Prediction of Protein Interactions on HIV-1–Human PPI Data Using a Novel Closure-Based Integrated Approach

Kartick Chandra Mondal[1], Nicolas Pasquier[1], Anirban Mukhopadhyay[2], Celia da Costa Pereira[1], Ujjwal Maulikand[3], and Andrea Tettamanzi[4]

[1] I3S (CNRS UMR 6070), Université de Nice-Sophia Antipolis, 06903 Sophia-Antipolis, France
[2] Department of Computer Science and Engineering, University of Kalyani, Kalyani, India
[3] Department of Compter Science and Engineering, University of Jadavpur, Kolkata, India
[4] Information Technology Department, Università degli Studi di Milano, Milan, Italy

e-mails: mondal@i3s.unice.fr, pasquier@i3s.unice.fr, anirban@klyuniv.ac.in, celia.pereira@unice.fr, umaulik@cse.jdvu.ac.in, andrea.tettamanzi@unimi.it

Abstract. Discovering Protein-Protein Interactions (PPI) is a new interesting challenge in computational biology. Identifying interactions among proteins was shown to be useful for finding new drugs and preventing several kinds of diseases. The identification of interactions between HIV-1 proteins and Human proteins is a particular PPI problem whose study might lead to the discovery of drugs and important interactions responsible for AIDS. We present the FIST algorithm for extracting hierarchical bi-clusters and minimal covers of association rules in one process. This algorithm is based on the frequent closed itemsets framework to efficiently generate a hierarchy of conceptual clusters and non-redundant sets of association rules with supporting object lists. Experiments conducted on a HIV-1 and Human proteins interaction dataset show that the approach efficiently identifies interactions previously predicted in the literature and can be used to predict new interactions based on previous biological knowledge.

1 Introduction

Acquired Immune Deficiency Syndrome (AIDS) is the last stage of HIV infection in human body. At this stage, the human immune system fails to protect the body from any kind of infections, and this eventually leads to death. HIV is a member of the retrovirus family (lentivirus) which infects important cells in the human immune system. This kind of infection is due to the interaction between proteins of both the virus and the human host in the human cells. Predicting such interactions is an important goal of PPI research. In particular, analyzing well-known interactions and finding new interactions can provide useful information to find new drugs and discover the reasons and mechanisms of this kind of viral disease [2].

PPI databases contain information about the fact that proteins can interact if they come into contact. The absence of such information does not imply that they cannot interact with each other as there is no information about non-interacting proteins. The HIV-1-Human PPI dataset is a database containing possible viral and human protein interactions. As stated above, only positive interactions are shown.

Several approaches for predicting interactions have been studied in the literature. These approaches are based on Bayesian networks [12], random forest classifiers [14], mixture-of-feature-expert classifiers [20], kernel methods [24, 5], or decision trees [28]. Most of them have been used to find interactions within a single organism, like yeast or human (intra-species interactions). Recently, two approaches have been proposed to predict the set of interactions between HIV-1 and human host cellular proteins [22, 16]. In particular, in [22] the authors proposed a supervised learning framework that integrates heterogeneous biological information to predict inter-species interactions. However, this approach solves the classification problem using the random forest classifier which, like most of the above mentioned approaches, needs both positive and negative samples of PPIs. Negative samples here are pairs of human and HIV proteins known not to interact, but such "negative interactions" (or, better, proven absence of interactions) are not known in the current state of knowledge in the PPI problem studied here. Negative samples have then to be prepared, for example by randomly selecting proteins pairs that are not present in the database, thus leading to a high dependency between the classifier performance and the choice of the negative samples. The approach proposed by Mukhopadhyay *et al.* [16] uses the well-known *Apriori* algorithm for mining association rules. The particularity of such an approach is that only information based on positive samples is used to predict viral-human interactions (inter-species interactions). This is also the case for the approach proposed here.
In this paper, we present FIST, a novel approach to integrated bi-clustering and association rule mining, whose aim is threefold: in a single process, (i) to efficiently mine frequent closed itemsets and generators, (ii) to generate minimal non-redundant covers of association rules, and (iii) to generate hierarchical conceptual bi-clusters. Moreover, compared to classical association rule mining methods, the list of objects (rows) of the dataset supporting each association rule is generated. From the viewpoint of bi-clustering [15], the generated clusters form a hierarchical lattice structure and can overlap, allowing an object to belong to several bi-clusters, if relevant. Another important aim of the FIST approach is to find out interactions between proteins and features in order to extract relationships between annotations (biological and publication) and interactions.
FIST has been validated by applying it to an HIV-1-Human PPI dataset for finding interactions between viral and host proteins. Most existing approaches extract relationships either between viral proteins or between host proteins. Unlike them, FIST extracts bi-clusters and association rules that show relationships involving viral proteins, host proteins, or both at the same time. Results are presented in the following sections.
The rest of the paper is organized as follows. The integrated frequent closed itemset based approach is presented in Section 2 and the FIST algorithm is described in Section 3. In Section 4, we present and discuss experimental results. Finally, Section 5 concludes the paper.

