Unifying Framework for Rule Semantics: Application to Gene Expression Data

Marie Agier*
DIAGNOGENE, Aurillac, FRANCE
LIMOS, UMR 6158 CNRS, Univ. Clermont-Ferrand II, FRANCE
agier@isima.fr

Jean-Marc Petit
LIRIS, UMR 5205 CNRS, INSA Lyon, FRANCE
jean-marc.petit@insa-lyon.fr

Einoshin Suzuki
Graduate School of Information Science and Electrical Engineering
Kyushu University, JAPAN
suzuki@i.kyushu-u.ac.jp

Abstract. The notion of rules is very popular and appears in different flavors, for example as association rules in data mining or as functional dependencies in databases. Their syntax is the same but their semantics widely differs. In the context of gene expression data mining, we introduce three typical examples of rule semantics and for each one, we point out that Armstrong’s axioms are sound and complete. In this setting, we propose a unifying framework in which any “well-formed” semantics for rules may be integrated. We do not focus on the underlying data mining problems posed by the discovery of rules, rather we prefer to discuss the expressiveness of our contribution in a particular application domain: the understanding of gene regulatory networks from gene expression data. The key idea is that biologists have the opportunity to choose - among some predefined semantics - or to define the meaning of their rules which best fits into their requirements. Our proposition has been implemented and integrated into an existing open-source system named MeV of the TIGR environment devoted to microarray data interpretation. An application has been performed on expression profiles of a sub-sample of genes from breast cancer tumors.

Keywords: Rules, implications, Armstrong’s axiom system, data mining.

* Address for correspondence: agier@isima.fr

Address for correspondence: LIMOS, UMR 6158 CNRS, Univ. Clermont-Ferrand II, FRANCE
1. Introduction

Microarray technology provides biologists with the ability to measure the expression levels of thousands of genes in a single experience. It is believed that genes of similar function yield similar expression patterns in microarray experiences [30]. As data from such experiences accumulates, it is essential to have accurate means for assigning functions to genes. Also, the interpretation of large-scale gene expression data provides opportunities for developing novel mining methods for selecting for example good drug candidates (all genes are potentially drug targets) from among tens of thousands of expression patterns [15, 29].

However, one real challenge lies in inferring important functional relationships from gene expression data. Beyond cluster analysis [13], a more ambitious purpose of genetic inference is to find out the underlying regulatory interactions from the expression data, using efficient inference procedures.

Rules concerning genes are a promising knowledge representation to reveal regulatory interactions from gene expression data. The conjecture that association rules could be a model for the discovery of gene regulatory networks has been partially validated in [4, 7, 5, 21, 11, 1, 10]. Nevertheless, we believe that many different kinds of rules could be useful to cope with different biological objectives and the restricted setting of association rules could not be enough.

Clearly, the notion of rules is very popular and appears in different flavors, the two more famous examples being association rules in data mining and functional dependencies in databases. A simple remark can be done on these rules: their syntax is the same but their semantics i.e. their meaning widely differs.

In this paper, we propose a unifying framework in which any “well-formed” semantics for rules may be integrated. The key features of our approach are the following:

1. Given a dataset, defining a semantics in collaboration with domain experts (e.g. biologists and physicians).
2. Verifying if the semantics fits into our framework, i.e. if Armstrong’s axiom system is sound and complete for this semantics [3].
3. For a given semantics, discovering rules satisfied in the dataset: More precisely a cover for exact rules [24, 12] and a cover for approximate rules [14, 22] have to be generated.
4. Computing in a post-processing step, several quality measures for the obtained rules.

Note that we do not focus on the underlying data mining problems posed by the discovery of rules, rather we prefer to emphasize the expressiveness of our contribution in a particular application domain: the understanding of gene regulatory networks from gene expression data.

We introduce in this paper three semantics devoted to gene expression data. The first one generates rules between genes according to their expression levels, i.e. under- or over-expressed genes. The second semantics analyzes the variations of gene expression levels and finally, the third semantics studies the evolution of gene expression levels between two consecutive samples, being understood that an order has to exist among samples.

Our proposition has been implemented in a friendly graphical user interface to make it useful by biologists. We chose to integrate it as a module into a microarray data analysis open-source software:
MeV. This tool is a part of an application suite, called TM4, developed by The Institute for Genomic Research (TIGR) [28].

Moreover, we have integrated five existing quality measures (support, confidence, lift, leverage and conviction). These measures have been defined for association rules but can be extended to these three semantics without any problem. An application has been performed on expression profiles of a sub-sample of genes from breast cancer tumors, some experimental results are also given.

