# Mining Frequent Patterns and Association Rules from Biological Data

*Ioannis Kavakiotis, George Tzanis, Ioannis Vlahavas*

## 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 rules mining has attracted the attention of the data mining research community since the early 90s, as a means of unsupervised, exploratory data analysis. Association rules were first introduced by Agrawal et al. (1993) as a market basket analysis tool, however since then, they have been effectively applied to many other application domains, including biology and bioinformatics. An association rule implies the co-existence of a number of items in 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 co-existence of high expression values of a number of genes in the same transactions-experiments indicates a possible co-expression pattern of these genes. Conversely, the rare or the absolutely non-co-existence of two products could also imply a negative association (e.g. a mutual exclusion) among them.

Many algorithms for mining association rules and others that extend the concept of association rules mining have been proposed so far. Agrawal and Srikant (1994) proposed Apriori, the first algorithm for mining association rules. Apriori is a level-wise 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. (1994) discovered independently a variation of Apriori, the OCD algorithm. The large number of algorithms that have been proposed since then, either improve the efficiency, such as FPGrowth (Han et al., 2000) and Eclat (Zaki, 2000), or address different problems from various application domains, such as spatial (Koperski and Han, 1995), temporal (Chen and Petrounias, 2000) and intertransactional association rules (Tung et al., 2003).

One of the major problems in association rules 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 (1995) presented the problem of mining for generalized association rules. These rules utilize item taxonomies (concept hierarchies) in order to discover more interesting rules. Thomas and Sarawagi (1998) proposed a technique for mining generalized association rules based on SQL queries. Han and Fu (1995) also describe the problem of mining "multiple-level" association rules, based on taxonomies and propose a set of top-down progressive deepening algorithms.

Another kind of approaches deal with negative associations between items. Savasere et al. (1998) 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. (2004) presented an efficient method for mining positive and negative associations and proposed a pruning strategy and an interestingness measure. Finally, another kind of negative associations that has been studied concerns the mining of mutually exclusive items (Tzanis et al., 2006; Tzanis and Berberidis 2007).

The process of Knowledge Discovery from Databases (KDD Process) consists of a number of steps that can be grouped in three categories: pre-processing, data mining and post-processing (Figure 1). Although the core of the process is the data mining step, where a data mining algorithm (e.g. Apriori for mining of association rules) is applied, the pre-processing and post-processing phases are particularly important and contribute sensibly to the extraction of valuable knowledge. The pre-processing phase usually includes the selection of an appropriate portion of the data, the cleaning of the selected data and the transformation of the data. The post-processing phase deals with the management of the produced patterns and focuses on the evaluation and interpretation of data mining results.

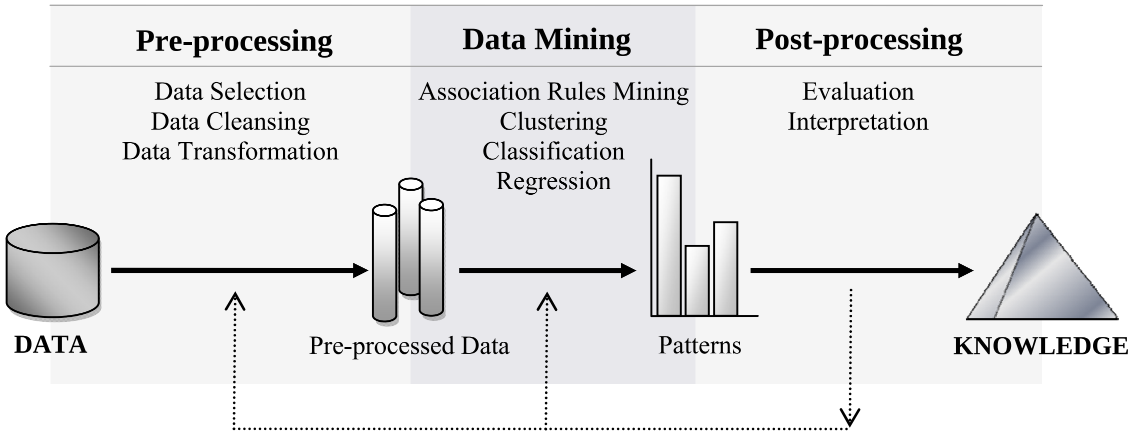

**Figure 1. The Knowledge Discovery in Databases (KDD) process.**

The central dogma of molecular biology, as coined by Francis Crick (1958), describes the flow of the biological information (Figure 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 describes also the basic kinds of data that can be produced by various biological experiments in relation to the three basic molecules of life.