2 Closure Concept based Integrated Approach

Early approaches to association rule mining showed that the problem can be divided into two parts: first, find frequent itemsets with their supports, which is the most time-consuming part, and then generate association rules from these itemsets [1]. Then, the frequent closed itemsets (FCIs) framework was defined to improve the efficiency of the mining in case of non-sparse data [17, 26]. These frequent closed itemsets, defined using the Galois closure [10], are a sub-order of the subset lattice. This framework was later used to define minimal covers, or

bases, of association rules [3, 18, 27]. This approach relies on the property that the frequent closed itemsets with supports constitute a non-redundant minimal representation of the frequent itemsets and their supports. It was experimentally shown that the set of frequent closed itemsets is on average much smaller for real-life datasets, thus making this process faster than directly mining frequent itemsets. Association rules, or association rule bases, are then directly generated from the frequent closed itemsets. See [6] for a comprehensive survey on association rule mining.

The FIST approach aims at providing the user with a minimal set of knowledge patterns representing relationships between data values in the dataset, without information loss. For this task, two types of patterns are generated: informative bases for association rules and hierarchical conceptual clusters. These compact sets of patterns can then be searched for specific information such as intra- and inter-species protein interactions, or relationships between protein interactions and features (biological annotations and characteristics, publications, etc.).

Extracted patterns depict relationships between proteins, which are viral or host proteins, or both. Let $V = \{v_1, \ldots, v_N\}$ be the set of viral proteins and $H = \{h_1, \ldots, h_M\}$ the set of human host proteins. We consider three possible kinds of patterns:

- $r_1 : v_1, v_2, \ldots, v_n \Longleftrightarrow h_1, h_2, \ldots, h_m$ where $v_i \in V$, $h_j \in H$;
- $r_2 : v_1, v_2, \ldots, v_n \Longrightarrow v_{n+1}, v_{n+2}, \ldots, v_{n+p}$ where $\{v_1, v_2, \ldots, v_n\} \cap \{v_{n+1}, v_{n+2}, \ldots, v_{n+p}\} = \emptyset$ and $v_i \in V$;
- $r_3 : h_1, h_2, \ldots, h_m \Longrightarrow h_{m+1}, h_{m+2}, \ldots, h_{m+q}$ where $\{h_1, h_2, \ldots, h_m\} \cap \{h_{m+1}, h_{m+2}, \ldots, h_{m+q}\} = \emptyset$ and $h_j \in H$.

Type r_1 relationships capture interactions between some viral proteins and some host proteins (inter-species PPI). Identifying such rules is similar to the problem of bi-clustering, that is, in the context of FIST, finding frequent closed itemsets with related object identifiers. Type r_2 and r_3 relationships are association rule patterns showing implications among viral proteins and host proteins respectively (intra-species PPI). Classification methods usually need both positive and negative examples of the predicted class, e.g., interacting and non-interacting protein pairs, in order to achieve an optimal supervised classification. However, in the case of HIV-1-Human PPI, information on non-interacting pairs of proteins is not available [8, 19]. Hence, descriptive methods, such as unsupervised classification (clustering) and association rule extraction, seem better suited to this PPI problem.