Paper organization In Section 2, the framework of our approach is given. In Section 3, three rule semantics are detailed and their compliances with the framework are shown in Section 4. Implementation details are given in Section 5 and experiments are given in Section 6. Finally, we conclude and give some perspectives in Section 7.

2. Framework of our approach

2.1. Preliminaries

In this paper, we consider rules to be defined on tabular datasets (or relations) over a set \( G \) of distinguished attributes (or columns). In the context of gene expression data, attributes correspond to genes and tuples to samples.

Let \( G \) be a finite set of genes. Each gene \( g \in G \) takes its possible values in \( \mathbb{R} \), the real numbers. A tuple over \( G \) is a mapping \( t : G \rightarrow \mathbb{R}^n \). A relation is a set of tuples. We say that \( r \) is a relation over \( G \), i.e. a gene expression dataset. Let \( g \in G \) be a gene and \( t \) be a tuple; we denote by \( t[g] \) the restriction of \( t \) to \( g \).

The syntax of a rule over \( G \) is an expression \( X \rightarrow Y \) i.e. “\( X \) implies \( Y \)" where \( X, Y \subseteq G \).

The semantics of a rule \( X \rightarrow Y \) over \( G \) is the meaning, the sense one wants to give to this rule: Given a relation \( r \), a rule \( X \rightarrow Y \) is said to be satisfied in \( r \) with the semantics \( s \), denoted by \( r \models_s X \rightarrow Y \), if the semantics \( s \) is true (or valid) in \( r \).

The reader may refer to [25, 16, 9] for different points of view concerning the notion of rules which are referred to as implications in discrete mathematics field or as functional dependencies in database field.

Example 2.1. Let us consider a running example made of a set of 6 tuples (or samples) \((t_1, t_2, t_3, t_4, t_5\) and \( t_6 \)) over a set of 8 genes \((g_1, g_2, g_3, g_4, g_5, g_6, g_7\) and \( g_8 \)) as depicted in Table 1.

Throughout this paper, we will illustrate our approach with this example.

2.2. Justification of our framework

Our approach is based on the notion of rule, also called implication. Let us recall that a rule is an expression of the shape \( X \rightarrow Y \) i.e. “\( X \) implies \( Y \)” and the semantics of the rule is the signification one wants
to give to this implication. For example, association rules in data mining or functional dependencies in databases are two types of semantics.

In this paper, we focus on special kinds of rules, which exhibit nice properties, i.e. Armstrong’s axiom system is sound and complete for the considered semantics. Such a semantics for rules is called “well-formed” in the sequel. We have chosen to focus on Armstrong’s axioms since they obviously apply for functional dependencies but also for implications defined on a closure system [16] and thus turn out to have many practical applications (see examples given in [16]).

Practical interests of a well-formed semantics are twofold:

- Firstly, we can perform some kind of reasoning on rules from the Armstrong’s axioms: From a set of rules $F$, it is possible to know if a rule is implied by this set of rules. This problem is known as the implication problem and can be resolved in linear time [6]. Thus, if there is a relation $r$ which satisfies $F$ then all rules that can be deduced from $F$ thanks to the Armstrong’s axioms will be also satisfied in this relation.

- We can also work on “small” covers of rules [23, 20] and propose a discovery process specific to the considered cover, but applicable to all well-formed semantics. It is also possible to propose covers for non-satisfied rules [17].

The theoretical framework that we propose to use for the generation of rules defined with well-formed semantics, comes from the inference of functional dependencies [24, 12]. Basically, since by definition the Armstrong’s axioms apply for any well-formed semantics, the augmentation axiom implies a monotone property: given an attribute $A$, $X \rightarrow A \Rightarrow \forall Y \supseteq X, Y \rightarrow A$.

That is to say that the predicate “$X$ implies $A$” is monotone with respect to set inclusion, thus the predicate “$X$ does not imply $A$” is anti-monotone. So well known characterization may be used to produce the rules [26].

In other words, the largest left-hand sides not implying $A$ constitute the positive border of the predicate “$X$ does not imply $A$” and the smallest left-hand sides implying $A$ constitute its negative border. Consequently, this negative border gives a subset of the canonical cover (i.e. rules with minimal left-hand sides and $A$ as right-hand side) while the positive border gives a subset of the Gottlob and Libkin cover [17] (i.e. rules with maximal left-hand sides and $A$ as right-hand side).

Details on the generation of rules are out of the scope of this paper, interested readers are referred to [24, 12, 26].
Moreover, an important key point of our approach is to take into account characteristics of gene expression data. Indeed, from the microarray analysis domain, two underlying “constraints” have to be understood: firstly, the number of samples is small (a few hundreds at most) whereas the number of genes is large (several thousands). Such a constraint differs widely from those usually held in databases or data mining where the number of tuples can be huge whereas the number of attributes (i.e. genes in the context of this paper) remains rather small. That is why for example our approach does not take into account a minimum support threshold unlike association rules. Statistical measures are computed \textit{a posteriori} on the discovered rules.

