PART H

ASSOCIATION RULES LEARNING FROM BIOLOGICAL DATA

CHAPTER 34

MINING FREQUENT PATTERNS AND ASSOCIATION RULES FROM BIOLOGICAL DATA

IOANNIS KAVAKIOTIS, GEORGE TZANIS, and IOANNIS VLAHAVAS
Department of Informatics, Aristotle University of Thessaloniki, Thessaloniki, Greece

34.1 INTRODUCTION

During the last years biology and computer science have been characterized by major advances that have attracted a lot of interest. Nowadays the collaboration between biologists and computer scientists is deemed a vital necessity for the further progress of biological research. Bioinformatics is a novel research area that has emerged as a solution to the aforementioned need for collaboration. Two relative subfields of computer science, data mining and machine learning, have provided biologists, as well as experts from other areas, a powerful set of tools to analyze new data types in order to extract various types of knowledge efficiently and effectively. These tools combine powerful techniques of artificial intelligence, statistics, mathematics, and database technology. This fusion of technologies aims to overcome the obstacles and constraints posed by the traditional statistical methods.

Association rule (AR) mining has attracted the attention of the data-mining research community since the early 1990s as a means of unsupervised, exploratory data analysis. ARs were first introduced by Agrawal et al. [1] as a market basket analysis tool. However, since then, they have been effectively applied to many other application domains, including biology and bioinformatics. An AR implies the coexistence of a number of itemsin a portion of a transaction database. The goal of this exploratory data analysis is to provide the decision maker with valuable knowledge about a certain domain modeled by a transaction database. The frequent existence of two or more items in the same transaction implies a relationship among them. For example, the existence of bread and butter in the same baskets implies a possible buying behavior pattern that can be further investigated in order to improve the sales of both products. Similarly, the coexistence of high-expression values of a number of genes in the same transactions–experiments indicates a possible coexpression pattern of these genes. Conversely, the rare or the absolutely non-coexistence of two products could also imply a negative association (e.g., a mutual exclusion) among them.

Biological Knowledge Discovery Handbook: Preprocessing, Mining, and Postprocessing of Biological Data, First Edition. Edited by Mourad Elloumi and Albert Y. Zomaya.

Many algorithms for mining ARs and others that extend the concept of AR mining have been proposed so far. Agrawal and Srikant [2] proposed APRIORI, the first algorithm for mining ARs. APRIORI is a levelwise algorithm which works by generating candidate sets of items and testing if they are frequent by scanning the database. It is one of the most popular data-mining algorithms and will be described in more detail later in the chapter. About the same time Mannila et al. [3] independently discovered a variation of APRIORI, the OCD algorithm. The large number of algorithms that have been proposed since then either improve efficiency, such as FP-GROWTH [4] and ECLAT [5], or address different problems from various application domains, such as spatial [6], temporal [7], and intertransactional ARs [8].

One of the major problems in AR mining is the large number of often uninteresting rules extracted. Some approaches that are based on concept hierarchies try to deal with this problem. Srikant and Agrawal [9] presented the problem of mining for generalized ARs. These rules utilize item taxonomies (concept hierarchies) in order to discover more interesting rules. Thomas and Sarawagi [10] proposed a technique for mining generalized ARs based on Structured Query Language (SQL) queries. Han and Fu [11] also describe the problem of mining "multiple-level" ARs based on taxonomies and propose a set of top-down progressive deepening algorithms.

Another kind of approach deals with negative associations between items. Savasere et al. [12] introduced this kind of problem. Negative associations relate to the problem of finding rules that imply what items are not likely to appear in a transaction when a certain set of items appears in the transaction. The approach of Savasere et al. demands the existence of a taxonomy and is based on the assumption that items belonging to the same parent of taxonomy are expected to have similar types of associations with other items. In another work Wu et al. [13] presented an efficient method for mining positive and negative associations and proposed a pruning strategy and an interestingness measure. Finally, another kind of negative association that has been studied concerns the mining of mutually exclusive items [14, 15].

The process of *knowledge discovery from databases* (KDD) consists of a number of steps that can be grouped in three categories: preprocessing, data mining, and postprocessing (Figure 34.1). Although the core of the process is the data-mining step, where a data-mining algorithm (e.g., APRIORI for mining of ARs) is applied, the preprocessing and postprocessing phases are particularly important and contribute sensibly to the extraction of valuable knowledge. The preprocessing phase usually includes the selection of an

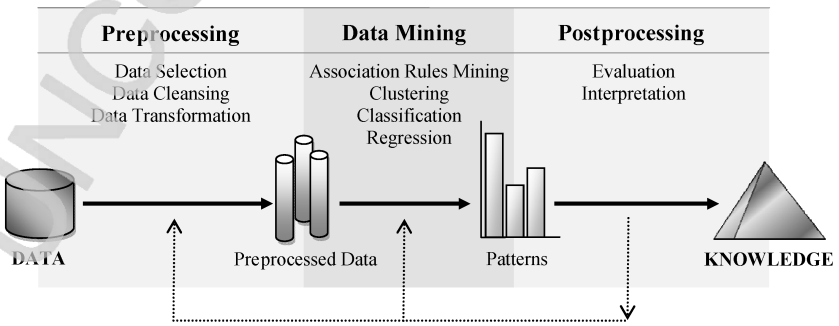

FIGURE 34.1 The KDD process.

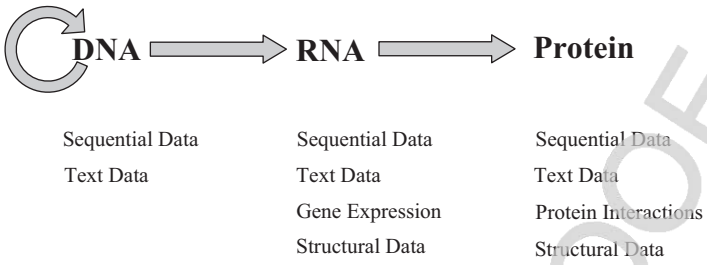

FIGURE 34.2 Central dogma of molecular biology and main kinds of biological data.

appropriate portion of the data, the cleaning of the selected data, and the transformation of the data. The postprocessing phase deals with the management of the produced patterns and focuses on the evaluation and interpretation of data-mining results.

The central dogma of molecular biology, as coined by Francis Crick [16], describes the flow of biological information (Figure 34.2). In most organisms DNA is transcribed into RNA and then RNA is translated into protein. The circular arrow around DNA denotes its ability to replicate itself. The figure also describes the basic kinds of data that can be produced by various biological experiments in relation to the three basic molecules of life.

It is important to mention that AR mining is a very prominent tool in application domains that include a large number of binary attributes and the associations among these attributes could be meaningful. A convenient application domain in biology is gene expression data, which include a large number of attributes (genes), and the associations among different genes are often particularly important. Although the expression values of genes are not binary, an appropriate discretization algorithm can convert these values to a suitable format for applying an AR mining algorithm successfully. Even though gene expression data are quite convenient for association analysis, AR mining is also applied to other kinds of biological data. These applications will be presented in the following sections.

In the next section, the problem of mining ARs will be defined. Then, some important mining algorithms, preprocessing and postprocessing methods, will be described. In the rest sections, the application of AR mining algorithms to various kinds of biological data will be presented. Finally, the chapter will be concluded and summarized.

34.2 DEFINITION OF AR MINING PROBLEM

The original definition of the AR mining problem has been given by Agrawal et al. in 1993 [1]. Let $I = \{i_1, i_2, \ldots, i_N\}$ be a finite set of binary attributes called *items* and $D = \{t_1, t_2, \ldots, t_N\}$ be a finite multiset of transactions called the *database*. Each *transaction* t_i contains a subset of items chosen from I and has a unique transaction ID. A set of items is referred to as an *itemset*. If an itemset contains k items, it is called a k-itemset. The number k is called the *size* or *length* of the itemset. The itemset that does not contain any items is called an empty itemset. A transaction $T \in D$ is said to contain an itemset $X \subseteq I$, if $X \subseteq T$.

An AR is an implication of the form $X \Rightarrow Y$ where $X \subset I$, $Y \subset I$, and $X \cap Y = \emptyset$. The itemset X is called the *antecedent* or *left-hand-side* (LHS) of the rule and the itemset Y is called the *consequent* or *right-hand-side* (RHS) of the rule.