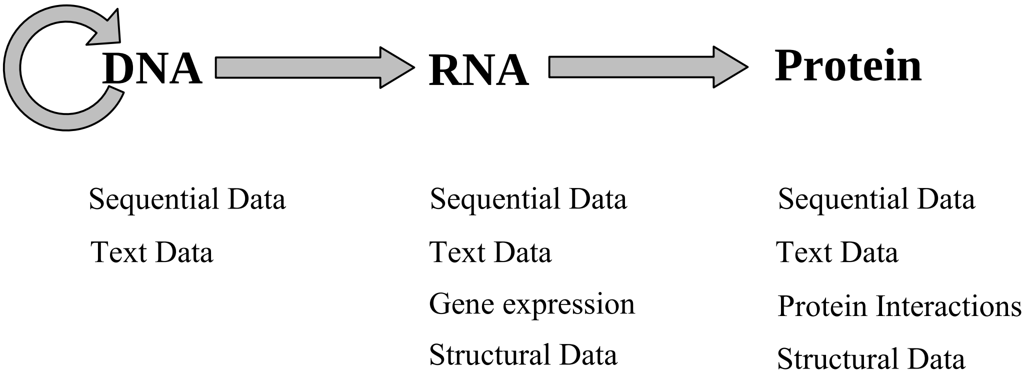

**Figure 2. The central dogma of molecular biology and the main kinds of biological data.**

It is important to mention that association rules 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 association rules mining algorithm successfully. Even though gene expression data are quite convenient for association analysis, association rules mining is also applied to other kinds of biological data. These applications will be presented in following sections.

In the next section, the problem of mining association rules will be defined. Then, some important mining algorithms, pre-processing and post-processing methods will be described. In the rest sections, the application of association rules mining algorithms to various kinds of biological data will be presented. Finally, the chapter will be concluded and summarized.

# Definition of the Association Rules Mining Problem

The original definition of the association rules mining problem has been given by Agrawal et al. in 1993. Let $I = \{i_1, i_2, ..., i_N\}$ be a finite set of binary attributes called *items* and $D = \{t_1, t_2, ..., t_N\}$ be a finite multiset of transactions, which is 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 reffered to as an *itemset*. If an itemset contains $k$ items, it is called a $k$-itemset. The number $k$ is called 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 *association rule* is an implication of the form $X \Rightarrow Y$ where $X \subset I$, $Y \subset I$ and $X \cap Y = \varnothing$. The itemset $X$ is called antecedent or left-hand-side (LHS) of the rule and the itemset $Y$ is called consequent or right-hand-side (RHS) of the rule.

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 dataset, 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:

$$support_D(X) = \frac{\left|\{T \in D \mid X \subseteq T\}\right|}{|D|}$$

The *confidence* of the rule *X*⇒*Y* is defined as the fraction of transactions in database that contains $X \cup Y$ 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:

$$confidence_D(X \Rightarrow Y) = \frac{supp_D(X \cup Y)}{supp_D(X)}$$

In order to make the concepts presented above more clear, a detailed example is going to be presented. Table 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 one represents a gene that is overexpressed.

**Table 1. Example of a binary gene expression matrix.**

| *Transaction ID* | *Gene1* | *Gene2* | *Gene3* | *Gene4* |
|---|---|---|---|---|
| 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 |

In this example the set of items is *I*={Gene1, Gene2, Gene3, Gene4}. In the database there are five transactions e.g. $t_3$={Gene1, Gene2}. An example of a rule for this database could be {Gene1, Gene2} ⇒ {Gene3}. The support for this rule is 1/5 (20%), because only one transaction ($t_5$) out of five contains Gene1, Gene2 and Gene3. The confidence of the rule is 1/2 (50%), because the support of itemset {Gene1,Gene2} (the antescedent of the rule) is 2/5 and the support of the itemset {Gene1, Gene2, Gene3} is 1/5.

The association rules 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.

# Algorithms for Mining Association Rules

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

## *Apriori*

Most of the algorithms for mining frequent itemsets are based on the principle of downward closure. This principle imposes that all non-empty subsets of a frequent itemset are also frequent. For example, if itemset {Gene1, Gene2, Gene5} is frequent according to a minimum support threshold, then itemsets {Gene1, Gene2}, {Gene1, Gene5}, {Gene2, Gene5}, {Gene1}, {Gene2}, and {Gene5} must also be frequent according to the same minimum support threshold. The most popular algorithm that employs this principle is Apriori (Agrawal and Srikant, 1994).

Apriori works by constructing candidate frequent itemsets and then checks which of them are indeed frequent (Table 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 itemsets mining in Apriori is a two step process:

3. The set of candidate frequent itemsets $C_i$ is constructed.

4. 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 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.