Secondly, data pre-processing steps on gene expression data are not fully understood yet and therefore, we have to take into account noisy data. Thus microarray technology delivers numerical values with a relatively small confidence on these values, biologists have to interpret the data, for example as levels of expression, which implies a discretization step. In this setting, we propose to deal with noise in data not as an explicit pre-processing step but implicitly within the semantics of the rules.

3. Example of semantics for rules

Our approach consists to interact with biologists in order to establish a rule semantics which fits into their objectives and adapted to their data. In the sequel, we present three semantics for gene expression data.

The first one, called $s_1$ (Gene expression levels), generates rules between genes according to their expression levels. The second semantics, called $s_2$ (Gene expression level variations), generates rules between genes according to the variation of their expression levels. And finally, the third semantics, called $s_3$ (Gene expression level evolution), generates rules between genes according to the evolution of their expression levels between two consecutive samples.

3.1. Semantics 1: Gene expression levels

This first semantics consists in studying the levels of expression of genes. We shall call this semantics $s_1$ in the sequel. Let us note that the definition of this semantics is close to the definition of association rules but is applied to quantitative data so that it does not require a discretization phase and no minimum support threshold has to be set up.

**Definition 3.1.** (Semantics 1: Gene expression levels) Let $X, Y \subseteq G$, be two sets of genes and $r$ a relation over $G$. A rule $X \rightarrow Y$ is satisfied in $r$ with the semantics $s_1$ defined with two thresholds $\varepsilon_1$ and $\varepsilon_2$, denoted by $r \models_{s_1} X \rightarrow Y$, if and only if $\forall t \in r$, if $\forall g \in X, \varepsilon_1 \leq t[g] \leq \varepsilon_2$ then $\forall g \in Y, \varepsilon_1 \leq t[g] \leq \varepsilon_2$.

In other words, thanks to the two thresholds $\varepsilon_1$ and $\varepsilon_2$, the biologists have the opportunity to define rules between over-expressed (or under-expressed) genes.

**Example 3.1.** Let us consider that biologists are interested in studying over-expressed genes. To do so, assume that thresholds are set as follows: $\varepsilon_1 = 1.0$ and $\varepsilon_2 = 2.0$, since they consider that a gene is over-expressed if its expression level is between 1.0 and 2.0.
In that case, the rule \( g_1 \rightarrow g_3 \) is satisfied in the relation \( r \) given in Table 1 (on the other hand the rule \( g_3 \rightarrow g_1 \) is not satisfied because of the tuple \( t_4 \)). The expression levels of the genes \( g_1 \) and \( g_3 \) are given in the figure 1.

![Figure 1. Expression levels of the genes \( g_1 \) and \( g_3 \)](image)

The rule \( g_1 \rightarrow g_3 \) is interpreted in the following way: for any sample, if the gene \( g_1 \) is over-expressed then the gene \( g_3 \) is also over-expressed.

### 3.2. Semantics 2: Gene expression level variations

In many cases, it does make sense to compare samples in a pairwise fashion to find some regularities between samples. Such kind of reasoning is well known in the database community through the notion of functional dependencies. However, in our context, the FD satisfaction - its meaning - has to be relaxed to take into account noise in gene expression data. Since a crisp FD \( X \rightarrow Y \) can be rephrased as “equal X-values correspond to equal Y-values”, we would like to obtain something like “close X-values correspond to close Y-values”. Thus, instead of requiring strong equality between attribute values, we admit an error less or equal to the absolute value of the difference (obviously, other norms should have been taken). This leads to the following semantics:

**Definition 3.2.** (Semantics 2: Gene expression level variations) Let \( X, Y \subseteq G \) be two sets of genes and \( r \) a relation over \( G \). A rule \( X \rightarrow Y \) is satisfied in \( r \) with the semantics \( s_2 \) defined with two thresholds \( \varepsilon_1 \) and \( \varepsilon_2 \), denoted by \( r \models_{s_2} X \rightarrow Y \), if and only if \( \forall t_1, t_2 \in r \), if \( \forall g \in X, \varepsilon_1 \leq |t_1[g] - t_2[g]| \leq \varepsilon_2 \) then \( \forall g \in Y, \varepsilon_1 \leq |t_1[g] - t_2[g]| \leq \varepsilon_2 \).

Classical satisfaction of functional dependencies is achieved when \( \varepsilon_1 = \varepsilon_2 = 0 \).