FIST was designed both to extract in one process different kinds of knowledge patterns, bi-clusters and association rules, and to extract additional information for each of these patterns compared to classical approaches. It can process discrete numerical, boolean, textual and nominal data. As for the majority of similar methods, in the case of continuous numerical data, a discretization method has to be applied before processing the data with FIST. This is for example the case for numerical gene expression data where numerical values must be discretized to identify "up-regulated", "unchanged" and "down-regulated" genes (rows) for each experimental biological conditions (columns). See [25] for a recent discussion on discretization methods used in data mining.

Consider the example dataset D_1 in Figure 1 where H_1 to H_6 are human proteins, V_1 to V_5 are viral proteins and Annot columns represent annotations of human proteins extracted from biological knowledge bases (Gene Ontology, KEGG, etc.) and publication bases (Pubmed, Reactome pathways, etc.). These annotations, represented as nominal data, describe biological knowledge on human proteins such as biological functions or characteristics (F_n) or bibliographic citation references (B_m). A "1" in column V_i for row H_j means that there is a positive (i.e.,

experimentally verified) interaction between H_j and V_i, while "-" means that no interaction has been reported. For example, we can state that there is a positive interaction between human protein H_1 and viral proteins V_1, V_3 and V_4, while no interaction between H_1, V_2 and V_5 has been reported. Besides, we can also state that H_1 is annotated by biological annotations F_1 and F_2 and referenced by bibliographical annotation B_1.

OID	V_1	V_2	V_3	V_4	V_5	Annot	Annot	Annot
H_1	1	-	1	1	-	F_1	F_2	B_1
H_2	1	-	1	-	-	F_2	B_1	B_2
H_3	-	-	1	-	1	B_3	-	-
H_4	1	-	1	1	-	F_2	F_3	B_1
H_5	-	1	-	-	-	F_4	-	-
H_6	1	-	1	1	1	F_2	B_1	B_3

Fig. 1: Example dataset D_1.

Conceptual bi-clusters extracted by FIST form a hierarchical structure and both HIV and Human proteins can participate to several bi-clusters according to their co-occurrences in the data. In the context of HIV-1-Human PPI, each hierarchical conceptual cluster associates a list of HIV proteins and a list of Human proteins that interact. FIST bi-clusters also associate to each bi-cluster the minimal set of common properties, called *generators*, required to construct it [11, 17]. Moreover, unlike most clustering methods, conceptual clustering does not need to define the number of clusters before the process as data are grouped according to their co-occurrences in the dataset. The Hasse diagram of the lattice structure of the four bi-clusters extracted from D_1 for $minsupport = 2/6$ is shown in Figure 2. The top bi-cluster in this figure is irrelevant from the viewpoint of informativeness and is not generated by FIST; it is represented here for completeness of the lattice. Examining the rightmost bi-cluster, we can see in this lattice that human proteins H_3 and H_6 both interact with viral proteins V_3 and V_5 and are cited in bibliographical reference B_3. The leftmost bi-clusters show that human proteins H_1, H_2, H_4, and H_6 all interact with viral proteins V_1 and V_3, are all annotated with F_2, and are cited in bibliographical reference B_1 and that human proteins H_1, H_4, and H_6 all interact with viral proteins V_1, V_3, and V_4, are all annotated with F_2, and are cited in bibliographical reference B_1. We can also see that the viral protein that interacts with the greatest number of human proteins is V_3, which interacts with H_1, H_2, H_3, H_4, and H_6, and that this interaction is the only property common to these five human proteins. It should be noted that for this *minsupport* value, there are 4 frequent closed itemsets, whereas there are 37 frequent itemsets for dataset D_1. These frequent closed itemsets are reprensented in the left element of the bi-clusters.

Association rules are implication rules of the form: $\{r$: *antecedent* $\Longrightarrow$ *consequent*, support(r), confidence$(r)\}$ where *antecedent* and *consequent* are sets of data values, support(r) is the number of objects (rows of the dataset) supporting the rule and confidence(r) is the proportion of rows verifying the rule in the dataset. FIST aims at improving the process compared to frequent itemsets based approaches. First, the number of extracted rules can be reduced by a significant proportion as redundant rules can represent the majority of extracted rules [3, 26]. Association rules extracted by FIST are constructed using generators, as antecedents, and frequent closed itemsets, as consequents. These rules, also called *min-max associ-*

