2011.

[3] W. Ayadi, M. Elloumi, and J.-K. Hao, "A biclustering algorithm based on a bicluster enumeration tree: application to dna microarray data," *BioData mining*, vol. 2, no. 1, p. 9, 2009.

[4] B. M. Bolstad, R. A. Irizarry, M. Åstrand, and T. P. Speed, "A comparison of normalization methods for high density oligonucleotide array data based on variance and bias," *Bioinformatics*, vol. 19, no. 2, pp. 185–193, 2003.

[5] E. Boros, P. L. Hammer, T. Ibaraki, A. Kogan, E. Mayoraz, and I. Muchnik, "An implementation of logical analysis of data," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 12, no. 2, pp. 292–306, 2000.

[6] M. Chee, R. Yang, E. Hubbell, A. Berno, X. C. Huang, D. Stern, J. Winkler, D. J. Lockhart, M. S. Morris, and S. P. Fodor, "Accessing genetic information with high-density dna arrays," *Science*, vol. 274, no. 5287, pp. 610–614, 1996.

[7] W. W. Cohen, "Fast effective rule induction," in *In Proceedings of the Twelfth International Conference on Machine Learning*, pp. 115–123, Morgan Kaufmann, 1995.

[8] T. F. Cox and G. Ferry, "Discriminant analysis using non-metric multidimensional scaling," *Pattern Recognition*, vol. 26, no. 1, pp. 145–153, 1993.

[9] F. H. Crick and J. D. Watson, "The complementary structure of deoxyribonucleic acid," *Proceedings of the Royal Society of London. Series A. Mathematical and Physical Sciences*, vol. 223, no. 1152, pp. 80–96, 1954.

[10] N. Cristianini and J. Shawe-Taylor, *An Introduction to Support Vector Machines and other kernel-based learning methods.* Cambridge University Press, 2000.

[11] O. Cuisenaire and B. Macq, "Fast euclidean distance transformation by propagation using multiple neighborhoods," *Computer Vision and Image Understanding*, vol. 76, no. 2, pp. 163–172, 1999.

[12] P.-E. Danielsson, "Euclidean distance mapping," *Computer Graphics and image processing*, vol. 14, no. 3, pp. 227–248, 1980.

[13] B. V. Dasarathy, *Nearest neighbor (NN) norms: NN pattern classification techniques.* 1990.

[14] R. Díaz-Uriarte and S. A. De Andres, "Gene selection and classification of microarray data using random forest," *BMC bioinformatics*, vol. 7, no. 1, p. 3, 2006.

[15] J. H. Do, D. Choi, *et al.*, "Normalization of microarray data: single-labeled and dual-labeled arrays," *Molecules and cells*, vol. 22, no. 3, p. 254, 2006.

[16] S. Dulli, S. Furini, and P. E., *Data Mining.* Springer, 2009.

[17] G. Felici and K. Truemper, "A minsat approach for learning in logic domains," *INFORMS Journal on Computing*, vol. 13, no. 3, pp. 1–17, 2002.

[18] E. Frank and I. H. Witten, "Generating accurate rule sets without global optimization," 1998.

[19] B. R. Gaines and P. Compton, “Induction of ripple-down rules applied to modeling large databases,” *Journal of Intelligent Information Systems*, 1995.

[20] R. C. Gentleman, V. J. Carey, D. M. Bates, B. Bolstad, M. Dettling, S. Dudoit, B. Ellis, L. Gautier, Y. Ge, J. Gentry, *et al.*, “Bioconductor: open software development for computational biology and bioinformatics,” *Genome biology*, vol. 5, no. 10, p. R80, 2004.

[21] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The weka data mining software: an update,” *SIGKDD Explor. Newsl.*, vol. 11, pp. 10–18, Nov. 2009.

[22] M. J. Heller, “Dna microarray technology: devices, systems, and applications,” *Annual review of biomedical engineering*, vol. 4, no. 1, pp. 129–153, 2002.

[23] A. J. Holloway, R. K. Van Laar, R. W. Tothill, and D. D. Bowtell, “Options availablefrom start to finishfor obtaining data from dna microarrays ii,” *Nature genetics*, vol. 32, pp. 481–489, 2002.

[24] E. A. Howe, R. Sinha, D. Schlauch, and J. Quackenbush, “Rna-seq analysis in mev,” *Bioinformatics*, vol. 27, no. 22, pp. 3209–3210, 2011.

[25] R. A. Irizarry, B. Hobbs, F. Collin, Y. D. Beazer-Barclay, K. J. Antonellis, U. Scherf, and T. P. Speed, “Exploration, normalization, and summaries of high density oligonucleotide array probe level data,” *Biostatistics*, vol. 4, no. 2, pp. 249–264, 2003.

[26] D. Jiang, C. Tang, and A. Zhang, “Cluster analysis for gene expression data: a survey,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 11, pp. 1370 – 1386, 2004.

[27] T. Jirapech-Umpai and S. Aitken, “Feature selection and classification for microarray data analysis: Evolutionary methods for identifying predictive genes,” *BMC Bioinformatics*, vol. 148, 2005.

[28] I. Jolliffe, *Principal component analysis*. Wiley Online Library, 2005.

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

[30] H. Kuehn, A. Liberzon, M. Reich, and J. P. Mesirov, “Using genepattern for gene expression analysis,” *Current Protocols in Bioinformatics*, pp. 7–12, 2008.