Given any couple of samples, \( X \rightarrow Y \) means that if the variation of the expression levels of each gene \( g \) of \( X \) is between \( \varepsilon_1 \) and \( \varepsilon_2 \) then the variation of the expression levels of each gene \( g \) of \( Y \) is also between \( \varepsilon_1 \) and \( \varepsilon_2 \).
Example 3.2. Let us suppose that biologists are interested in small variations of expression levels between samples. Thresholds should be defined as follows: $\varepsilon_1 = 0.0$ and $\varepsilon_2 = 0.2$. The hypothesis is that a gene does not vary between any couple of samples if the difference of the expression levels is between 0.0 and 0.2.

The expression levels of the genes $g_6$ and $g_7$ are plotted in Figure 2. In that case, the rule $g_6 \rightarrow g_7$ is satisfied in the relation $r$ (on the other hand the rule $g_7 \rightarrow g_6$ is not satisfied because of the variation between the samples $t_3$ and $t_4$).

![Figure 2. Expression levels of the genes $g_6$ and $g_7$](image)

The rule $g_6 \rightarrow g_7$ is interpreted in the following way: Given any couple of samples, if the expression level of the gene $g_6$ does not vary then the expression level of the gene $g_7$ does not vary neither.

3.3. Semantics 3: Gene expression level evolution

This semantics generates rules between genes according to the evolution of their expression levels between two consecutive moments. In that case, an order on the samples is required in order to compare the expression level of a gene at $i+1$ with respect to its expression level at $i$.

Definition 3.3. (Semantics 3: Gene expression level evolution) Let $X, Y$ be two sets of genes, $\varepsilon_1$ and $\varepsilon_2$ be two thresholds and $r$ a relation.

A rule $X \rightarrow Y$ is satisfied in $r$ with the third semantics, denoted by $r \models_{s_3} X \rightarrow Y$, if and only if $\forall t_i, t_{i+1} \in r$, if $\forall g \in X, \varepsilon_1 \leq t_{i+1}[g] - t_i[g] \leq \varepsilon_2$ then $\forall g \in Y, \varepsilon_1 \leq t_{i+1}[g] - t_i[g] \leq \varepsilon_2$.

We notice that we got rid of the absolute value because the order of samples is significant.

Example 3.3. The samples must be ordered. For example, the sample $t_1$ can represent the state of a cell at the moment $i_0$, then after injection of a drug, the cell is analyzed six hours later to give sample $t_2$ etc.
until sample \( t_6 \) 30 hours later. This process allows to show the impact of a drug on gene expression of the cell in the time.

Suppose that biologists are interested in genes whose expression levels grow in the time with the following thresholds: \( \varepsilon_1 = 1.0 \) and \( \varepsilon_2 = 4.0 \). That is, the expression of a gene grows between \( i \) and \( i+1 \) if its expression level at \( i+1 \) is greater or equal to more than 1.0 point to its expression level at the moment \( i \).

In that case, the rule \( g_2 \to g_4 \) is satisfied in the relation \( r \) (on the other hand the rule \( g_4 \to g_2 \) is contradicted by the evolution between \( t_2 \) and \( t_3 \)). The expression levels of the genes \( g_2 \) and \( g_4 \) are given in the figure 3.

![Figure 3. Expression levels of the genes \( g_2 \) and \( g_4 \)](image)

The rule \( g_2 \to g_4 \) is interpreted in the following way: between two consecutive samples \( i \) and \( i+1 \), if the expression level of the gene \( g_2 \) grows then the expression level of the gene \( g_4 \) grows.

### 4. Well-formed semantics

The second step of the process is to verify the well-formedness of the semantics defined by domain experts, i.e. verify that this semantics can be used within our framework. This step is very important since many semantics could be defined, some of them verifying these requirements, others not (see Theorem 4.2).

To do that, we introduce the notion of well-formed semantics:

**Definition 4.1.** A semantics \( s \) is well-formed if Armstrong’s axiom system is sound and complete for \( s \).

Let us recall the Armstrong’s axiom system for a set of rules \( F \) defined over a set of attributes (i.e. genes in our context) \( G \):
1. (reflexivity) if $X \subseteq Y \subseteq G$ then $F \vdash Y \rightarrow X$

2. (augmentation) if $F \vdash X \rightarrow Y$ and $W \subseteq G$, then $F \vdash XW \rightarrow YW$

3. (transitivity) if $F \vdash X \rightarrow Y$ and $F \vdash Y \rightarrow Z$ then $F \vdash X \rightarrow Z$

The notation $F \vdash X \rightarrow Y$ means that a proof of $X \rightarrow Y$ can be obtained using Armstrong’s axiom system from $F$. Moreover, given a semantics $s$, the notation $F \models_s X \rightarrow Y$ means that for all relations $r$ over $G$, if $r \models_s F$ then $r \models_s X \rightarrow Y$.