TABLE 34.1 Example of Binary Gene Expression Matrix

Transaction ID	Gene 1	Gene 2	Gene 3	Gene 4
T1	0	1	0	0
T2	0	0	0	1
T3	1	1	0	0
T4	0	0	1	0
T5	1	1	1	0

There are many measures that have been proposed in order to evaluate a rule's interestingness. The most popular are *support* and *confidence*. They respectively reflect the usefulness and certainty of discovered rules. More specifically, support determines how often a rule is applicable to a given data set, whereas confidence determines how frequently items in Y appear in transactions that contain X. The support of a rule $X \Rightarrow Y$ is equal to the support of the itemset $X \cup Y$ and is defined as the fraction of transactions in the database which contain the itemset. The support of an itemset X is calculated as presented in the following equation:

$$\text{support}_D(X) = \frac{|\{T \in D | X \subseteq T\}|}{|D|}$$

The confidence of the rule $X \Rightarrow Y$ is defined as the fraction of transactions in the database that contain $X \cup Y$ over the number of transactions that contain only X. In other words, confidence is equal to the fraction of the support of $X \cup Y$ in D over the support of X in D. The equation that defines confidence is presented below:

$$\text{confidence}_D(X \Rightarrow Y) = \frac{\text{supp}_D(X \cup Y)}{\text{supp}_D(X)}$$

In order to make the concepts presented above more clear, a detailed example is presented. Table 34.1 presents a binary data matrix (database) concerning the expressions of four genes. Every transaction represents a separate measurement of gene expression levels. All measurements have been discretized so that only two levels of gene expression are possible. A value of zero represents a gene that is underexpressed, whereas a value of unity represents a gene that is overexpressed.

In this example the set of items is $I = \{\text{gene 1, gene 2, gene 3, gene 4}\}$. In the database there are five transactions, for example, $t_3 = \{\text{gene 1, gene 2}\}$. An example of a rule for this database could be $\{\text{gene 1, gene 2}\} \Rightarrow \{\text{gene 3}\}$. The support for this rule is $\frac{1}{5}$ (20%), because only one transaction (t_5) out of five contains gene 1, gene 2 and gene 3. The confidence of the rule is $\frac{1}{2}$ (50%), because the support of itemset $\{\text{gene 1, gene 2}\}$ (the antecedent of the rule) is $\frac{2}{5}$ and the support of itemset $\{\text{gene 1, gene 2, gene 3}\}$ is $\frac{1}{5}$.

The AR mining problem can be decomposed in two major subtasks:

1. *Frequent-Itemset Generation* Its purpose is to find all itemsets that satisfy a user-specified minimum-support threshold (min_sup). These itemsets are called *frequent itemsets*.
2. *Rule Generation* Its purpose is to extract from the frequent itemsets all the rules that satisfy a user-specified minimum-confidence threshold (min_conf).

The first subtask is the more computationally complex and has concentrated all the focus of the research community. The second subtask is a straightforward task and does not attract the interest of researchers.

34.3 ALGORITHMS FOR MINING ARs

In this section three popular algorithms that are based on three main methodologies for mining all frequent itemsets and consequently ARs will be presented.

34.3.1 APRIORI

Most of the algorithms for mining frequent itemsets are based on the principle of downward closure. This principle imposes that all nonempty subsets of a frequent itemset are also frequent. For example, if itemset {gene 1, gene 2, gene 5} is frequent according to a minimum-support threshold, then itemsets {gene 1, gene 2}, {gene 1, gene 5}, {gene 2, gene 5}, {gene 1}, {gene 2}, and {gene 5} must also be frequent according to the same minimum-support threshold. The most popular algorithm that employs this principle is APRIORI [2].

APRIORI works by constructing candidate frequent itemsets and then checks which of them are indeed frequent (Table 34.2). For the generation of candidate k-frequent itemsets, the set of $(k-1)$-frequent itemsets is exploited according to the principle of downward closure. Thus, the process of frequent-itemset mining in APRIORI is a two-step process:

1. The set of candidate frequent itemsets C_i is constructed.
2. Then the set of frequent itemsets L_i is constructed by scanning the database and checking which candidates in C_i using the minimum-support threshold.

To clarify the process, the algorithm initially constructs all the candidate frequent 1-itemsets (C_1), which actually include all the items. Then it is checked by scanning the database whether the minimum-support threshold is satisfied for these candidates and so the set of frequent 1-itemsets (L_1) is generated. Next, L_1 is used in order to generate the set of

TABLE 34.2 Pseudocode of APRIORI Algorithm

Input: Database D, minimum support threshold (min_sup)
Output: All the frequent itemsets

```
L1 ← {frequent items}
for (k ← 2; Lk−1 !=∅; k++) do
    Ck ← {candidates from Lk−1} (Lk−1× Lk−1 and downward closure)
    for each t ∈ D do
        for each c ∈ Ck do
            if (c ⊆ t)
                c.count++
    Lk ← {c ∈ Ck | c.count = min_sup}
return ⋃k Lk
```

candidate frequent 2-itemsets (C_2), exploiting the downward closure principle. Following the same process the 2-frequent dataset (L_2) is generated. The process ends in a specific number of iterations or when there are no more candidate frequent itemsets. The number of APRIORI's database scans is equal to the length of the longest candidate frequent.

34.3.2 FP-GROWTH

The main drawbacks of the APRIORI algorithm is that it not only generates a huge number of candidate itemsets but also scans the database several times. Both drawbacks are very costly. Han et al. [4] proposed an algorithm, called FP-GROWTH, that is not based on the generation of candidate frequent itemsets. In general, the algorithm uses a tree structure, the *frequent-pattern tree* (FP-Tree), which stores all the database. This structure can compress the data up to 200 times, and it is stored to the computer's memory, which leads to a more effective rule extraction. Moreover, this algorithm uses a divide-and-conquer approach in order to decompose the rule extraction process in simpler parts.

More specifically, in the first step, the algorithm compresses the database into an FP-Tree structure that is highly condensed but is complete for the purposes of frequent pattern mining. The beneficial consequent is that it avoids costly database scans during candidate generation. Then, in the second step, it extracts frequent itemsets directly from the FP-Tree using the divide-and-conquer methodology. Generally FP-GROWTH is faster than APRIORI.

34.3.3 ECLAT

Zaki [5] proposed a new algorithm for mining frequent patterns, called *equivalence class transformation* (ECLAT). The main difference between the two algorithms presented previously and ECLAT is that the first two mine frequent itemsets from a set of transactions in horizontal data format while ECLAT mines frequent itemsets in a vertical data format.

First, the algorithm builds the TID_set of all items in the transaction database. As in APRIORI, in ECLAT the frequent k-itemsets are generated from the frequent $(k-1)$-itemsets. In order to clarify the process, an example is presented below. In this example the minimum-support threshold is set to 20%. Table 34.3 presents the transaction database in which the first column represents the transaction ID (TID) and the second the items included in each transaction.

TABLE 34.3 Example Transaction Database

TID	Item IDs
T1	I4, I5
T2	I1, I3, I4, I5
T3	I1, I3, I5
T4	I3, I4
T5	I3, I4
T6	I4, I5
T7	I2, I3, I5
T8	I2, I5
T9	I3, I4, I5

TABLE 34.4 Transactional Database of Table 34.3 in Vertical Data Format

Itemset	TID_set
{I1}	T2, T3
{I2}	T7, T8
{I3}	T2, T3, T4, T5, T7, T9
{I4}	T1, T2, T4, T5, T6, T9
{I5}	T1, T2, T3, T6, T7, T8, T9

TABLE 34.5 Frequent 2-Itemsets (highlighted)

Itemset	TID_set	Support (%)
{I1, I3}	T2, T3	22
{I1, I4}	T2	11
{I1, I5}	T2, T3	22
{I2, I3}	T7	11
{I2, I5}	T7, T8	22
{I3, I4}	T2, T4, T5, T9	44
{I3, I5}	T2, T3, T7, T9	44
{I4, I5}	T1, T2, T6, T9	44

TABLE 34.6 Frequent 3-Itemsets

Itemset	TID_set	Support (%)
{I1, I3, I5}	T2, T3	22
{I3, I4, I5}	T2, T9	22

Table 34.4 presents the transactional database of Table 34.3 in vertical-data format. The first column represents the itemset and the second column the transactions which contain the particular itemset.

By intersecting the TID_set, the frequent 2-itemsets (highlighted) in vertical-data format are generated. These itemsets are presented in Table 34.5.

Again, by intersecting the TID_sets, the frequent 3-itemsets (Table 34.6) in vertical-data format are generated. An optimization based on the principle of downward closure is that there is no need to intersect {I1, I5} and {I4, I5} because {I1, I4} is not frequent and as a concequence {I1, I4, I5} cannot be frequent.

The process ends when there are no more candidate frequent itemsets or frequent itemsets.