**Table 2. Pseudocode of the Apriori algorithm.**

**Input**: Database *D*, minimum support threshold (*min_sup*)

**Output**: All the frequent itemsets

```
$L_1$ ← {frequent items}
for ($k$ ← 2; $L_{k-1}$!=∅; $k$++) do
    $C_k$ ← {candidates from $L_{k-1}$} ($L_{k-1} \times L_{k-1}$ and downward closure)
    for each $t \in D$ do
        for each $c \in C_k$ do
            if ($c \subseteq t$)
                $c$.count++
    $L_k$ ← {$c \in C_k$ | $c$.count ≥ $min\_sup$}
return $\bigcup_k L_k$
```

## FP- Growth

The main drawbacks of the Apriori algorithm is that not only generates a huge number of candidate itemsets, but also scans the database several times. Both drawbacks are very costly. In 2000, Han et al. 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 FP-Tree (Frequent Pattern 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.

## *Eclat*

In 2000, Zaki 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.

Firstly, the algorithm builds the TID_set of all items in the transaction database. Likewise 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 3 presents the transaction database in which the first column represents the Transaction ID and the second the items included in each transaction.

**Table 3. Example of a transaction database.**

| *TID* | *List of Items 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 4 presents the transactional database of Table 3 in vertical data format. The first column represents the itemset and the second column present 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 5.

**Table 4. The transactional database of Table 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 5. The 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% |

Again by intersecting the TID_sets the frequent 3-itemsets (Table 6) in Vertical Data Format are generated. An optimization based in 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 concequent {I1,I4,I5} cannot be frequent.

**Table 6. The frequent 3-itemsets.**

| *Itemset* | *TID_set* | *Support* |
|---|---|---|
| {I1,I3,I5} | T2,T3 | 22% |
| {I3,I4,I5} | T2,T9 | 22% |

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

# Pre-processing and Post-processing

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

## *Discretization*

Many algorithms in Data Mining and the vast majority of the association rules 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* (Dougherty et al., 1995).

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 in which the instance belongs to. 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 in 1992. 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 following 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 by the following equation:

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

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 of 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 dataset, 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, that may skew the range.

Another basic and simple unsupervised strategy is the *Equal Frequency Intervals* 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 association rule mining (Vannucci and Colla, 2004). 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 (Becquet et al., 2002) were used for discretizing gene expression data that were produced using the serial analysis of gene expression (SAGE) method (Velculescu et al., 1995).

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 one and the replacement of all other values with the value of zero.

The second approach is the *mid-range-based cutoff*. This consist 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 one.

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 one. The rest ones are assigned with the value of zero.

In 2004, Vanucci and Colla proposed an unsupervised discretization method for dicretizing data for the purposes of association rules mining. The main thought is to preserve the original sample distribution. One idea was to use the *K*–means algorithm in order to generate a *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 results 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) (Kohonen, 1990) 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 association rule mining was proposed by Ludl and Widmer in 2000. The algorithm RUDE (Relative Unsupervised DiscrEtization) combines aspects of both supervised and unsupervised discretization. The method does not require a class attribute, hence it belongs to unsupervised methods. Furthermore, the split points for an attribute are constructed in dependence of the other attributes, hence it is called relative. The basic idea when discretizing a particular attribute (the target attribute) is to take in consideration information about the distribution of the values of the other attributes (source attributes).

## *Post-Processing of Association Rules*

As already mentioned in the introduction, the large number of association rules 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.

Post-processing usually consists of four steps, pruning, summarizing, grouping, and visualization (Baesens et al., 2000). 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 (using usually 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 post-processing procedure.

# Gene Expression Data Mining

*Gene expression* is the process by which the genetic information encoded in DNA is converted into a functional broduct 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 *microarray* (Schena et al., 1995). 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 (Aas, 2001). Another method for measuring gene expression is Serial Analysis of Gene Expression (SAGE), which allows the quantitative profiling of a large number of mRNA transcripts (Velculescu et al., 1995). 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 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 (Tuzhilin and

Adomavicius, 2002). The *M* 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 expression of the respective gene in the particular sample and the expression of the same gene in a control sample (microarrays).

**Table 7. A 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* | $a_{M1}$ | $a_{M2}$ | … | $a_{MN}$ |

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, i.e {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) is when the data are in the vertical format, i.e. {item: TID_set}. Table 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.