As expected, the three semantics previously introduced verify these requirements.

**Theorem 4.1.** The semantics $s_1, s_2, s_3$ are well-formed.

We show the result only for $s_1$, the proof for $s_2$ and $s_3$ being quite similar. We need to show that Armstrong’s axiom system is sound and complete for $s_1$.

**Lemma 4.1.** Armstrong’s axiom system is sound for $s_1$.

**Proof:**
Let $F$ be a set of rules. We need to show that if $F \vdash X \rightarrow Y$ then $F \models_{s_1} X \rightarrow Y$.

Let $r$ be a relation over a set of genes $G$.

1. (reflexivity) evident.

2. (augmentation) Let $t \in r$ such that $\forall g \in X \cup W, \varepsilon_1 \leq t[g] \leq \varepsilon_2$. We need to show that $\forall g \in Y \cup W, \varepsilon_1 \leq t[g] \leq \varepsilon_2$, which implies that $r \models_{s_1} XW \rightarrow YW$. By assumption $F \vdash X \rightarrow Y$, then we have $\forall g \in Y, \varepsilon_1 \leq t[g] \leq \varepsilon_2$. The result follows.

3. (transitivity) Let $t \in r$ such that $\forall g \in X, \varepsilon_1 \leq t[g] \leq \varepsilon_2$. We need to show that $\forall g \in Z, \varepsilon_1 \leq t[g] \leq \varepsilon_2$, which implies that $r \models_{s_1} X \rightarrow Z$. By assumption, $F \vdash X \rightarrow Y$ and $F \vdash Y \rightarrow Z$, then $\forall g \in Y, \varepsilon_1 \leq t[g] \leq \varepsilon_2$ and $\forall g \in Z, \varepsilon_1 \leq t[g] \leq \varepsilon_2$ respectively. The result follows.

**Lemma 4.2.** Armstrong’s axiom system is complete for $s_1$.

**Proof:**
We need to show that if $F \models_{s_1} X \rightarrow Y$ then $F \vdash X \rightarrow Y$ or equivalently, if $F \not\vdash X \rightarrow Y$ then $F \not\models_{s_1} X \rightarrow Y$. As a consequence, assuming that $F \not\vdash X \rightarrow Y$, it is enough to give a counter-example relation $r$ such that $r \models_{s_1} F$ but $r \not\models_{s_1} X \rightarrow Y$.

Let $r$ over $G$ be the relation shown in Table 2, with $\varepsilon_1 = 1.0$ and $\varepsilon_2 = 2.0$.

<table>
<thead>
<tr>
<th>$X^+$</th>
<th>$G - X^+$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.5 ... 1.5</td>
<td>0.5 ... 0.5</td>
</tr>
</tbody>
</table>

Table 2. Counter-example
Firstly, we have to show that \( r \models_{s_1} F \). We suppose the contrary that \( r \not\models_{s_1} F \) and thus, \( \exists V \rightarrow W \in F \) such that \( r \not\models_{s_1} V \rightarrow W \). It follows by the construction of \( r \) that \( V \subseteq X^+ \) and \( \exists A \in W \) such that \( A \in G - X^+ \). Since \( V \in X^+ \), we have \( F \vdash X \rightarrow V \) and since \( F \vdash V \rightarrow A \), we have \( F \vdash X \rightarrow A \). Thus, by the transitivity rule, \( F \vdash X \rightarrow A \) and thus \( A \in X^+ \). This leads to a contradiction since \( A \in W \), and thus \( r \models_{s_1} F \).

Secondly, we have to show that \( r \not\models_{s_1} X \rightarrow Y \). We suppose the contrary that \( r \models_{s_1} X \rightarrow Y \). It follows by the construction of \( r \) that \( Y \subseteq X^+ \) and thus \( F \vdash X \rightarrow Y \). It leads to a contradiction since \( F \not\vdash X \rightarrow Y \) was assumed, and thus \( r \not\models_{s_1} X \rightarrow Y \).

As an example of semantics which does not fit into our framework, let us consider the following semantics, noted \( s'_2 \), which extends the semantics \( s_2 \) with an additional constraint:

**Definition 4.2.** Let \( X, Y \subseteq G \), be two sets of genes and \( r \) a relation over \( G \). A rule \( X \rightarrow Y \) is satisfied in \( r \) with the semantics \( s'_2 \) defined with two thresholds \( \varepsilon_1 \) and \( \varepsilon_2 \), denoted by \( r \models_{s'_2} X \rightarrow Y \), if and only if \( \forall t_1, t_2 \in r, \forall g \in X, \varepsilon_1 \leq |t_1[g] - t_2[g]| \leq \varepsilon_2 \) then \( \forall g \in Y, \varepsilon_1 \leq |t_1[g] - t_2[g]| \leq \varepsilon_2 \) AND \( \exists t_1, t_2 \in r \) such that \( \forall g \in X, \varepsilon_1 \leq |t_1[g] - t_2[g]| \leq \varepsilon_2 \).