[31] W. K. Leow and R. Li, “The analysis and applications of adaptive-binning color histograms,” *Computer Vision and Image Understanding*, vol. 94, no. 1, pp. 67–91, 2004.

[32] C. Li and W. H. Wong, “Dna-chip analyzer (dchip),” *The analysis of gene expression data: methods and software*, pp. 120–141, 2003.

[33] J. MacQueen *et al.*, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, vol. 1, pp. 281–297, California, USA, 1967.

[34] J. C. Marioni, C. E. Mason, S. M. Mane, M. Stephens, and Y. Gilad, “Rna-seq: an assessment of technical reproducibility and comparison with gene expression arrays,” *Genome research*, vol. 18, no. 9, pp. 1509–1517, 2008.

[35] A. Mortazavi, B. A. Williams, K. McCue, L. Schaeffer, and B. Wold, “Mapping and quantifying mammalian transcriptomes by rna-seq,” *Nature methods*, vol. 5, no. 7, pp. 621–628, 2008.

[36] J. R. Quinlan, “Improved use of continuous attributes in C4.5,” *Journal of Artificial Intelligence Research*, vol. 4, pp. 77–90, 1996.

[37] J. R. Quinlan, *C4. 5: programs for machine learning*, vol. 1. Morgan kaufmann, 1993.

[38] R Core Team, *R: A Language and Environment for Statistical Computing*. 2013.

[39] M. Reich, T. Liefeld, J. Gould, J. Lerner, P. Tamayo, and J. P. Mesirov, “Genepattern 2.0,” *Nature genetics*, vol. 38, no. 5, pp. 500–501, 2006.

[40] L. B. Romdhane, H. Shili, and B. Ayeb, “Mining microarray gene expression data with unsupervised possibilistic clustering and proximity graphs,” *Applied intelligence*, vol. 33, no. 2, pp. 220–231, 2010.

[41] A. I. Saeed, N. K. Bhagabati, J. C. Braisted, W. Liang, V. Sharov, E. A. Howe, J. Li, M. Thiagarajan, J. A. White, and J. Quackenbush, “[9] tm4 microarray software suite,” *Methods in enzymology*, vol. 411, pp. 134–193, 2006.

[42] F. Sanger, S. Nicklen, and A. R. Coulson, “Dna sequencing with chain-terminating inhibitors,” *Proceedings of the National Academy of Sciences*, vol. 74, no. 12, pp. 5463–5467, 1977.

[43] E. E. Schadt, C. Li, B. Ellis, and W. H. Wong, “Feature extraction and normalization algorithms for high-density oligonucleotide gene expression array data,” *Journal of Cellular Biochemistry*, vol. 84, no. S37, pp. 120–125, 2001.

[44] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown, “Quantitative monitoring of gene expression patterns with a complementary dna microarray,” *Science*, vol. 270, no. 5235, pp. 467–470, 1995.

[45] M. Schena, D. Shalon, R. W. Davis, and P. O. Brown, “Quantitative monitoring of gene expression patterns with a complementary dna microarray,” *Science*, vol. 270, no. 5235, pp. 467–470, 1995.

[46] J. W. Song and K. C. Chung, “Observational studies: cohort and case-control studies,” *Plastic and reconstructive surgery*, vol. 126, no. 6, p. 2234, 2010.

[47] D. Stekel, *Microarray bioinformatics*. Cambridge University Press, 2003.

[48] R. B. Stoughton, “Applications of dna microarrays in biology,” *Annu. Rev. Biochem.*, vol. 74, pp. 53–82, 2005.

[49] P. Tan, M. Steinbach, and V. Kumar, *Introduction to Data Mining*. Addison Wesley, 2005.

[50] V. Torra and Y. Narukawa, “On a comparison between mahalanobis distance and choquet integral: The choquet–mahalanobis operator,” *Information Sciences*, vol. 190, pp. 56–63, 2012.

[51] C. Trapnell, B. A. Williams, G. Pertea, A. Mortazavi, G. Kwan, M. J. van Baren, S. L. Salzberg, B. J. Wold, and L. Pachter, “Transcript assembly and quantification by rna-seq reveals unannotated transcripts and isoform switching during cell differentiation,” *Nature biotechnology*, vol. 28, no. 5, pp. 511–515, 2010.

[52] V. N. Vapnik, *Statistical learning theory*. Wiley, 1998.

[53] Z. Wang, M. Gerstein, and M. Snyder, “Rna-seq: a revolutionary tool for transcriptomics,” *Nature Reviews Genetics*, vol. 10, no. 1, pp. 57–63, 2009.

[54] E. Weitschek, G. Felici, and P. Bertolazzi, “Mala: a microarray clustering and classification software,” in *Database and Expert Systems Applications (DEXA), 2012 23rd International Workshop on Biological Knowledge Discovery*, pp. 201–205, IEEE, 2012.

[55] E. Weitschek, G. Fiscon, G. Felici, and P. Bertolazzi, “Gela: Gene expression logic analyzer,” in *From structural bioinformatics to integrative systems biology, Nettab2014 Workshop*, pp. 85–87, 2014.

[56] R. Xu, D. Wunsch, *et al.*, “Survey of clustering algorithms,” *IEEE Transactions on Neural Networks*, vol. 16, no. 3, pp. 645–678, 2005.