**Table 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 |

## *Mining in Horizontal Data Format*

As mentioned before, the most popular association rules 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 9 is a discretized gene expression matrix. A value of zero represents a gene that is underexpressed, whereas a value of one 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 9. Discretized gene expression matrix that is converted in transactional format.**

| | *G1* | *G2* | *G3* | *G4* |
|---|---|---|---|---|
| *C1* | 1 | 0 | 0 | 0 |
| *C2* | 1 | 1 | 0 | 0 |
| *C3* | 1 | 0 | 1 | 1 |
| *C4* | 1 | 1 | 1 | 1 |
| *C5* | 1 | 1 | 1 | 0 |
| *C6* | 0 | 0 | 1 | 1 |
| *C7* | 0 | 1 | 1 | 1 |
| *C8* | 0 | 1 | 0 | 1 |
| *C9* | 0 | 1 | 1 | 0 |
| *C10* | 0 | 1 | 1 | 1 |

$\Rightarrow$

| *TID* | *Items* |
|---|---|
| C1 | G1 |
| C2 | G1,G2 |
| C3 | G1, G3,G4 |
| C4 | G1G2,G3,G4 |
| C5 | G1,G2,G3 |
| C6 | G3,G4 |
| C7 | G2,G3,G4 |
| C8 | G2,G4 |
| C9 | G2,G3 |
| C10 | G2,G3,G4 |

In the first step of 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 9 is presented below:

- support({G1}) = 5/10 = 50% ≥ *min_sup*
- support({G2}) = 7/10 = 70% ≥ *min_sup*
- support({G3}) = 7/10 = 70% ≥ *min_sup*
- support({G4}) = 6/10 = 60% ≥ *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% ≥ *min_sup*
- support({G2, G4}) = 40% ≥ *min_sup*
- support({G3, G4}) = 50% ≥ *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 itemsets mining procedure terminates.

The next step of the Apriori algorithm include 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 ⇒ G3 = 5/7 = 71% < *min_conf* (discarded)
  - G3 ⇒ G2 = 5/7 = 71% < *min_conf* (discarded)

- {G3, G4}
  - G3 ⇒ G4 = 5/7 = 71% < *min_conf* (discarded)
  - G4 ⇒ G3 = 5/6 = 83% ≥ *min_conf* (accepted)

- {G2, G4}
  - G2 ⇒ G4 = 4/7 = 57% < *min_conf* (discarded)
  - G4 ⇒ G2 = 4/6 = 66% < *min_conf* (discarded).

Finally, only one rule is generated (G4 ⇒ 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 association rules. The association rule that was generated could mean that when gene4 (G4) is overexpressed then it is also likely, with a high possibility (83%), that gene3 (G3) is also overexpressed.

The majority of frequent pattern mining algorithms are based on the Apriori's 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., (2001) 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 CLOSET+ and CHARM. The definition of a closed pattern as is given by Han et al, (2007), states that *a* is a closed frequent pattern in a database *D* if *a* is frequent in *D* and there exists no proper super-pattern *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 is achieved better scalability and interpretability.

## *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 10000 – 100000 columns but only 100 – 1000 rows. As a result, it became obvious that the methods which use the horizontal data format for these dataset are not very suitable. Thus, many new algorithms where 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 dataset in vertical data format, was CARPENTER (Pan et al., 2003). CARPENTER discovers frequent closed patterns by performing depth-first row-wise enumeration instead of the conventional feature (column) enumeration. Furthermore, pruning techniques, are used in order to optimize the algorithm's efficiency.

Another approach (FARMER) has been proposed by Cong et al., in 2004. FARMER finds interesting rule groups and builds classifiers based on them. The concept of rule groups imply that rules supported by exactly the same set of rows are grouped together. FARMER is designed specifically to generate rules of the form *X*⇒*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 non-cancer.

Pan and his colleagues extended the CARPENTER algorithm in order to handle datasets with both large number of columns and rows. The algorithm, which is called COBBLER (Pan et al., 2004), switches dynamically between column and row enumeration based on 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. 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 prune the search space dramatically.

# Sequential Data Mining

The technological advances of the last decades have driven to 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 (Houle et al., 2000) and the assignment of functional information to known sequences is 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, requires the use of advanced techniques including data mining.

For example, eukaryotic genes consist of coding parts (exons) that are separated by intervening non-coding 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 (i.e. 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 to specific sequence signals could be associated to each other using association rules. 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 transformed the continuous-valued features to categorical ones.

An approach which used association analysis and combining gene expression and biological sequences proposed by Icev et al., (2003). 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 Association Rule Mining algorithm (DARM) is based on the Apriori algorithm. DARM has the ability to involve multiple motifs and to predict expressions in multiple cell types. Moreover, association rules 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 kind of frequent patterns that can be used for discriminated two classes are the emerging patterns. In a recent study (Tzanis et al., 2011), 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 (Dong and Li, 1999) are itemsets whose supports increase significantly from one dataset to another. Given two datasets $D_1$ 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 $D_2$):

$$gr_{1\to2}(X) = \begin{cases} 0, & \text{if } supp_1(X) = 0 \text{ and } supp_2(X) = 0 \\ \infty, & \text{if } supp_1(X) = 0 \text{ and } supp_2(X) > 0 \\ \dfrac{supp_2(X)}{supp_1(X)}, & \text{otherwise} \end{cases}$$