We have the following result:

**Theorem 4.2.** The semantics \( s'_2 \) is not well-formed.

**Proof:**

Let \( F \) be a set of rules and \( X \subseteq G \), we have \( F \vdash X \rightarrow X \) by the reflexivity axiom. Nevertheless, \( F \not\models_{s'_2} X \rightarrow X \). Let us consider the following counter-example made of 4 tuples \((t_1, t_2, t_3, t_4)\) over a set of 2 genes \((g_1, g_2)\) as depicted in Table 3.

<table>
<thead>
<tr>
<th>( r )</th>
<th>( g_1 )</th>
<th>( g_2 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t_1 )</td>
<td>-1.8</td>
<td>1.8</td>
</tr>
<tr>
<td>( t_2 )</td>
<td>-1.7</td>
<td>0.2</td>
</tr>
<tr>
<td>( t_3 )</td>
<td>0.2</td>
<td>-1.4</td>
</tr>
<tr>
<td>( t_4 )</td>
<td>0.3</td>
<td>-1.8</td>
</tr>
</tbody>
</table>

**Table 3.** Counter-example for \( s'_2 \)

Let us consider the thresholds \( \varepsilon_1 = 0.0 \) and \( \varepsilon_2 = 0.2 \). We can see that \( r \not\models_{s'_2} g_2 \rightarrow g_2 \) because \( \exists t_1, t_2 \in r \) such that \( 0.0 \leq |t_1[g_2] - t_2[g_2]| \leq 0.2 \). By the way, the result is proved since the reflexivity axiom is not sound.

\[\Box\]

5. Implementation

We have implemented the generation of rules as a C++/STL modules integrated into an open-source freeware devoted to microarray data analysis: MeV (MultiExperimentViewer) [28]. This tool is a part
of an application suite, called TM4, developed by The Institute for Genomic Research (TIGR). These tools devoted to microarray data propose various functions such as storing the data, image analysis, normalization, interpretation of the results.

MeV is the application devoted to the analysis of gene expression data. Furthermore, MeV takes in input several file formats resulting from various image analysis software, has an important number of functionalities already integrated and is based on a GUI easy to use for biologists.

5.1. Interface for the biologists

For the interface, we chose to limit as much as possible the options proposed to the users to make it easier. An example of the graphical user interface developed on top of MeV is presented in Figure 4.

![Graphical user interface of the software](image)

Figure 4. Graphical user interface of the software

At first biologists have to specify the genes they want to see in right hand sides of their rules. This step allows to concentrate on interesting genes. Then, the user chooses one semantics out of three depending on its objectives and its data.

A choice of relevant thresholds is then proposed according to either the distribution of the *levels* of expression for the first semantics, or the distribution of the *variations* of the levels of expression for
second, or the distribution of the *evolution* of the levels of expression for the last one. Nevertheless, biologists can set up manually the thresholds which best fit into their requirements. Note also that different thresholds could be defined for each gene.

### 5.2. Post-processing of rules

The software was tested on several datasets and we were naturally interested in the post-processing of rules. Without being exhaustive, five quality measures (support, confidence, lift, leverage and conviction) are computed to be able to sort the rules (see Figure 5).

![Figure 5. Graphical user interface of the software](image)

These interestingness measures were introduced for association rules [2, 8, 27] but here the computation of these measures is adapted to the three semantics.

All the measures for a rule $X \rightarrow Y$ are computed from four initial values, depending on the chosen semantics:

For the semantics $s_1$, these parameters are computed as follows:
• \( n \): number of tuples \( t \) in the relation.

• \( n_X \): number of tuples \( t \) such that \( \forall g \in X, \varepsilon_1 \leq t[g] \leq \varepsilon_2 \).

• \( n_Y \): number of tuples \( t \) such that \( \forall g \in Y, \varepsilon_1 \leq t[g] \leq \varepsilon_2 \).

• \( n_{XY} \): number of tuples \( t \) such that \( \forall g \in X, \varepsilon_1 \leq t[g] \leq \varepsilon_2 \) and \( \forall g \in Y, \varepsilon_1 \leq t[g] \leq \varepsilon_2 \).

For the semantics \( s_2 \), the computation is different since couples of samples are considered:

• \( n \): number of couples of tuples \( t_1, t_2 \) in the relation.