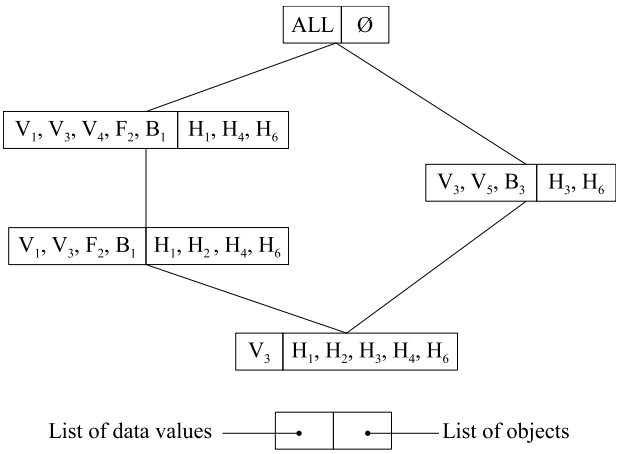

Fig. 2: Hierarchical bi-clusters extracted from D_1 for *minsupport* = 2/6.

ation rules, constitute the *informative base of association rules* [18]. FIST extracts rules in two disctinct sets: *exact association rules* that have confidence = 1, i.e., with no counter example in the dataset, and *approximate association rules*, having confidence < 1. It also extends the association rules by adding information to each rule: The list of objects (rows) supporting each one is also generated, allowing the user to see which objects verify this rule in the dataset as shown in Figure 3. We can see that for *minsupport* = 2/6 and *minconfidence* = 2/6, 6 exact and 6 approximate min-max association rules are generated by FIST from dataset D_1 whereas 117 exact and 75 approximate association rules are generated by classical Apriori-like approaches. These 192 association rules are shown in Figure 4.

Association rule	supp	conf	Objects
$B_1 \Longrightarrow V_1, V_3, F_2$	4	1	H_1, H_2, H_4, H_6
$F_2 \Longrightarrow V_1, V_3, B_1$	4	1	H_1, H_2, H_4, H_6
$V_1 \Longrightarrow V_3, F_2, B_1$	4	1	H_1, H_2, H_4, H_6
$V_4 \Longrightarrow V_1, V_3, F_2, B_1$	3	1	H_1, H_4, H_6
$B_3 \Longrightarrow V_3, V_5$	2	1	H_3, H_6
$V_5 \Longrightarrow V_3, B_3$	2	1	H_3, H_6
$V_3 \Longrightarrow V_1, F_2, B_1$	4	0.80	H_1, H_2, H_4, H_6
$B_1 \Longrightarrow V_1, V_3, V_4, F_2$	3	0.75	H_1, H_4, H_6
$F_2 \Longrightarrow V_1, V_3, V_4, B_1$	3	0.75	H_1, H_4, H_6
$V_1 \Longrightarrow V_3, V_4, F_2, B_1$	3	0.75	H_1, H_4, H_6
$V_3 \Longrightarrow V_1, V_4, F_2, B_1$	3	0.60	H_1, H_4, H_6
$V_3 \Longrightarrow V_5, B_3$	2	0.40	H_3, H_6

Fig. 3: Minimal non-redundant association rules extracted from D_1 for *minsupport* = 2/6 and *minconfidence* = 2/6.