Given a minimum growth rate threshold $\rho$ > 1, an itemset $X$ is said to be *$\rho$-emerging pattern*, or simply *emerging pattern*, from $D_1$ to $D_2$, if $gr_{1\to2}(X) \geq \rho$. $D_1$ is called *background dataset* and $D_2$ is called *target dataset.*

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

$$strength_{1\to2}(X) = \begin{cases} supp_2(X), & \text{if } gr_{1\to2}(X) = \infty \\ supp_2(X)\dfrac{gr_{1\to2}(X)}{gr_{1\to2}(X)+1}, & \text{otherwise} \end{cases}$$

Emerging patterns in contrast to other patterns or models 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 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 minimum growth rate threshold is set high, then those emerging patterns with a low growth rate, but with a high support will be lost. There have been proposed some interestingness measures in order 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 patterns, called *Chi Emerging Patterns* (Fan, 2004), which are 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. $supp_2(X) \geq \sigma$, where $\sigma$ is a minimum support threshold.

2. $gr_{1\to2}(X) \geq \rho$, where $\rho$ is a minimum growth rate threshold.

3. $\forall Y \subset X, gr_{1\to 2}(Y) < gr_{1\to 2}(X)$

4. $|X| = 1 \vee |X| > 1 \wedge (\forall Y \subset X \wedge |Y| = |X| \text{-} 1 \wedge chi(X,Y) \geq \eta)$, where $\eta$ = 3.84 is a minimum chi value threshold and *chi*(*X*, *Y*) is computed using chi-squared test.

The first condition ensures that the mined emerging patterns will have at least a minimum coverage over the training dataset 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 in order 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 not any 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 datasets than the distributions of its immediate subsets.

# 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, 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 than 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 (Birzele, 2005): 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%. (Rost, 2001) The accuracy is defined as 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 that 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 (2006). In their approach they used the level-wise search strategy (Mannila and Toivonen, 1997), which is a string mining algorithm, in order to find frequent patterns in protein databases. From those frequent patterns were extracted features in order to be used on the prediction of the secondary structure using Support Vector Machines (SVMs).

Finally, in 2009 Beccerra et al. proposed an algorithm for biological sequence feature classification. The approach includes two main features. Firstly, the use of association analysis in order to extract interesting relationships hidden in biological datasets and secondly, 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 of all, the sequences are scanned as subsequensed of *N* symbols (*N*-Grams). The *N*-Grams represent patterns of variable size in the biological sequence and they are represented as binary vectors which allow the application of association analysis. The second step consists of finding frequent patterns in the sequences. Lastly, the third step consist of finding association rules. In this approach the Apriori algorithm was used for the association rule extraction process.

# 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 (Alberts et al., 2004, Stryer, 1988). Some of their numerous and versatile functions are referred below:

- 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 (Panchenco and Przytycka, 2008). 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 (2000) have reviewed and presented many of them including DIP, BIND, MIPS, PROTEOME, PROTONET, CURAGEN, and PIM.

Although a huge amount of information are 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 (Pandey et al., 2006). The noise can affect the efficacy of the algorithms at the task of function prediction between different datasets (Deng et al., 2003).

Recently, some data mining techniques have been proposed in order to determine the reliability of a given interaction. One of them (Pandey et al.,2007) uses the h-confidence measure (Xong et al., 2006) 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. Secondly, 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 in order to predict protein functions from a protein interaction network. In the context of graphs the frequent patterns are stood for the frequent sub graphs in a set of graphs (Kuramoch and Karipis 2004). 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., (2005) developed an algorithm, called CODENCE, in order 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 (Hartuv and Shamir 2000). It is worth mentioning that this method can integrate heterogeneous network data, such as protein interaction networks, genetic interaction networks and co-expression networks to reveal consistent biological signals.

Also, Xiong et al. (2005) 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 (Xiong et al., 2005).

- 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 $min(conf(\{p_1\} \Rightarrow \{p_2, p_3, \ldots p_m\}), conf(\{p_2\} \Rightarrow \{p_1, p_3, \ldots p_m\}), conf(\{p_3\} \Rightarrow \{p_1, p_2, \ldots p_m\}), \ldots conf(\{p_m\} \Rightarrow \{p_1, p_2, \ldots p_{m-1}\})$, where *conf* is the confidence of association rule.
- 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. (2004), who introduced the concept of differential association rule 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 association rule mining techniques.

Another perspective to the field of protein interaction networks is to find frequent sub-networks in a given network in a different manner than in the previously described approaches which focused on mining frequent sub-graphs in a set of networks. Such an algorithm is NeMoFinder, which was proposed by Chen et al. (2006) and was inspired by the Apriori algorithm.