• \( n_X \): number of couples of tuples \( t_1, t_2 \) such that \( \forall g \in X, \varepsilon_1 \leq |t_1[g] - t_2[g]| \leq \varepsilon_2 \).

• \( n_Y \): number of couples of tuples \( t_1, t_2 \) such that \( \forall g \in Y, \varepsilon_1 \leq |t_1[g] - t_2[g]| \leq \varepsilon_2 \).

• \( n_{XY} \): number of couples of tuples \( t_1, t_2 \) such that \( \forall g \in X, \varepsilon_1 \leq |t_1[g] - t_2[g]| \leq \varepsilon_2 \) and \( \forall g \in Y, \varepsilon_1 \leq |t_1[g] - t_2[g]| \leq \varepsilon_2 \).

For the semantics \( s_3 \), we consider couples of consecutive samples:

• \( n \): number of couples of consecutive tuples \( t_i, t_{i+1} \) in the relation.

• \( n_X \): number of couples of tuples \( t_i, t_{i+1} \) such that \( \forall g \in X, \varepsilon_1 \leq t_{i+1}[g] - t_i[g] \leq \varepsilon_2 \).

• \( n_Y \): number of couples of tuples \( t_i, t_{i+1} \) such that \( \forall g \in Y, \varepsilon_1 \leq t_{i+1}[g] - t_i[g] \leq \varepsilon_2 \).

• \( n_{XY} \): number of couples of tuples \( t_i, t_{i+1} \) such that \( \forall g \in X, \varepsilon_1 \leq t_{i+1}[g] - t_i[g] \leq \varepsilon_2 \) and \( \forall g \in Y, \varepsilon_1 \leq t_{i+1}[g] - t_i[g] \leq \varepsilon_2 \).

From these parameters, we can define for the rule \( X \rightarrow Y \), five quality measures in the following way:

\[
\text{Support} (X \rightarrow Y) = \text{Support} (Y \rightarrow X) = P(XY) = n_{XY}/n.
\]

It corresponds to the probability that \( X \) and \( Y \) are simultaneously satisfied [2].

\[
\text{Confidence} (X \rightarrow Y) = P(XY|X) = n_{XY}/n_X.
\]

It corresponds to the probability that \( Y \) is satisfied knowing that \( X \) is satisfied. When the confidence equals to 1, we say that the rule is exact, otherwise it is said approximate[2].

\[
\text{Lift} (X \rightarrow Y) = \text{Lift} (Y \rightarrow X) = \frac{P(XY) \cdot P(X)P(Y)}{P(X)P(Y)} = \frac{(n_{XY} \ast n)}{(n_X \ast n_Y)}.
\]

This indication measures the dependence between \( X \) and \( Y \). It corresponds to the quotient between the actual probability to have \( X \) and \( Y \) satisfied and the expected probability which we would obtain if \( X \) and \( Y \) were independent [8]. A lift of 1 means that both variables are totally independent.

\[
\text{Leverage} (X \rightarrow Y) = \text{Leverage} (Y \rightarrow X) = P(XY) \cdot P(X)\ast P(Y) = \frac{(n_{XY} \ast n)}{(n_X/\ast n_Y)} - \frac{(n_X/\ast n_Y)}{(n_X/\ast n_Y)}.
\]

The leverage also measures the dependence between \( X \) and \( Y \). It measures the difference between the actual probability to have \( X \) and \( Y \) satisfied and the expected probability which we would obtain if \( X \) and \( Y \) were independent [27]. A leverage equals to 0 means that \( X \) and \( Y \) are independent.
Conviction \((X \rightarrow Y) = (P(X) \times P(\text{not}Y)) / P(X \text{ and not}Y) = (n_X \times (n - n_Y))/(n \times (n_X - n_{XY}))\). The conviction compares the probability to have \(X\) satisfied and \(Y\) not satisfied if they were independent with the actual probability to have \(X\) satisfied and \(Y\) not satisfied \([8]\). A conviction equals to 1 means that \(X\) and \(Y\) are independent. Note that for exact rules, the conviction cannot be computed since \(P(X \text{ and not}Y)\) equals to 0.

The user looks at only the rules s/he considers interesting according to these various indications. Other quality measures of rules, like those defined in \([31]\) can be integrated without any overhead.

### 6. Application from breast cancer tumors

An application has been performed on expression profiles of a sub-sample of genes from breast cancer tumors \([1]\). Publicly available DNA microarray data for 34 young patients, who developed distant metastases within 5 years, and 5,000 genes, were selected by the domain experts (biologists and physicians)\([32]\). The biologists were interested in studying both over- and under-expressed genes.