#	Association rule	supp	conf	#	Association rule	supp	conf	#	Association rule	supp	conf
1.	$V_1 \Longrightarrow V_3$	4	1	65.	$V_4 \Longrightarrow F_2\ B_1$	3	1	129.	$V_1 \Longrightarrow V_3\ V_4$	3	0.75
2.	$F_2 \Longrightarrow V_1$	4	1	66.	$V_3\ V_4\ F_2 \Longrightarrow V_1$	3	1	130.	$V_1\ F_2 \Longrightarrow V_4$	3	0.75
3.	$V_1 \Longrightarrow F_2$	4	1	67.	$V_1\ V_4\ F_2 \Longrightarrow V_3$	3	1	131.	$F_2 \Longrightarrow V_1\ V_4$	3	0.75
4.	$B_1 \Longrightarrow V_1$	4	1	68.	$V_1\ V_3\ V_4 \Longrightarrow F_2$	3	1	132.	$V_1 \Longrightarrow V_4\ F_2$	3	0.75
5.	$V_1 \Longrightarrow B_1$	4	1	69.	$V_4\ F_2 \Longrightarrow V_1\ V_3$	3	1	133.	$V_1\ B_1 \Longrightarrow V_4$	3	0.75
6.	$F_2 \Longrightarrow V_3$	4	1	70.	$V_3\ V_4 \Longrightarrow V_1\ F_2$	3	1	134.	$B_1 \Longrightarrow V_1\ V_4$	3	0.75
7.	$B_1 \Longrightarrow V_3$	4	1	71.	$V_1\ V_4 \Longrightarrow V_3\ F_2$	3	1	135.	$V_1 \Longrightarrow V_4\ B_1$	3	0.75
8.	$B_1 \Longrightarrow F_2$	4	1	72.	$V_4 \Longrightarrow V_1\ V_3\ F_2$	3	1	136.	$V_3\ F_2 \Longrightarrow V_4$	3	0.75
9.	$F_2 \Longrightarrow B_1$	4	1	73.	$V_3\ V_4\ B_1 \Longrightarrow V_1$	3	1	137.	$F_2 \Longrightarrow V_3\ V_4$	3	0.75
10.	$V_3\ F_2 \Longrightarrow V_1$	4	1	74.	$V_1\ V_4\ B_1 \Longrightarrow V_3$	3	1	138.	$V_3\ B_1 \Longrightarrow V_4$	3	0.75
11.	$V_1\ F_2 \Longrightarrow V_3$	4	1	75.	$V_1\ V_3\ V_4 \Longrightarrow B_1$	3	1	139.	$B_1 \Longrightarrow V_3\ V_4$	3	0.75
12.	$V_1\ V_3 \Longrightarrow F_2$	4	1	76.	$V_4\ B_1 \Longrightarrow V_1\ V_3$	3	1	140.	$F_2\ B_1 \Longrightarrow V_4$	3	0.75
13.	$F_2 \Longrightarrow V_1\ V_3$	4	1	77.	$V_3\ V_4 \Longrightarrow V_1\ B_1$	3	1	141.	$B_1 \Longrightarrow V_4\ F_2$	3	0.75
14.	$V_1 \Longrightarrow V_3\ F_2$	4	1	78.	$V_1\ V_4 \Longrightarrow V_3\ B_1$	3	1	142.	$F_2 \Longrightarrow V_4\ B_1$	3	0.75
15.	$V_3\ B_1 \Longrightarrow V_1$	4	1	79.	$V_4 \Longrightarrow V_1\ V_3\ B_1$	3	1	143.	$V_1\ V_3\ F_2 \Longrightarrow V_4$	3	0.75
16.	$V_1\ B_1 \Longrightarrow V_3$	4	1	80.	$V_4\ F_2\ B_1 \Longrightarrow V_1$	3	1	144.	$V_3\ F_2 \Longrightarrow V_1\ V_4$	3	0.75
17.	$V_1\ V_3 \Longrightarrow B_1$	4	1	81.	$V_1\ V_4\ B_1 \Longrightarrow F_2$	3	1	145.	$V_1\ F_2 \Longrightarrow V_3\ V_4$	3	0.75
18.	$B_1 \Longrightarrow V_1\ V_3$	4	1	82.	$V_1\ V_4\ F_2 \Longrightarrow B_1$	3	1	146.	$V_1\ V_3 \Longrightarrow V_4\ F_2$	3	0.75
19.	$V_1 \Longrightarrow V_3\ B_1$	4	1	83.	$V_4\ B_1 \Longrightarrow V_1\ F_2$	3	1	147.	$F_2 \Longrightarrow V_1\ V_3\ V_4$	3	0.75
20.	$F_2\ B_1 \Longrightarrow V_1$	4	1	84.	$V_4\ F_2 \Longrightarrow V_1\ B_1$	3	1	148.	$V_1 \Longrightarrow V_3\ V_4\ F_2$	3	0.75
21.	$V_1\ B_1 \Longrightarrow F_2$	4	1	85.	$V_1\ V_4 \Longrightarrow F_2\ B_1$	3	1	149.	$V_1\ V_3\ B_1 \Longrightarrow V_4$	3	0.75
22.	$V_1\ F_2 \Longrightarrow B_1$	4	1	86.	$V_4 \Longrightarrow V_1\ F_2\ B_1$	3	1	150.	$V_3\ B_1 \Longrightarrow V_1\ V_4$	3	0.75