34.4 PREPROCESSING AND POSTPROCESSING

As previously discussed, data preprocessing is an essential step before applying a data-mining algorithm. The preprocessing phase usually includes the selection of an appropriate portion of the data and the cleaning and transformation of the data. In AR mining problems the most necessary and most frequently used preprocessing procedure is discretization, which involves the transformation of the continuous range of an attribute's values to discrete intervals.

34.4.1 Discretization

Many algorithms in data mining and the vast majority of AR discovery algorithms and their derivations work only with categorical attributes. So it is essential to use data discretization techniques in order to reduce the number of values for a given continuous attribute by dividing the range of the attribute into intervals. In other words the aim of discretization techniques is to convert the numeric attributes into nominal ones. There are three axes by which discretization methods can be classified: *global/local, supervised/unsupervised*, and *static/dynamic* [17].

Local discretization algorithms produce partitions that are applied to localized regions of the instance space. Instead, global algorithms group values of each feature into intervals by considering other features. Supervised discretization algorithms are applicable only when the data are divided into classes. On the other hand, unsupervised algorithms do not consider the class value. Lastly, static algorithms discretize each feature in one iteration independent of the other features, whereas dynamic algorithms search for all possible intervals for all features simultaneously. In the rest of this section some of the most popular discretization strategies will be presented according to the second categorization (supervised/usupervised).

Supervised discretization is used when dealing with classification problems or a dataset in which a label attribute is assigned to each instance. This label indicates the class to which the instance belongs. This information is used to guide the discretization process. The most well known approach for supervised discretization is the one proposed by Fayyad and Irani [18]. This method uses the information gain in order to recursively define the best bins. The entropy of each bin should be minimal. The method works by recursively splitting the intervals until a stopping criterion is reached.

Let S be a training set. All instances in the training set should belong to one of c different classes. Each class is denoted by a number from 1 to c. Considering the above, entropy E of the set S is defined by the equation

$$E(S) = -\sum_{i=1}^{c} p_i \log_2(p_i)$$

In the above equation, p_i is the fraction of instances in S that belong to class c_i. The entropy of S decreases when the homogeneity of S with respect to the class where each instance belongs increases. For example, entropy is zero when all instances belong to the same class. On the contrary, entropy increases when the homogeneity of the instances decreases. By definition, if p_i is zero, then the term $p_i \log_2(p_i)$ is also equal to zero.

Let $T = \{t_1, \ldots, t_N\}$ be an arranged set of N split points for the values of an attribute A which splits the training set S into $N + 1$ subsets $\{S_1, \ldots, S_{N+1}\}$. Then, information gain G is defined as

$$G(S; A, T) = E(S) - \sum_{i=1}^{N+1} \frac{|S_i|}{|S|} E(S_i)$$

Information gain measures the reduction of the entropy which is caused if the values of attribute A are divided in the $N + 1$ intervals which are generated from the N split points included in T. The main purpose is the minimization of the heterogeneity among the instances that belong to the same interval of the attribute values. In other words, from the candidate splits of the data set, the one which minimizes the entropy and maximizes the information gain should be chosen.

On the contrary, unsupervised discretization is used in the absence of any knowledge of the class memberships of the instances. Thus, unsupervised discretization methods are generally based on the distribution of attribute values. The basic and simplest unsupervised strategy is the *equal-interval-width* discretization, which divides the values of the attribute into k equal bins, where k is a user-specified parameter. This method is vulnerable to outliers, namely observations that are numerically distant from the data, which may skew the range.

Another basic and simple unsupervised strategy is the *equal-frequency-interval* discretization, which divides a continuous attribute into k intervals which include the same number of values. Each interval contains n/k bins where n is the number of values.

These methods are rather simple but have some drawbacks which are very important in AR mining [19]. First of all, discretization must reflect the original distribution of the attribute. Second, discretization intervals should not hide the association and the patterns which exist in the values and, last, intervals should be semantically meaningful and must make sense to human experts.

Some interesting approaches which take under consideration the drawbacks mentioned before have been proposed. Three methods that can be grouped under the term *threshold methods* because they calculate a threshold which helps the discretization process are presented below. These methods [20] were used for discretizing gene expression data that were produced using the *serial analysis of gene expression* (SAGE) method [21].

The first discretization method is called max minus x% and includes the identification of the highest expression value (max) initially and then the replacement of each value that is greater than max-x/100 with the value of unity and the replacement of all other values with the value of zero.

The second approach is the *midrange-based cutoff*. This consists of identifying the highest and lowest expression values for each gene and then the calculation of their arithmetic mean. All expression values below or equal to the arithmetic mean are set to zero and all the expression values exceeding the arithmetic mean are set to unity.

The last method is x% of highest value. This consists of finding the x% of the highest values and replacing them with the value of unity. The rest are assigned the value of zero.

Vanucci and Colla [19] proposed an unsupervised discretization method for dicretizing data for the purposes of AR mining. The main thought is to preserve the original sample distribution. One idea was to use the K-means algorithm in order to generate K number of partitions which reflect the original distribution of the partitioned attribute. The main drawback of using the K-means algorithm was that the result obtained was very sensitive to the value of K because the value must be given by the user before the execution of the algorithm. So a false estimation of the K value could lead to unsatisfactory results. To overcome this disadvantage, a *self-organizing map* (SOM) [22] can be used. The SOM also preserves the initial distribution. The basic advantage of SOMs is that the number of clusters that will be generated is not required to be known in advance, as in the case with K-means. The only parameter that should be given before the execution of the algorithm is the maximum number of clusters (intervals).

An interesting approach in discretizing continuous attributes for AR mining was proposed by Ludl and Widmer [23]. The *relative unsupervised discretization* (RUDE) algorithm combines aspects of both supervised and unsupervised discretization. The method does not require a class attribute and hence belongs to unsupervised methods. Furthermore, the split points for an attribute are constructed in dependence of the other attributes and hence it is called relative. The basic idea when discretizing a particular attribute (the target attribute)

is to consider information about the distribution of the values of the other attributes (source attributes).

34.4.2 Postprocessing of ARs

As already mentioned in the introduction, the large number of ARs that are discovered poses a great challenge in the data-mining research community. The vast amount of generated rules makes interpretation much more complex and often guides to misleading conclusions and consequently to wrong decisions. The assessment of the usefulness of the generated rules becomes an essential necessity and introduces the need to effectively deal with various kinds of redundant and uninteresting data.

Postprocessing usually consists of four phases: pruning, summarizing, grouping, and visualization [24]. In the pruning phase, rules are deleted because they are uninteresting or redundant. The summarizing phase tries to summarize the rules into more general concepts (usually using taxonomies). Then, the remaining rules are grouped into rule packets in the grouping phase. Finally, the extracted useful knowledge is illustrated in a visualization phase. Often the distinction among these phases is not very strict, and there are major interactions among each other. However, they comprise different steps that could be integrated in any postprocessing procedure.

34.5 GENE EXPRESSION DATA MINING

Gene expression is the process by which the genetic information encoded in DNA is converted into a functional product that can be either a protein or an RNA molecule in the case of non-protein-coding genes such as rRNA or tRNA genes.

Each organism contains a number of genes that code the synthesis of an mRNA or protein molecule. Every cell in an organism—with only few exceptions—has the same set of chromosomes and genes. However, two cells may have very different properties and functions. This is due to the differences in abundance of proteins. The abundance of a protein is partly determined by the levels of mRNA, which in turn are determined by the expression levels of the corresponding gene.

A popular tool for the measurement of gene expression is the *microarray* [25]. A microarray experiment measures the relative mRNA levels of thousands of genes, providing the ability to compare the expression levels of different biological samples. These samples may correlate with different time points taken during a biological process or with different tissue types such as normal cells and cancer cells [26]. Another method for measuring gene expression is SAGE, which allows the quantitative profiling of a large number of mRNA transcripts [21]. Although this method is more expensive than microarrays, it has the advantage that the experimenter does not have to preselect the mRNA sequences that will be studied.

The gene expression data are represented by an $M \times N$ matrix (Table 34.7). Biologists conduct a number of experiments measuring gene expression levels of a cell or group of cells under various conditions affecting these expression levels [27]. The M columns of the matrix represent the samples (e.g., microarray experiments or SAGE libraries), which can be related to the type of tissue, the age of the organism, or the environmental conditions. The N columns represent all the genes. The values included in the cells of the matrix are either counts of mRNA molecules (SAGE) or ratios that indicate the variance between the

TABLE 34.7 Typical Gene Expression Matrix

	Gene 1	Gene 2	⋯	Gene N
Sample 1	a_{11}	a_{12}	⋯	a_{1N}
Sample 2	a_{21}	a_{22}	⋯	a_{2N}
⋮	⋮	⋮	⋱	⋮
Sample M	a_{M1}	a_{M2}	⋯	a_{MN}

TABLE 34.8 Horizontal- and Vertical-Data Format

Horizontal Format		Vertical Format	
TID	Items	Items	TID_set
1	G1, G3	G1	1, 4
2	G2, G4, G5	G2	2, 4
3	G4, G5	G3	1
4	G1, G2, G5	G4	2, 3
		G5	2, 3, 4

expression of the respective gene in the particular sample and the expression of the same gene in a control sample (microarrays).