# Text Data Mining

Text mining in molecular biology, defined as the automatic extraction of information about genes, proteins and their functional relationships from text documents (Krallinger and Valencia, 2005), 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 on-line 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 1st July 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 5,000 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 on-line resources. The researchers’ need to exploit this enormous volume of available information, along with the avail of high performance and efficiency data mining, natural language processing and information retrieval tools have 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 (Tzanis et al., 2009).

Several studies have categorized the tasks of BTM from different points of view. Cohen and Hersh (2005) 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 type of information.
- Synonym and abbreviation extraction. This task deals with the problem that many biological entities have multiple names so in biomedical literature 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 they are inferred by other explicit relationships.

An early work which used association rules for text mining was proposed by Hristovski et al., in 2001. 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 MeSH (Medical Subject Headings) terms that describe the content of the item in the database. The main idea was the use of association rules in order to find all concepts *Y* that are related to the starting concept *X*. Then, to find all the concepts *Z* that are related to the concept *Y*. The next step included the examination 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. The 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 association rules for biomedical text mining, was proposed by Berardi et al. (2005). The purpose of their method was to detect associations between concepts as indication of the existence of biomedical relation. This method also used Medline abstract and the MeSH taxonomy. The hierarchical nature of the MeSH taxonomy made it possible to mine multilevel association rules (generalized association rules) (Srikant and Agrawal, 1995). Generalized association rules include association rules of the form $X \Rightarrow Y$, where no item in $Y$ is an ancestor of any item in $X$ in a given taxonomy.

# Conclusion

Frequent patterns and association rules are useful data mining tools that have attracted the research interest since 1993 as a means of unsupervised, exploratory data analysis. Although they initially proposed as a market basket analysis tool, they almost immediately applied to other application domains and nowadays include a large number of applications. The community of biologists and bioinformaticians have used association rules 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 association rules more efficiently, as well as for mining new kinds of patterns and extending the concept of association rules will be proposed in the future. All these novel association rules mining tools will provide the means for more efficient and effective analyses of biological data. As a result, the research efforts of biologists are going to be enhanced by the gain of new biological insights and the rise of new biological questions that will guide to unexplored research directions.

# References

Aas, K. (2001). Microarray Data Mining: A Survey. NR Note, SAMBA, Norwegian Computing Center.

Agrawal, R. and Srikant, R. (1994). Fast Algorithms for Mining Association Rules in Large Databases. In Proceedings of the 20th International Conference on Very Large Databases, 478-499.

Agrawal, R., Imielinski, T., and Swami, A. (1993). Mining Association Rules Between Sets of Items in Large Databases. In Proceedings of the ACM SIGMOD Conference on Management of Data, 207-216.

Alberts Bruce, Bray Dennis, Hopkin Karen, Johnson Alexander, Lewis Julian, Raff Martin, Roberts Keith, Walter Peter, Essential Cell Biology, Second Edition, 2004

Aleksandar Icev, Carolina Ruiz, Elizabeth F. Ryder, "Distance-basedAssociation Rules Mining," BIOKDD03: Proc. 3rd ACM SIGKDD Workshop on Data Mining in Bioinformatics, p.34– 40,2003.

Baesens, B., Viaene, S., and Vanthienen, J. (2000). Postprocessing of Association Rules. In Proceedings of the Wrkshop 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, USA.

Becerra, D. Sandoval, A. Restrepo-Montoya, D., Nino, L.F., An association rule based approach for biological sequence feature classification CEC '09. IEEE Congress on Evolutionary Computation, 2009.

Becquet, C. Blachon, S., Jeudy, B., Boulicaut, J.-F., and Gandrillon, O. (2002). Strong-Association-Rule Mining for Large-Scale Gene-Expression Data Analysis: A Case Study on Human SAGE Data. Genome Biology, 3(12): research0067.

Berardi, M., Lapi, M., Leo, P., and Loglisci, C. (2005). Mining generalized association rules on biomedical literature, Proc. of the 18th International Conference on Innovations in Applied Artificial Intelligence, LNCS 3533, pp. 500-509.

Besemann C, Denton A, Yekkirala A**:** Differential association rule mining for the study of protein-protein interaction networks**.** BIOKDD04: 4th Workshop on Data Mining in Bioinformatics (with SIGKDD Conference) 2004, 72-80.

Birzele F and Kramer S. A new representation for protein secondary structure prediction based on frequent patterns. Bioinformatics, 22:2628–2634, November 2006.

Birzele F. Data mining for protein secondary structure prediction. Master's thesis, Technische Universitδt Mónchen, January 2005.

Chen, J., Hsu, W., Lee, M. L., AND Ng, See-Kiong. 2006. NeMoFinder: dissecting genome-wide protein-protein interactions with meso-scale network motifs. In KDD '06:

Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining. 106–115.