23.	$B_1 \Longrightarrow V_1\ F_2$	4	1	87.	$V_4\ F_2\ B_1 \Longrightarrow V_3$	3	1	151.	$V_1\ B_1 \Longrightarrow V_3\ V_4$	3	0.75
24.	$F_2 \Longrightarrow V_1\ B_1$	4	1	88.	$V_3\ V_4\ B_1 \Longrightarrow F_2$	3	1	152.	$V_1\ V_3 \Longrightarrow V_4\ B_1$	3	0.75
25.	$V_1 \Longrightarrow F_2\ B_1$	4	1	89.	$V_3\ V_4\ F_2 \Longrightarrow B_1$	3	1	153.	$B_1 \Longrightarrow V_1\ V_3\ V_4$	3	0.75
26.	$F_2\ B_1 \Longrightarrow V_3$	4	1	90.	$V_4\ B_1 \Longrightarrow V_3\ F_2$	3	1	154.	$V_1 \Longrightarrow V_3\ V_4\ B_1$	3	0.75
27.	$V_3\ B_1 \Longrightarrow F_2$	4	1	91.	$V_4\ F_2 \Longrightarrow V_3\ B_1$	3	1	155.	$V_1\ F_2\ B_1 \Longrightarrow V_4$	3	0.75
28.	$V_3\ F_2 \Longrightarrow B_1$	4	1	92.	$V_3\ V_4 \Longrightarrow F_2\ B_1$	3	1	156.	$F_2\ B_1 \Longrightarrow V_1\ V_4$	3	0.75
29.	$B_1 \Longrightarrow V_3\ F_2$	4	1	93.	$V_4 \Longrightarrow V_3\ F_2\ B_1$	3	1	157.	$V_1\ B_1 \Longrightarrow V_4\ F_2$	3	0.75
30.	$F_2 \Longrightarrow V_3\ B_1$	4	1	94.	$V_3\ V_4\ F_2\ B_1 \Longrightarrow V_1$	3	1	158.	$V_1\ F_2 \Longrightarrow V_4\ B_1$	3	0.75
31.	$V_3\ F_2\ B_1 \Longrightarrow V_1$	4	1	95.	$V_1\ V_4\ F_2\ B_1 \Longrightarrow V_3$	3	1	159.	$B_1 \Longrightarrow V_1\ V_4\ F_2$	3	0.75
32.	$V_1\ F_2\ B_1 \Longrightarrow V_3$	4	1	96.	$V_1\ V_3\ V_4\ B_1 \Longrightarrow F_2$	3	1	160.	$F_2 \Longrightarrow V_1\ V_4\ B_1$	3	0.75
33.	$V_1\ V_3\ B_1 \Longrightarrow F_2$	4	1	97.	$V_1\ V_3\ V_4\ F_2 \Longrightarrow B_1$	3	1	161.	$V_1 \Longrightarrow V_4\ F_2\ B_1$	3	0.75
34.	$V_1\ V_3\ F_2 \Longrightarrow B_1$	4	1	98.	$V_4\ F_2\ B_1 \Longrightarrow V_1\ V_3$	3	1	162.	$V_3\ F_2\ B_1 \Longrightarrow V_4$	3	0.75
35.	$F_2\ B_1 \Longrightarrow V_1\ V_3$	4	1	99.	$V_3\ V_4\ B_1 \Longrightarrow V_1\ F_2$	3	1	163.	$F_2\ B_1 \Longrightarrow V_3\ V_4$	3	0.75
36.	$V_3\ B_1 \Longrightarrow V_1\ F_2$	4	1	100.	$V_3\ V_4\ F_2 \Longrightarrow V_1\ B_1$	3	1	164.	$V_3\ B_1 \Longrightarrow V_4\ F_2$	3	0.75
37.	$V_3\ F_2 \Longrightarrow V_1\ B_1$	4	1	101.	$V_1\ V_4\ B_1 \Longrightarrow V_3\ F_2$	3	1	165.	$V_3\ F_2 \Longrightarrow V_4\ B_1$	3	0.75
38.	$V_1\ B_1 \Longrightarrow V_3\ F_2$	4	1	102.	$V_1\ V_4\ F_2 \Longrightarrow V_3\ B_1$	3	1	166.	$B_1 \Longrightarrow V_3\ V_4\ F_2$	3	0.75
39.	$V_1\ F_2 \Longrightarrow V_3\ B_1$	4	1	103.	$V_1\ V_3\ V_4 \Longrightarrow F_2\ B_1$	3	1	167.	$F_2 \Longrightarrow V_3\ V_4\ B_1$	3	0.75
40.	$V_1\ V_3 \Longrightarrow F_2\ B_1$	4	1	104.	$V_4\ B_1 \Longrightarrow V_1\ V_3\ F_2$	3	1	168.	$V_1\ V_3\ F_2\ B_1 \Longrightarrow V_4$	3	0.75
41.	$B_1 \Longrightarrow V_1\ V_3\ F_2$	4	1	105.	$V_4\ F_2 \Longrightarrow V_1\ V_3\ B_1$	3	1	169.	$V_3\ F_2\ B_1 \Longrightarrow V_1\ V_4$	3	0.75
42.	$F_2 \Longrightarrow V_1\ V_3\ B_1$	4	1	106.	$V_3\ V_4 \Longrightarrow V_1\ F_2\ B_1$	3	1	170.	$V_1\ F_2\ B_1 \Longrightarrow V_3\ V_4$	3	0.75