Although the results of the experiments are expressed as real numbers, biologists are usually not interested in those values. These values are used in comparison to normal expression levels in an organism. For that reason, the absolute values are normalized under certain normalization criteria and discretized according to certain predetermined thresholds. After this process the values are grouped under three different levels: unchanged, upregulated (or overexpressed), and downregulated (or underexpressed).

The most common way to mine frequent patterns from a set of transactions is when these transactions are in horizontal-data format, that is, {TID: Itemset}, where TID is the transaction ID and Itemset is the set of items found in that transaction. Another way to mine frequent items (see ECLAT algorithm above) is when the data are in the vertical format, that is, {item: TID_set}. Table 34.8 presents an example database in both horizontal and vertical format. The vertical-data format is popular in application domains, where the data consist of many features (items) and a few samples, as is the case with gene expression data.

34.5.1 Mining in Horizontal-Data Format

As mentioned before, the most popular AR mining algorithm is APRIORI. A detailed example of the application of APRIORI on gene expression data is presented below.

In this example the minimum-support threshold (min_sup) is set to 40% and the minimum-confidence threshold (min_conf) is set to 80%. The matrix presented in Table 34.9 is a discretized gene expression matrix. A value of zero represents a gene that is underexpressed, whereas a value of unity represents a gene that is overexpressed. The samples that represent various experimental conditions C_i are in rows, whereas the genes G_j are in columns. Moreover, the data are transformed in transactional format.

TABLE 34.9 Discretized Gene Expression Matrix Converted in Transactional Format

	G1	G2	G3	G4		TID	Items
C1	1	0	0	0		C1	G1
C2	1	1	0	0		C2	G1, G2
C3	1	0	1	1		C3	G1, G3, G4
C4	1	1	1	1		C4	G1, G2, G3, G4
C5	1	1	1	0	$\Longrightarrow$	C5	G1, G2, G3
C6	0	0	1	1		C6	G3, G4
C7	0	1	1	1		C7	G2, G3, G4
C8	0	1	0	1		C8	G2, G4
C9	0	1	1	0		C9	G2, G3
C10	0	1	1	1		C10	G2, G3, G4

In the first step of the APRIORI algorithm, the support of every single gene (item) is calculated, and the genes that satisfy the minimum-support threshold constitute the set of frequent 1-itemsets. The support for every gene in Table 34.9 is presented below:

- support($\{G1\}$) = 5/10 = 50% $\geq$ min_sup
- support($\{G2\}$) = 7/10 = 70% $\geq$ min_sup
- support($\{G3\}$) = 7/10 = 70% $\geq$ min_sup
- support($\{G4\}$) = 6/10 = 60% $\geq$ min_sup

Consequently the set of frequent 1-itemsets contains all the genes: $L_1 = \{G1, G2, G3, G4\}$.

In the next step, APRIORI generates all the pairs of genes using L_1. The set of candidate frequent 2-itemsets is $C_2 = \{\{G1, G2\}, \{G1, G3\}, \{G1, G4\}, \{G2, G3\}, \{G2, G4\}, \{G3, G4\}\}$. After the construction of C_2, the support of each pair is calculated by scanning the database and counting the appearance of the pairs in each transaction. The support for every candidate frequent 2-itemset is presented below:

- support($\{G1, G2\}$) = 30% < min_sup
- support($\{G1, G3\}$) = 30% < min_sup
- support($\{G1, G4\}$) = 20% < min_sup
- support($\{G2, G3\}$) = 50% $\geq$ min_sup
- support($\{G2, G4\}$) = 40% $\geq$ min_sup
- support($\{G3, G4\}$) = 50% $\geq$ min_sup

Only three out of the six pairs of genes have support greater than or equal to the threshold (40%). Consequently the set of frequent 2-itemsets is $L_2 = \{\{G2, G3\}, \{G2, G4\}, \{G3, G4\}\}$.

In the next step, from the L_2 will arise the C_3, the set which contains all the candidate frequent 3-itemsets. Since the items included in each itemset are ordered according to their ID, then only the itemsets that have the first item in common are merged in order to provide the candidate frequent 3-itemsets. So by merging $\{G2, G3\}$ and $\{G2, G4\}$, the 3-itemset $\{G2, G3, G4\}$ arises that is the only candidate. Moreover, all the subsets of this itemset are frequent, since they are included in L_1 and in L_2, so this itemset is not pruned due to

violation of the downward-closure principle. So $C_3 = \{\{G2, G3, G4\}\}$. Next, the support of this itemset is calculated by scanning the database. Its support is equal to 30%, which is below the minimum-support threshold. As a result, L_3 is an empty set (L3=$\{\}$), thus the frequent itemset mining procedure terminates.

The next step of the APRIORI algorithm includes the generation of the rules which occur from L_2 and then the calculation of the confidence for every rule in order to determine which ones satisfy the minimum-confidence threshold:

- $\{G2, G3\}$
  - $G2 \Rightarrow G3 = 5/7 = 71\% <$ min_conf (discarded)
  - $G3 \Rightarrow G2 = 5/7 = 71\% <$ min_conf (discarded)
- $\{G3, G4\}$
  - $G3 \Rightarrow G4 = 5/7 = 71\% <$ min_conf (discarded)
  - $G4 \Rightarrow G3 = 5/6 = 83\% \geq$ min_conf (accepted)
- $\{G2, G4\}$
  - $G2 \Rightarrow G4 = 4/7 = 57\% <$ min_conf (discarded)
  - $G4 \Rightarrow G2 = 4/6 = 66\% <$ min_conf (discarded).

Finally, only one rule is generated ($G4 \Rightarrow G3$). It is very clear that if the minimum-confidence threshold had been set to 70%, then four rules would have been generated. This indicates that it is very important to make a careful choice of the interestingness measures (e.g., support and confidence) thresholds before mining the ARs. The AR that was generated could mean that when gene 4 (G4) is overexpressed, then it is also likely, with a high possibility (83%), that gene 3 (G3) is also overexpressed.

The majority of frequent pattern-mining algorithms are based on the APRIORI candidate generation procedure. The main drawback of these methods is the high computational cost of the evaluation of all candidates. For this reason a lot of algorithms that are inspired by the FP-GROWTH algorithm and exploit the use of a tree structure have been proposed. Kotala et al. [28] proposed a method for mining microarray data using *Peano count trees* (P-Trees). This method treats the microarray data as spartial data.

Two interesting approaches which are based on the frequent closed-pattern idea are [29] and [30]. The definition of a closed pattern as is given by Han et al. [31], states that a is a closed frequent pattern in a database D if a is frequent in D and there exists no proper superpattern b such that b has the same support as a in D. Mining closed itemsets provides an interesting alternative to mining frequent itemsets because it generates a much smaller set of results and so it achieved better scalability and interpretability.

34.5.2 Mining in Vertical-Data Format

As already mentioned before, gene expression data usually contain a very large number of columns, which represent the genes, in comparison to rows, which are the experiments. For example, a gene expression matrix may contain 10,000–100,000 columns but only 100–1000 rows. As a result, it became obvious that the methods which use the horizontal-data format for these data sets are not very suitable. Thus, many new algorithms were proposed in order to handle high-dimensional data, such as gene expression data, in vertical-data format.

The first algorithm designed to handle a microarray data set in vertical-data format was CARPENTER [32]. CARPENTER discovers frequent closed patterns by performing depth-first rowwise enumeration instead of the conventional feature (column) enumeration. Furthermore, pruning techniques are used in order to optimize the algorithm's efficiency.

Another approach, called FARMER, has been proposed by Cong et al. [33]. FARMER finds interesting rule groups and builds classifiers based on them. The concept of rule groups implies that rules supported by exactly the same set of rows are grouped together. FARMER is designed specifically to generate rules of the form $X \Rightarrow C$, where X is a set of genes and C is a class label. For that reason, each experiment in microarray data should be related to a class label, such as cancer or noncancer.