Chen, X. and Petrounias, I. (2000). Discovering Temporal Association Rules: Algorithms, Language and System. In Proceedings of the 16th International Conference on Data Engineering.

Cohen, A.M. and Hersh, W.R. (2005). A survey of current work in biomedical text mining. Briefings in Bioinformatics, 6(1), 57–71.

Cong G, Tung A. K. H., Xu X, Pan F, and Yang J. Farmer: Finding interesting rule groups in microarray datasets. In 23rd ACM International Conference on Management of Data, 2004.

Crick, F.H.C. (1958). On protein synthesis. Symposium of the Society for Experimental Biology XII, 139-163.

Deng, M., Sun, F., Chen, T. 2003. Assessment of the reliability of protein–protein interactions andprotein function prediction. In Pac Symp Biocomputing. 140–151.

Dong G. and Li J, “Efficient mining of emerging patterns: Discovering trends and differences”. In Proceedings of ACM-SIGKDD’99, 1999, pp. 43–52.

Dougherty J., Kohavi R, Sahami M, Supervised and unsupervised discretization of continuous features In A. Prieditis and S. Russell, eds., Machine learning Proc. 12th Int. Conf., 1995, Morgan Kaufmann Publishers, San Francisco, CA.

Fan H. Efficient Mining of Interesting Emerging Patterns and Their Effective Use in Classification, PhD Thesis, University of Melbourne, Australia, 2004.

Fayyad, U. and Irani, K. 1992. On the handling of continuous-valued attributes in decision tree generation. Machine Learning, 8:87–102.

Gaurav Pandey, Vipin Kumar and Michael Steinbach, "Computational Approaches for Protein Function Prediction: A Survey", TR 06-028, Department of Computer Science and Engineering, University of Minnesota, Twin Cities

Han J, Cheng H, Xin D, Yan X (2007) Frequent pattern mining: current status and future directions. Data Mining Knowl Discov 15(1):55–86

Han, J., and Fu, Y. (1995). Discovery of Multiple-Level Association Rules from Large Databases. In Proceedings of the 21st International Conference on Very Large Databases. 420-431.

Han, J., Pei, J., Yin, Y. (2000). Mining Frequent Patterns Without Candidate Generation. In Proceedings of the ACM SIGMOD International Conference on Management of Data, Dallas, Texas, USA, 1-12.

Hartuv, E., Shamir, R. 2000. A clustering algorithm based on graph connectivity. Inf. Process. Lett. 76, 4-6, 175–181.

Houle, J.L., Cadigan, W., Henry, S., Pinnamaneni, A. and Lundahl, S. (2004. March 10). Database Mining in the Human Genome Initiative. Whitepaper, Bio-databases.com, Amita Corporation. Available: http://www.biodatabases.com/whitepaper01.html (Last Access: July 15, 2011).

Hristovski d., Stare J., Peterlin B., and Dzeroski S.: "Supporting discovery in medicine by association rule mining in Medline and UMLS", Proceedings of MedInfo Conference, London, England, September 2-5, 2001, 10(2), pp 1344-1348.

Kohonen T., The self-organizing map Proc. IEEE, Vol. 78, No. 9, pp. 1464-1480, 1990.

Koperski, K. and Han, J. (1995). Discovery of Spatial Association Rules in Geographic Information Databases. In Proceedings of the 4th International Symposium on Large Spatial Databases. 47-66.

Kotala P, Perera A, Zhou JK,. 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, USA, 2001; 15–16.

Krallinger, M. and Valencia, A. (2005). Text-mining and information-retrieval services for molecular biology. Genome Biology, 6(224).

Kuramochi, M., Karypis, G. 2004. An efficient algorithm for discovering frequent subgraphs. IEEETrans. Knowl. Data Eng. 16, 9, 1038–1051.

Liu H, Han J, Xin D, and Shao Z. "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.

Ludl, M.-C., and Widmer, G. Relative unsupervised discretization for association rule mining. In Proceedings of the Fourth European Conference on Principles and Practice of Knowledge Discovery in Databases (2000).

Mannila, H., Toivonen, H., and Verkamo, A.I. (1994). Efficient Algorithms for Discovering Association Rules. In Proceedings of AAAI Workshop on Knowledge Discovery in Databases. 181-192.

Mannila,H. and Toivonen,H. (1997) Levelwise search and borders of theories in knowledge discovery. Data Mining and Knowledge Discovery, 3, 241–258.

Pan F, Cong G, Tung AK, et al. Carpenter: finding closed patterns in long biological datasets. In: Proceedings of the the Ninth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. Washington, DC, USA: ACM Press, 2003;637–42.