We present only results obtained on under-expressed genes for which the generated rules were more interesting. To do so, the first semantics was chosen with the following thresholds: \(\varepsilon_1 = -2.00, \varepsilon_2 = -1.18\), that means that a gene is considered as under-expressed if its expression level is between \(-2.00\) and \(-1.18\). Moreover, the number of studied genes was set up to 24.

For this semantics, 60 rules were generated (46 with a 100% confidence). We have then selected the more interesting rules according to the different quality measures. They are given in Table 4.

<table>
<thead>
<tr>
<th>Rule</th>
<th>Support</th>
<th>Confidence</th>
<th>Lift</th>
<th>Leverage</th>
<th>Conviction</th>
</tr>
</thead>
<tbody>
<tr>
<td>(ESR1 \rightarrow TFF1)</td>
<td>0.41</td>
<td>1.00</td>
<td>1.36</td>
<td>10.9</td>
<td>(\infty)</td>
</tr>
<tr>
<td>(MYB \rightarrow TFF1)</td>
<td>0.38</td>
<td>1.00</td>
<td>1.36</td>
<td>10.12</td>
<td>(\infty)</td>
</tr>
<tr>
<td>(ERBB4, TFF1 \rightarrow MYB)</td>
<td>0.21</td>
<td>1.00</td>
<td>2.62</td>
<td>12.72</td>
<td>(\infty)</td>
</tr>
<tr>
<td>(ERBB4, ESR1 \rightarrow MYB)</td>
<td>0.21</td>
<td>1.00</td>
<td>2.62</td>
<td>12.72</td>
<td>(\infty)</td>
</tr>
<tr>
<td>(KRT18 \rightarrow ESR1)</td>
<td>0.21</td>
<td>1.00</td>
<td>2.43</td>
<td>12.11</td>
<td>(\infty)</td>
</tr>
<tr>
<td>(ERBB4, TFF1 \rightarrow ESR1)</td>
<td>0.21</td>
<td>1.00</td>
<td>2.43</td>
<td>12.11</td>
<td>(\infty)</td>
</tr>
<tr>
<td>(ERBB4, MYB \rightarrow ESR1)</td>
<td>0.21</td>
<td>1.00</td>
<td>2.43</td>
<td>12.11</td>
<td>(\infty)</td>
</tr>
<tr>
<td>(BCL2, ESR1 \rightarrow MYB)</td>
<td>0.18</td>
<td>1.00</td>
<td>2.62</td>
<td>10.9</td>
<td>(\infty)</td>
</tr>
<tr>
<td>(BCL2, MYB \rightarrow ESR1)</td>
<td>0.18</td>
<td>1.00</td>
<td>2.43</td>
<td>10.38</td>
<td>(\infty)</td>
</tr>
<tr>
<td>(ERBB4, KRT18 \rightarrow MYB)</td>
<td>0.15</td>
<td>1.00</td>
<td>2.62</td>
<td>9.08</td>
<td>(\infty)</td>
</tr>
</tbody>
</table>

Table 4. Rules from breast cancer tumors

Results obtained so far to reveal interactions between over-expressed and under-expressed genes are promising. We demonstrated the rule \(ESR1\) under-expressed implies \(TFF1\) under-expressed, an already well-known interaction for this kind of tumor \([18]\): \(ESR1\) encodes a nuclear receptor, a super-family of ligand activated transcription factors that modulate specific gene expression. Estrogen exerts its effects...
only through interaction with estrogen receptor. TFF1 (pS2) is an estrogen-inducible gene involved in various biological processes. In the absence of ESR1 expression, estrogen cannot regulate the mRNA level of this gene. The other rules have similar biological interpretations [19].

At present, supplementary experiments need to be done to obtain new gene expression data of mammary tumors. Indeed, it is worth verifying whether rules discovered in a dataset remain valid in another dataset.

7. Conclusion

In order to attempt a reverse engineering of gene regulatory networks from gene expression data, we have proposed an on-going work aiming at defining different semantics of rules between genes, fitting in the same theoretical framework. Such rules form a complementary and hopefully new knowledge with respect to classical unsupervised techniques used so far [13].

The framework proposed in this paper, based on Armstrong’s axiom system, is able to deal with different kinds of semantics in a unified manner. The semantics proposed for gene expression data have been implemented as an extension of an open-source software dedicated to the analysis of microarray data (MeV of TIGR Institute).

We believe that an interesting open issue remains to define new semantics devoted to gene expression data. Moreover, soundness and completeness of the Armstrong’s axiom system have to be proved for each new semantics. We are currently working on a generic definition of a semantics which ensures that a semantics is well-formed if and only if it complies with this definition. To end up, this work may also suggest a more interactive process of discovery of rules, which would consist in requiring to the biologists some “templates” for the rules they are interested in, and then determining the semantics for which these rules are satisfied.

References