Pan et al. extended the CARPENTER algorithm in order to handle data sets with large numbers of both columns and rows. The algorithm, which is called COBBLER [34], switches dynamically between column and row enumeration based on the estimated cost of processing. Moreover, COBBLER is more efficient than the previously mentioned algorithms: CHARM, CLOSET+, and CARPENTER.

Finally, another interesting approach was proposed in 2006 by Liu et al. [35]. They developed an algorithm, TD-CLOSE, to find the complete set of frequent closed itemsets. The main difference with the existing approaches is that TD-CLOSE adapts a top-down row enumeration search strategy which enables the use of the minimum-support threshold to dramatically prune the search space.

34.6 SEQUENTIAL DATA MINING

Technological advances of the last decades have driven the collection of vast amounts of biological sequences. After the completion of genome-sequencing projects, the sequenced genomes have to be analyzed and annotated. The observed paradigm shift from static structural genomics to dynamic functional genomics [36] and the assignment of functional information to known sequences are considered particularly important. Gene prediction is the step that usually follows sequencing and is concerned with the identification of stretches of DNA that are biologically functional. As it is not a straightforward task, especially for the more complex eukaryotic genomes, the use of advanced techniques, including data mining, is required.

For example, eukaryotic genes consist of coding parts, called *exons*, that are separated by intervening noncoding sequences, called *introns*. Introns are removed from the transcribed RNA sequence by the process of splicing. The recognition of the splice sites, namely the boundaries between adjacent exons and introns, is a difficult problem that exploits data-mining methods. The problem becomes even more challenging if one considers the possibility of alternative splicing, that is, the production of different mature mRNA molecules, depending on the number of the exons that are finally concatenated.

Other important sequence analysis tasks are the prediction of regulatory regions, that is, promoters and enhancers, which are segments of DNA where regulatory proteins bind preferentially and thus control gene expression and consequently protein abundance. The prediction of the transcription start site, where transcription of DNA to RNA starts, the prediction of translation initiation site, where translation of mRNA to protein initiates, and the prediction of polyadenylation sites where a polyA (multiple adenines) tail is added at the $3'$ end of an mRNA sequence are also some important sequence analysis tasks.

Quite often, around these important sites there are found some signals or patterns that appear with variable frequency in each case. These patterns could be exported using frequent itemset mining algorithms. Moreover, various patterns or sequence parts that are found near specific sequence signals could be associated to each other using ARs. In most cases the representation of a particular biological sequence is accomplished by a number of features that should be extracted from this sequence. These features usually record the frequencies of some variable-length nucleotide patterns. In such a case, if a frequent-itemset mining algorithm should be used, it is essential to apply a discretization method first in order to transform the continuous-valued features to categorical ones.

An approach which used association analysis and combined gene expression and biological sequences was proposed by Icev et al. [37]. In their approach they focused on characterization of the expression patterns of genes based on their promoter regions. The promoter region contains short sequences called motifs to which may bind gene regulatory proteins which possibly control the gene expression mechanisms. The *distance-based AR mining* (DARM) algorithm is based on the APRIORI algorithm. DARM has the ability to involve multiple motifs and to predict expression in multiple cell types. Moreover, ARs in DARM are enhanced with information about the distances among the motifs that are present in the rules in order to investigate whether the order and spacing of the motifs can affect expression.

Another category of frequent pattern that can be used for discriminating two classes is the emerging patterns. In a recent study [38], a method called PolyA-iEP, which exploits the advantages of emerging patterns as well as a distance-based scoring method, has been proposed. This method aims to effectively predict polyadenylation sites in biological sequences and can be used for both descriptive and predictive analysis.

Emerging patterns [39] are itemsets whose supports increase significantly from one data set to another. Given two data sets D_1 and D_2, the *growth rate* of an itemset X from D_1 to D_2 is defined as follows (indices 1 and 2 are used instead of D_1 and D_2):

$$\mathrm{gr}_{1\to 2}(X) = \begin{cases} 0 & \text{if } \mathrm{supp}_1(X) = 0 \text{ and } \mathrm{supp}_2(X) = 0 \\ \infty & \text{if } \mathrm{supp}_1(X) = 0 \text{ and } \mathrm{supp}_2(X) > 0 \\ \frac{\mathrm{supp}_2(X)}{\mathrm{supp}_1(X)} & \text{otherwise} \end{cases}$$

Given a minimum growth rate threshold $\rho > 1$, an itemset X is said to be *ρ-emerging pattern*, or simply an *emerging pattern*, from D_1 to D_2 if $\mathrm{gr}_{1\to 2}(X) \geq \rho$, where D_1 is called the *background data set* and D_2 the *target data set*.

The *strength* of an emerging pattern X from D_1 to D_2 is defined as

$$\mathrm{strength}_{1\to 2}(X) = \begin{cases} \mathrm{supp}_2(X) & \text{if } \mathrm{gr}_{1\to 2}(X) = \infty \\ \mathrm{supp}_2(X)\frac{\mathrm{gr}_{1\to 2}(X)}{\mathrm{gr}_{1\to 2}(X)+1} & \text{otherwise} \end{cases}$$

In contrast to other patterns or models, emerging patterns are easily interpretable and understood. Moreover, emerging patterns, especially those with a large growth rate and strength, provide a great potential for discriminating examples of different classes. This twofold benefit of emerging patterns makes them a useful tool for exploring domains that are not well understood, providing the means for descriptive and predictive analysis as well.

A disadvantage of emerging pattern mining is that the number of emerging patterns may be huge, especially when minimum-support and minimum-growth-rate thresholds are set very low. Increasing the thresholds is not an ideal solution, since valuable emerging

patterns may not be discovered. For example, if the minimum-support threshold is set high, then those emerging patterns with a low support but with a high growth rate will be lost. Conversely, if the minimum-growth-rate threshold is set high, then those emerging patterns with a low growth rate but with a high support will be lost. Some interestingness measures have been proposed to reduce the number of mined emerging patterns without sacrificing valuable emerging patterns, or at least sacrificing as less as possible. Such an interestingness measure includes a special kind of emerging pattern, called a *chi emerging pattern* [40], defined as follows: Given a background dataset D_1 and a target dataset D_2, an itemset X is called a *chi emerging pattern* if all the following conditions are true:

1. $\text{supp}_2(X) \geq \sigma$, where σ is a minimum support threshold.
2. $\text{gr}_{1\to 2}(X) \geq \rho$, where ρ is a minimum growth rate threshold.
3. $\forall Y \subset X, gr_{1\to 2}(Y) < gr_{1\to 2}(X)$
4. $|X| = 1 \vee |X| > 1 \wedge (\forall Y \subset X \wedge |Y| = |X| - 1 \wedge \text{chi}(X, Y) \geq \eta)$, where $\eta = 3.84$ is a minimum chi value threshold and chi(X, Y) is computed using the chi-squared test.

The first condition ensures that the mined emerging patterns will have at least a minimum coverage over the training data set in order to generalize well on new instances. The second condition ensures that the mined emerging patterns will have an adequate discriminating power. The third condition is used to filter out those emerging patterns that have a subset with higher or equal growth rate and higher or equal support (any itemset has equal or greater support than any of its supersets). Since the subset has fewer items, there is no reason to keep this emerging pattern. Finally, the fourth condition ensures that an emerging pattern has a significantly (95%) different support distribution in target and background data sets than the distributions of its immediate subsets.

34.7 STRUCTURAL DATA MINING

Structural bioinformatics is the subfield of bioinformatics which is related to the analysis and prediction of the three-dimensional structure of biological macromolecules, especially for proteins. The application of machine learning and data mining in structural bioinformatics is quite challenging, since structural data are not linear. Moreover, the search space for most structural problems is continuous and infinite and demands highly efficient and heuristic algorithms.

Important problems of structural bioinformatics that utilize machine learning and data-mining methods are the RNA secondary-structure prediction, the inference of a protein's function from its structure, the identification of protein–protein interactions, and the efficient design of drugs based on structural knowledge of their target.

Machine learning and data-mining methods are also applied for protein secondary-structure prediction. This problem has been studied for over 35 years and many techniques have been developed. Initially, statistical approaches were adopted to deal with this problem. Later, more accurate techniques based on information theory, Bayes theory, nearest neighbors, and neural networks were developed. Combined methods such as integrated multiple-sequence alignments with neural network or nearest-neighbor approaches improve prediction accuracy.