Pan F, Tung A, Cong G, and Xu X. COBBLER: combining column and row enumeration for closed pattern discovery. Scientific and Statistical Database Management, 2004. Proceedings. 16th International Conference on, pages 21–30, 2004.

Panchenko, A. and Przytycka, T. (2008). Protein – Protein Interactions and Networks. Identification, Computer Analysis and Prediction, Springer.

Pandey, G., Steinbach, M., Gupta, R., Garg, T., Kumar, V. 2007. Association analysis-based transformations for protein interaction networks: a function prediction case study. In KDD '07: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining. 540–549.

Rost B. Review: protein secondary structure prediction continue torise. J Struct Biol 2001;134:204–218.

Savasere, A., Omiecinski, E, and Navathe, S.B. (1998). Mining for Strong Negative Associations in a Large Database of Customer Transactions. In Proceedings of the 14th International Conference on Data Engineering. 494-502.

Schena, M., Shalon, D., Davis, R.W., Brown, P.O. (1995). Quantitative monitoring of gene expression patterns with a complementary DNA microarray. Science 270 (5235): 467–470.

Srikant, R., and Agrawal, R. (1995). Mining Generalized Association Rules. In Proceedings of the 21st VLDB Conference, 407-419.

Stryer Lubert, Biochemistry, 3$^{rd}$ Edition W.H. Freeman and Company, 1988.

Tan P.-N.,Steinbach M, and Kumar V. Introduction to Data Mining. Addison- Wesley, 2006.

Thomas, S. and Sarawagi, S. (1998). Mining Generalized Association Rules and Sequential Patterns Using SQL Queries. In Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining. 344-348.

Tung, A. K. H., Lu, H., Han, J., and Feng, L. (2003). Efficient Mining of Intertransaction Association Rules. IEEE Transactions On Knowledge And Data Engineering. 15(1), 43-56.

Tuzhilin and G. Adomavicius, "Handling very large numbers of association rules in the analysis of microarray data," in Proc. 8th KDD, 2002.

Tzanis, G. and Berberidis, C. (2007). Mining for Mutually Exclusive Items in Transaction Databases. International Journal of Data Warehousing and Mining, 3(3), Idea Group Publishing.

Tzanis, G. Berberidis, C., and Vlahavas, I. (2006). 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.

Tzanis, G., Berberidis, C., Vlahavas, I. (2009). Machine Learning and Data Mining in Bioinformatics, Handbook of Research on Innovations in Database Technologies and Applications: Current and Future Trends, Laura C. Rivero, Jorge H. Doorn and Viviana E. Ferraggine (Eds.), IGI Global.

Tzanis, G., Kavakiotis, I,, Vlahavas, I. (2011), PolyA-iEP: A Data Mining Method for the Effective Prediction of Polyadenylation Sites, Expert Systems with Applications, Elsevier, 38(10).

Vannucci M, Colla V, Meaningful disretization 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.

Velculescu, V.E., Zhang, L. Vogelstein, B., and Kinzler, K.W. (1995). Serial Analysis of Gene Expression, Science, 270 (5235), 484-487.

Wang J, Han J, and Pei J. Closet+: Searching for the best strategies for mining frequent closed itemsets. In Proceedings of ACM SIGKDD'03, Washington, DC, 2003.

Wu, X., Zhang, C., and Zhang, S. (2004). Efficient Mining of both Positive and Negative Association Rules. ACM Transactions on Information Systems. 22(3), 381-405.

Xenarios, I. and Eisenberg, D. 2001. Protein interaction databases. Curr.Opin.Biotechnol. 12: 334–339.

Xiong, H., He, X., Ding, C., Zhang, Y., Kumar, V., Holbrook, S. R. 2005. Identification of functional modules in protein complexes via hyperclique pattern discovery. In Proc. Pacific Symposium on Biocomputing (PSB). 221–232.

Xiong, H., Tan, P.-N., Kumar, V. 2006. Hyperclique pattern discovery. DataMin. Knowl. Discov. 13, 2,219–242.

Xiong, H., Tan, P.-N., Kumar, V. 2003. Mining strong affinity association patterns in data sets with skewed support distribution. In ICDM '03: Proceedings of the Third IEEE International Conference on Data Mining. 387–394.

Zaki M. and Hsiao C.. Charm: An efficient algorithm for closed itemset mining. In Proceedings of SIAM'02, Arlington, Apr. 2002.

Zaki MJ (2000) Scalable algorithms for association mining. IEEETransKnowl Data Eng 12:372–390

Zaki, M.J. and Hsiao, C.J. 2002. CHARM: An efficient algorithm for closed itemset mining. In Proc. 2002 SIAMInt. Conf. Data Mining, Arlington, VA, pp. 457–473.