43.	$V_1 \Longrightarrow V_3\ F_2\ B_1$	4	1	107.	$V_1\ V_4 \Longrightarrow V_3\ F_2\ B_1$	3	1	171.	$V_1\ V_3\ B_1 \Longrightarrow V_4\ F_2$	3	0.75
44.	$V_4 \Longrightarrow V_1$	3	1	108.	$V_4 \Longrightarrow V_1\ V_3\ F_2\ B_1$	3	1	172.	$V_1\ V_3\ F_2 \Longrightarrow V_4\ B_1$	3	0.75
45.	$V_4 \Longrightarrow V_3$	3	1	109.	$V_5 \Longrightarrow V_3$	2	1	173.	$F_2\ B_1 \Longrightarrow V_1\ V_3\ V_4$	3	0.75
46.	$V_4 \Longrightarrow F_2$	3	1	110.	$B_3 \Longrightarrow V_3$	2	1	174.	$V_3\ B_1 \Longrightarrow V_1\ V_4\ F_2$	3	0.75
47.	$V_4 \Longrightarrow B_1$	3	1	111.	$B_3 \Longrightarrow V_5$	2	1	175.	$V_3\ F_2 \Longrightarrow V_1\ V_4\ B_1$	3	0.75
48.	$V_3\ V_4 \Longrightarrow V_1$	3	1	112.	$V_5 \Longrightarrow B_3$	2	1	176.	$V_1\ B_1 \Longrightarrow V_3\ V_4\ F_2$	3	0.75
49.	$V_1\ V_4 \Longrightarrow V_3$	3	1	113.	$V_5\ B_3 \Longrightarrow V_3$	2	1	177.	$V_1\ F_2 \Longrightarrow V_3\ V_4\ B_1$	3	0.75
50.	$V_4 \Longrightarrow V_1\ V_3$	3	1	114.	$V_3\ B_3 \Longrightarrow V_5$	2	1	178.	$V_1\ V_3 \Longrightarrow V_4\ F_2\ B_1$	3	0.75
51.	$V_4\ F_2 \Longrightarrow V_1$	3	1	115.	$V_3\ V_5 \Longrightarrow B_3$	2	1	179.	$B_1 \Longrightarrow V_1\ V_3\ V_4\ F_2$	3	0.75
52.	$V_1\ V_4 \Longrightarrow F_2$	3	1	116.	$B_3 \Longrightarrow V_3\ V_5$	2	1	180.	$F_2 \Longrightarrow V_1\ V_3\ V_4\ B_1$	3	0.75
53.	$V_4 \Longrightarrow V_1\ F_2$	3	1	117.	$V_5 \Longrightarrow V_3\ B_3$	2	1	181.	$V_1 \Longrightarrow V_3\ V_4\ F_2\ B_1$	3	0.75
54.	$V_4\ B_1 \Longrightarrow V_1$	3	1	118.	$V_3 \Longrightarrow V_1$	4	0.8	182.	$V_3 \Longrightarrow V_4$	3	0.6
55.	$V_1\ V_4 \Longrightarrow B_1$	3	1	119.	$V_3 \Longrightarrow F_2$	4	0.8	183.	$V_3 \Longrightarrow V_1\ V_4$	3	0.6
56.	$V_4 \Longrightarrow V_1\ B_1$	3	1	120.	$V_3 \Longrightarrow B_1$	4	0.8	184.	$V_3 \Longrightarrow V_4\ F_2$	3	0.6
57.	$V_4\ F_2 \Longrightarrow V_3$	3	1	121.	$V_3 \Longrightarrow V_1\ F_2$	4	0.8	185.	$V_3 \Longrightarrow V_4\ B_1$	3	0.6
58.	$V_3\ V_4 \Longrightarrow F_2$	3	1	122.	$V_3 \Longrightarrow V_1\ B_1$	4	0.8	186.	$V_3 \Longrightarrow V_1\ V_4\ F_2$	3	0.6
59.	$V_4 \Longrightarrow V_3\ F_2$	3	1	123.	$V_3 \Longrightarrow F_2\ B_1$	4	0.8	187.	$V_3 \Longrightarrow V_1\ V_4\ B_1$	3	0.6
60.	$V_4\ B_1 \Longrightarrow V_3$	3	1	124.	$V_3 \Longrightarrow V_1\ F_2\ B_1$	4	0.8	188.	$V_3 \Longrightarrow V_4\ F_2\ B_1$	3	0.6
61.	$V_3\ V_4 \Longrightarrow B_1$	3	1	125.	$V_1 \Longrightarrow V_4$	3	0.75	189.	$V_3 \Longrightarrow V_1\ V_4\ F_2\ B_1$	3	0.6
62.	$V_4 \Longrightarrow V_3\ B_1$	3	1	126.	$F_2 \Longrightarrow V_4$	3	0.75	190.	$V_3 \Longrightarrow V_5$	2	0.4
63.	$V_4\ B_1 \Longrightarrow F_2$	3	1	127.	$B_1 \Longrightarrow V_4$	3	0.75	191.	$V_3 \Longrightarrow B_3$	2	0.4
64.	$V_4\ F_2 \Longrightarrow B_1$	3	1	128.	$V_1\ V_3 \Longrightarrow V_4$	3	0.75	192.	$V_3 \Longrightarrow V_5\ B_3$	2	0.4