Secondary-structure prediction methods can be divided in four generations [41]: First-generation methods were based on single amino acid propensities. Second-generation

methods used prospensities of 3–51 adjacent residues. The prediction accuracy was at 60% [42]. The accuracy is defined as the percentage of residues predicted correctly in one of the tree states: helix, strand, and other. Third-generation methods used information from homologue sequences and machine learning methods. Lastly, in fourth-generation methods, a matching between secondary and tertiary protein structure was used. Using the fourth-generation approaches, the accuracy reached around 77%. Nowadays a great amount of research has been focused on better representations of secondary-structure features which is believed will improve significantly the prediction accuracy of the algorithms. The better representation is achieved by discovering frequent patterns in protein databases.

An approach to the representation of secondary structure has been proposed by Birzele and Kramer [43]. In their approach they used the levelwise search strategy [44], which is a string-mining algorithm, in order to find frequent patterns in protein databases. From those frequent patterns were extracted features to be used in the prediction of the secondary structure using *support vector machines* (SVMs).

Finally, Beccerra et al. [45] proposed an algorithm for biological sequence feature classification. The approach includes two main features. First, the use of association analysis in order to extract interesting relationships hidden in biological data sets and, second, the use of machine learning classifiers trained with the data obtained from the association analysis. More specifically, the first feature consists of three main phases. First, the sequences are scanned as subsequences of N symbols (N-grams). The N-grams represent patterns of variable size in the biological sequence and are represented as binary vectors which allow the application of association analysis. The second step consists of finding frequent patterns in the sequences. The third step consist of finding ARs. In this approach the APRIORI algorithm was used for the AR extraction process.

34.8 PROTEIN INTERACTIONS: GRAPH DATA MINING

Proteins are the most versatile macromolecules in living systems. They are the building blocks from which the cells are assembled, and they constitute most of the cell's dry mass. Proteins not only provide the cell with shape and structure but also execute nearly all its numerous functions [46, 47]. Following are some of their numerous and versatile functions:

- Function as catalyst in chemical reactions. Enzymes are proteins that increase the rates of chemical reactions.
- Transport and store other molecules such as oxygen. For example, hemoglobin carries oxygen to the erythrocytes (red blood cells).
- Provide mechanical support. For example, collagen is the main component of connective tissue and is the most abundant protein in mammals.
- Immune protection. Antibodies are proteins used by the immune system to identify and neutralize foreign objects like bacteria and viruses.
- Generate movement. For instance, myosin in skeletal muscle cells provides the motive force for humans to move.
- Detects and transmits nerve impulses to the cells' response machinery. For instance, rhodopsin in the retina detects light.
- Control growth and differentiation.

It is obvious that proteins have a wide range of functions. In fact, a protein almost never performs its function in isolation. It should interact with other proteins in order to accomplish a certain function. It has been discovered that the vast majority of proteins interact with multiple partners (on average six to eight other proteins) and thousands of different proteins form intricate interaction networks or highly regulated pathways [48]. The most common way for the representation of these networks are as undirected graphs. In these graphs the proteins are represented as nodes and the interactions are represented as edges between two nodes (proteins).

The last years many interactions have been discovered by researchers. This has been achieved through new high-throughput methods which have been recently proposed. The huge number of interactions discovered have been stored in many databases. Xenarios and Eisenberg [49] have reviewed and presented many of them, including DIP, BIND, MIPS, PROTEOME, PROTONET, CURAGEN, and PIM.

Although a huge amount of information is contained in these databases, there are several issues associated with them and the most important is the large amount of noise that is present in high-throughput interaction data [50]. The noise can affect the efficacy of the algorithms at the task of function prediction between different data sets [51].

Recently, some data-mining techniques have been proposed to determine the reliability of a given interaction. One of them [52] uses the *h-confidence* measure [53] from the field of association analysis which can be used to estimate the similarity between two proteins based on the number of their shared neighbors. The importance of using the *h*-confidence measure for all pairs of proteins in this network is twofold. First, it can address the problem of noise in the data mentioned before. For instance, when an interaction is already known, the *h*-confidence measure between the two particular proteins is low. Second, it can also address another important problem of the interaction data, which is the problem of incompleteness. For example, if an interaction between two proteins is not known and the *h*-confidence measure for these proteins is high, then it is probable that an interaction between those two proteins exists.

Association analysis can be used to predict protein functions from a protein interaction network. In the context of graphs the frequent patterns stand for the frequent subgraphs in a set of graphs [54]. In the context of protein interaction networks a set of graphs may be a set of protein interaction networks and frequent patterns (subgraphs) may be functional modules. The discovered network modules can be used in many biological applications such as the prediction of the function of unknown genes or the construction of the transcription modules.

Hu et al. [55] developed an algorithm, called CODENCE, to efficiently mine frequent coherent dense subgraphs across a large number of massive graphs. The two main steps of the algorithm include the construction of a summary graph across multiple relation graphs $G_1, \ldots, G_n$ and then the mining of dense summary graphs using the MODES algorithm [56]. It is worth mentioning that this method can integrate heterogeneous network data, such as protein interaction networks, genetic interaction networks, and coexpression networks to reveal consistent biological signals.

Also, Xiong et al. [57] proposed a hyperclique pattern discovery approach in order to extract functional modules from protein complexes. A *hyperclique pattern* is a type of association pattern that contains highly affiliated proteins, that is, every pair of proteins in the same hyperclique pattern is guaranteed to have the cosine similarity above a certain level. For that reason, if a protein is found to belong in a protein complex, then it is very probable that the other proteins in the same hyperclique pattern also belong to the same

protein complex. The *h*-confidence measure mentioned before is designed to capture the strength of this association. The definitions of *h-confidence* and hyperclique pattern are given below according to [57]:

- The *h*-confidence of a pattern $X = \{p_1, p_2, \ldots, p_m\}$, denoted as hconf($X$), is a measure that reflects the overall affinity among proteins within the pattern. This measure is defined as min(conf($\{p_1\} \Rightarrow \{p_2, p_3, \ldots, p_m\}$), conf($\{p_2\} \Rightarrow \{p_1, p_3, \ldots, p_m\}$), conf($\{p_3\} \Rightarrow \{p_1, p_2, \ldots, p_m\}$),... conf($\{p_m\} \Rightarrow \{p_1, p_2, \ldots, p_{m-1}\}$), where conf is the confidence of AR.
- A pattern X is a hyperclique pattern if hconf(X) $\geq h_c$, where h_c is a user-specified minimum *h*-confidence threshold. A hyperclique pattern is a maximal hyperclique pattern if no superset of this pattern is also a hyperclique pattern.

Another approach to the field of protein interaction networks was proposed by Besemann et al. [58], who introduced the concept of differential AR mining to study the annotations of proteins in the context of one or more interaction networks. The goal of this technique was to highlight the differences among items belonging to different interacting nodes or different networks, something that could not be achieved with the standard relational AR mining techniques.

Another perspective to the field of protein interaction networks is to find frequent subnetworks in a given network in a different manner than in the previously described approaches which focused on mining frequent subgraphs in a set of networks. Such an algorithm is NeMoFINDER, which was proposed by Chen et al. [59] and was inspired by the APRIORI algorithm.

34.9 TEXT MINING

Text mining in molecular biology, defined as the automatic extraction of information about genes, proteins, and their functional relationships from text documents [60], has emerged as a hybrid discipline on the edges of the fields of information science, bioinformatics, and computational linguistics.

It is critically important for biologists to have access to the most up-to-date information on their field of research. Current research practice involves online search for gene-related information utilizing the latest technologies in information retrieval, semantic Web, and text mining. A new term lately used by bioinformaticians to describe the text body where they can extract information such as ontology, interaction, and function between biological entities is the *textome*. Generally, it can include all parseable and computable scientific text body.

The rapid progress in biomedical research has led to a dramatic increase in the amount of available information in terms of published articles, journals, books, and conference proceedings. Pubmed is a free database accessing primarily the MEDLINE database of references and abstracts on life sciences and biomedical topics. As of July 1, 2011, PubMed has over 21 million records, 11.0 milion articles are listed with their abstracts, and 3.3 milion articles are available full text for free. Every year about 500,000 new records are added. In total, more than 5000 journals are currently indexed by PubMed. Although PubMed is by far the richest database of abstracts, citations, and full text articles, there is a plethora of

such sources of scientific publications on biology such as NCBI BookShelf for e-books and a large number of online resources. The researchers' need to exploit this enormous volume of available information, along with the availability of high-performance and efficient data mining, natural language processing, and information retrieval tools, has given birth to a new field of research and application called *bioinformatics text mining* (BTM). Other terms for BTM are bio(logical) text mining and biomedical text mining.

From a data miner's point of view, biomedical literature has certain characteristics that require special attention, such as heavy use of domain-specific terminology, polysemic words (word sense disambiguation), low-frequency words (data sparseness), creation of new names, and terms and different writing styles [61].

Several studies have categorized the tasks of BTM from different points of view. Cohen and Hersh [62] provide a high-level categorization identifying the main tasks to be the following:

- *Named entity recognition* (NER) The goal is to find and classify atomic elements in text into predefined categories.
- *Text Classification* The goal is to determine whether a document has particular characteristics, usually based on whether the document includes certain types of information.
- *Synonym and Abbreviation Extraction* This task deals with the problem that many biological entities have multiple names, so in the biomedical literature there are many synonyms and abbreviations.
- *Relationship Extraction* The goal of relationship extraction is to find a specific type of relationship between a pair of entities of given types.
- *Hypothesis Generation* The goal is to find relationships that are not present in text but are inferred by other explicit relationships.

An early work which used ARs for text mining was proposed by Hristovski et al. [63]. The goal of the system they presented was to discover new, potentially meaningful relations of a given concept of interest with other concepts that have not been published in the medical literature before. All the known relations among the concepts came from MEDLINE. Each citation in MEDLINE is associated with a set of *medical subject heading* (MeSH) terms that describe the content of the item in the database. The main idea was the use of ARs to find all concepts Y that are related to the starting concept X and then to find all the concepts Z that are related to the concept Y. The next step included examination of whether the concepts X and Z are found together in the medical literature. If they do not appear, it is possible that a new relation has been discovered. Evaluation of the discovered associations was done by human experts, laboratory methods, or clinical investigations, depending of the nature of X and Z.

Another interesting approach that used ARs for biomedical text mining was proposed by Berardi et al. [64]. The purpose of their method was to detect associations between concepts as an indication of the existence of the biomedical relation. This method also used the MEDLINE abstract and MeSH taxonomy. The hierarchical nature of the MeSH taxonomy made it possible to mine multilevel ARs (generalized ARs) [9]. Generalized ARs include ARs of the form $X \Rightarrow Y$, where no item in Y is an ancestor of any item in X in a given taxonomy.

34.10 CONCLUSION

Frequent patterns and ARs are useful data-mining tools that have attracted research interest since 1993 as a means of unsupervised, exploratory data analysis. Although they were initially proposed as a market basket analysis tool, they were almost immediately applied to other application domains and nowadays include a large number of applications. The community of biologists and bioinformaticians have used ARs for analyzing a quite variable set of biological data. Gene expression data, biological sequences, biological structural data, protein interaction networks, and biological texts are the most popular kinds of biological data that have been effectively analyzed using these data-mining tools.

As already done in the past, it is deemed that several new algorithms for mining ARs more efficiently as well as for mining new kinds of patterns and extending the concept of ARs will be proposed in the future. All these novel AR mining tools will provide the means for more efficient and effective analyses of biological data. As a result, the research efforts of biologists will be enhanced by the gain of new biological insights and the rise of new biological questions that will guide unexplored research directions.

REFERENCES

1. R. Agrawal, T. Imielinski, and A. Swami. Mining association rules between sets of items in large databases. In P. Buneman and S. Jajodia (Eds.), *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, Washington, DC, May 26–28, 1993, ACM Press, 1993.
2. R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. J. B. Bocca, M. Jarke, and C. Zaniolo (Eds.). *Proceedings of the 20th International Conference on Very Large Data Bases (VLDB'94)*, Santiago de Chile, Chile, September 12–15, 1994, Morgan Kaufmann, 1994.
3. H. Mannila, H. Toivonen, and A. I. Verkamo. Efficient algorithms for discovering association rules. In U. M. Fayyad and R. Uthurusamy (Eds.), *Knowledge Discovery in Databases: Papers from the 1994 AAAI Workshop*, Technical Report WS-94-03, AAAI Press, Seattle, WA, July 1994, pp. 181–192.
4. J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In W. Chen, J. F. Naughton, and P. A. Bernstein (Eds.), *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, Dallas, TX, May 16–18, 2000, ACM, 2000, pp. 1–12.
5. M. J. Zaki. Scalable algorithms for association mining. *IEEE Trans. Knowledge Data Eng.*, 12:372–390, 2000.
6. K. Koperski and J. Han. Discovery of spatial association rules in geographic information databases. In M. J. Egenhofer and J. R. Herring (Eds.), *Advances in Spatial Databases: Proceedings of the 4th International Symposium (SSD'95)*, Portland, ME, USA, August 6–9, 1995, *Lecture Notes in Computer Science*, Springer, 1995, pp. 47–66.
7. X. Chen and I. Petrounias. Discovering temporal association rules: Algorithms, language and system. In *Proceedings of the 16th International Conference on Data Engineering*, 2000.
8. A. K. H. Tung, H. Lu, J. Han, and L. Feng. Efficient mining of intertransaction association rules. *IEEE Trans. Knowledge Data Eng.*, 15(1):43–56, 2003.
9. R. Srikant and R. Agrawal. Mining generalized association rules. In U. Dayal, P. M. D. Gray, and S. Nishio, *Proceedings of 21st International Conference on Very Large Data Bases (VLDB'95)*, Zurich, Switzerland, September 11–15, 1995, Morgan Kaufmann, 1995, pp. 407–419.

10. S. Thomas and S. Sarawagi. Mining generalized association rules and sequential patterns using SQL queries. In R. Agrawal, P. E. Stolorz, and G. Piatetsky-Shapiro (Eds.), *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD'98)*, New York City, NY, USA, August 27–31, 1998, AAAI Press 1998, pp. 344–348.
11. J. Han and Y. Fu. Discovery of multiple-level association rules from large databases. In U. Dayal, P. M. D. Gray, S. Nishio, *Proceedings of 21st International Conference on Very Large Data Bases (VLDB'95)*, Zurich, Switzerland, September 11–15, 1995, Morgan Kaufmann, 1995, pp. 420–431.
12. A. Savasere, E. Omiecinski, and S. B. Navathe. Mining for strong negative associations in a large database of customer transactions. In *Proceedings of 14th International Conference on Data Engineering*, Orlando, FL, February 23–27, 1998, pp. 494–502.
13. X. Wu, C. Zhang, and S. Zhang. Efficient mining of both positive and negative association rules. *ACM Trans. Inform. Syst.*, 22(3):381–405, 2004.
14. G. Tzanis and C. Berberidis. Mining for mutually exclusive items in transaction databases. *Int. J. Data Warehousing Mining*, 3(3):45–59, 2007.
15. G. Tzanis, C. Berberidis, and I. Vlahavas. On the discovery of mutually exclusive items in a market basket database. In *Proceedings of the 2nd ADBIS Workshop on Data Mining and Knowledge Discovery*, Thessaloniki, Greece, 2006.
16. F. H. C. Crick. On protein synthesis. *Sympo. Soc. Exp. Biol.*, 12:139–163, 1958.
17. J. Dougherty, R. Kohavi, and M. Sahami. Supervised and unsupervised discretization of continuous features. In A. Prieditis and S. J. Russell (Eds.), *Machine Learning: Proceedings of the Twelfth International Conference on Machine Learning*, Tahoe City, CA, USA, July 9–12, 1995, Morgan Kaufmann, 1995.
18. U. Fayyad and K. Irani. On the handling of continuous-valued attributes in decision tree generation. *Machine Learning*, 8:87–102, 1992.
19. M. Vannucci and V. Colla. Meaningful discretization of continuous features for association rules mining by means of a SOM. In *Proceedings of the ESANN2004 European Symposium on Artificial Neural Networks*, Belgium, 2004, pp. 489–494.
20. C. Becquet, S. Blachon, B. Jeudy, J.-F. Boulicaut, and O. Gandrillon. Strong-association-rule mining for large-scale gene-expression data analysis: A case study on human SAGE data. *Genome Biol.*, 3(12): 2002.
21. V. E. Velculescu, L. Zhang, B. Vogelstein, and K. W. Kinzler. Serial analysis of gene expression. *Science*, 270(5235):484–487, 1995.
22. T. Kohonen. The self-organizing map. *Proc. IEEE*, 78(9):1464–1480, 1990.
23. M.-C. Ludl and G. Widmer. Relative unsupervised discretization for association rule mining. In *Proceedings of the Fourth European Conference on Principles and Practice of Knowledge Discovery in Databases*, 2000.
24. B. Baesens, S. Viaene, and J. Vanthienen. Postprocessing of association rules. In *Proceedings of the Workshop Post Processing in Machine Learning and Data Mining: Interpretation, Visualization, Integration, and Related Topics, Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Boston, MA, 2000.
25. M. Schena, D. Shalon, R. W. Davis, and P. O. Brown. Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science* 270(5235):467–470, 1995.
26. K. Aas. Microarray data mining: A survey. NR Note. SAMBA, Norwegian Computing Center, Oslo, 2001.
27. A. Tuzhilin and G. Adomavicius. Handling very large numbers of association rules in the analysis of microarray data. In *Proceedings of the Eighth ACM SIGKDD International Conference on Data Mining and Knowledge Discovery*, Edmonton, Canada, July 23–26, 2002, pp. 396–404.

28. P. Kotala, A. Perera, and J. K. Zhou. Gene expression profiling of DNA microarray data using peano count tree (p-trees). In *Proceedings of the First Virtual Conference on Genomics and Bioinformatics*, North Dakota State University, 2001, pp. 15–16.
29. J. Wang, J. Han, and J. Pei. Closet+: Searching for the best strategies for mining frequent closed itemsets. In L. Getoor, T. E. Senator, P. Domingos, and C. Faloutsos (Eds.), *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Washington, DC, USA, August 24–27, 2003, ACM, 2003.
30. M. J. Zaki and C. J. Hsiao. CHARM: An efficient algorithm for closed itemset mining. In *Proc. 2002 SIAMInt. Conf. Data Mining*, Arlington, VA, 2002, pp. 457–473.
31. J. Han, H. Cheng, D. Xin, and X. Yan. Frequent pattern mining: Current status and future directions. *Data Mining Knowledge Discov.*, 15(1):55–86, 2007.
32. F. Pan et al. Carpenter: Finding closed patterns in long biological datasets. In L. Getoor, T. E. Senator, P. Domingos, and C. Faloutsos (Eds.), *Proceedings of the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Washington, DC, USA, August 24–27, 2003, pp. 637–642.
33. G. Cong, A. K. H. Tung, X. Xu, F. Pan, and J. Yang. Farmer: Finding interesting rule groups in microarray datasets. *Proceedings of the 2004 ACM SIGMOD International Conference on Management of Data*, Paris, France, June 13–18, 2004.
34. F. Pan, A. Tung, G. Cong, and X. Xu. COBBLER: Combining column and row enumeration for closed pattern discovery. In *Proceedings of the 16th International Conference on Scientific and Statistical Database Management*, 2004, pp. 21–30.
35. H. Liu, J. Han, D. Xin, and Z. Shao. Mining frequent patterns from very high dimensional data: A top-down row enumeration approach. In *Proceeding of the 2006 SIAM International Conference on Data Mining (SDM 06)*, Bethesda, MD, Citeseer, 2006, pp. 280–291.
36. J. L. Houle, W. Cadigan, S. Henry, A. Pinnamaneni, and S. Lundahl. Database mining in the Human Genome Initiative. Whitepaper, Bio-databases.com, Amita Corporation. Available: `http://www.biodatabases.com/whitepaper01.html`, last Accessed July 15, 2011.
37. A. Icev, C. Ruiz, and E. F. Ryder. Distance-based association rules mining. M. J. Zaki, J. T.-L. Wang, and H. Toivonen (Eds.), *Proceedings of the 3rd ACM SIGKDD Workshop on Data Mining in Bioinformatics (BIOKDD'03)*, August 27, 2003, Washington, DC, USA, 2003, pp. 34–40.
38. G. Tzanis, I. Kavakiotis, and I. Vlahavas. PolyA-iEP: A data mining method for the effective prediction of polyadenylation sites. *Expert Syst. Appl.*, 38(10):12398–12408, 2011.
39. G. Dong and J. Li. Efficient mining of emerging patterns: Discovering trends and differences. In U. M. Fayyad, S. Chaudhuri, and D. Madigan (Eds.), *Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Diego, CA, USA, August 15–18, 1999, ACM 1999, pp. 43–52.
40. H. Fan. Efficient mining of interesting emerging patterns and their effective use in classification. Ph.D. Thesis. University of Melbourne, Australia, 2004.
41. F. Birzele. Data mining for protein secondary structure prediction. Master's thesis. Technische Universitδt Mónchen, Jan. 2005.
42. B. Rost. Review: Protein secondary structure prediction continue torise. *J. Struct. Biol.*, 134:204–218, 2001.
43. F. Birzele and S. Kramer. A new representation for protein secondary structure prediction based on frequent patterns. *Bioinformatics*, 22:2628–2634, Nov. 2006.
44. H. Mannila and H. Toivonen. Levelwise search and borders of theories in knowledge discovery. *Data Mining Knowledge Discov.*, 3:241–258, 1997.
45. D. Becerra, D. Vanegas, G. Cantor, and L. Niño. An association rule based approach for biological sequence feature classification. In *Proceedings of the Eleventh CEC*, IEEE, 2009, pp. 3111–3118.

46. B. Alberts, D. Bray, K. Hopkin, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter. *Essential Cell Biology*, 2nd ed. Garland Publishing, 2004.
47. L. Stryer. *Biochemistry*, 3rd ed. W. H. Freeman, 1988.
48. A. Panchenko and T. Przytycka. *Protein–Protein Interactions and Networks. Identification, Computer Analysis and Prediction*. Springer, 2008.
49. I. Xenarios and D. Eisenberg. Protein interaction databases. *Curr. Opin. Biotechnol.*, 12:334–339, 2001.
50. G. Pandey, V. Kumar, and M. Steinbach. Computational approaches for protein function prediction: A survey. Technical Report 06-028. Department of Computer Science and Engineering, University of Minnesota, Twin Cities, MN, 2006.
51. M. Deng, F. Sun, and T. Chen. Assessment of the reliability of protein–protein interactions and protein function prediction. *Pac. Symp. Biocomput.*, 140–151, 2003.
52. G. Pandey, M. Steinbach, R. Gupta, T. Garg, and V. Kumar. Association analysis-based transformations for protein interaction networks: A function prediction case study. In P. Berkhin, R. Caruana, and X. Wu (Eds.), *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Jose, CA, USA, August 12–15, 2007, pp. 540–549.
53. H. Xiong, P.-N. Tan, and V. Kumar. Hyperclique pattern discovery. *Data Mining Knowledge Discov.*, 13(2):219–242, 2006.
54. M. Kuramochi and G. Karypis. An efficient algorithm for discovering frequent subgraphs. *IEEE Trans. Knowledge Data Eng.*, 16(9):1038–1051, 2004.
55. H. Hu, X. Yan, Y. Huang, et al. Mining coherent dense subgraphs across massive biological networks for functional discovery. *Bioinformatics*, 21, 2005.
56. E. Hartuv and R. Shamir. A clustering algorithm based on graph connectivity. *Inform. Process. Lett.*, 76:4–6, 175–181, 2000.
57. H. Xiong, X. He, C. Ding, Y. Zhang, V. Kumar, S. R. Holbrook. Identification of functional modules in protein complexes via hyperclique pattern discovery. In *Proc. Pacific Symposium on Biocomputing (PSB)*, 2005, pp. 221–232.
58. C. Besemann, A. Denton, and A. Yekkirala. Differential association rule mining for the study of protein-protein interaction networks. Paper presented at BIOKDD04: 4th Workshop on Data Mining in Bioinformatics (with SIGKDD Conference), Seattle, WA, USA, 2004, pp. 72–80.
59. J. Chen, W. Hsu, M. L. Lee, and S.-K. Ng. NeMoFinder: Dissecting genome-wide protein–protein interactions with meso-scale network motifs. In T. Eliassi-Rad, L. H. Ungar, M. Craven, and D. Gunopulos (Eds.), *Proceedings of the Twelfth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Philadelphia, PA, USA, August 20–23, 2006. ACM, 2006, pp. 106–115.
60. M. Krallinger and A. Valencia. Text-mining and information-retrieval services for molecular biology. *Genome Biol.*, 6:224, 2005.
61. G. Tzanis, C. Berberidis, and I. Vlahavas. Machine learning and data mining in bioinformatics. In L. C. Rivero, J. H. Doorn, and V. E. Ferraggine (Eds.), *Handbook of Research on Innovations in Database Technologies and Applications: Current and Future Trends*. IGI Global, 2009.
62. A. M. Cohen and W. R. Hersh. A survey of current work in biomedical text mining. *Brief. Bioinformatics*, 6(1):57–71, 2005.
63. D. Hristovski, J. Stare, B. Peterlin, and S. Dzeroski. Supporting discovery in medicine by association rule mining in Medline and UMLS. *Stud. Health Technol. Inform.*, 84(Part 2):1344–1348, 2001.
64. M. Berardi, M. Lapi, P. Leo, and C. Loglisci. Mining generalized association rules on biomedical literature. *Lecture Notes Comput. Sci.*, 3533:500–509, 2005.