Fig. 4: Association rules extracted from D_1 for *minsupport* $= 2/6$ and *minconfidence* $= 2/6$.

3 FIST: A Novel Integrated Algorithm

In this section, we present the new FIST (Frequent Itemset mining using Suffix-Trees) algorithm. The FIST algorithm is a three-phase process: (1) preprocessing the dataset, (2) extracting frequent closed itemsets, (3) finding bases for association rules and hierarchical conceptual bi-clusters. Its general flow is shown in Algorithm 1. The input of the algorithm is a dataset represented as a data matrix in which rows, or lines, are called *objects* and columns are called *attributes*. Each distinct value of an *attribute* constitutes an *item*. The FIST algorithm performs one scan of the input dataset to generate a compressed database that is scanned once for generating frequent closed itemsets, generators, bases for association rules, and conceptual bi-clusters.

Algorithm 1 FIST algorithm.

Input: Input dataset, *minsupport* value, *minconfidence* value
Output: Frequent closed itemsets, generators, conceptual clusters, association rules

```
1: begin
     /* Phase 1: Preparing the database */
2:   Generate Item Table
3:   Generate Sorted Frequent Database
     /* Phase 2: Mining frequent closed itemsets */
4:   Create frequent Generalized Itemset Suffix-Tree
5:   Find frequent closed itemsets
     /* Phase 3: Generating knowledge patterns */
6:   Find generators of each frequent closed itemsets
7:   Find conceptual bi-clusters
8:   Generate basis of exact association rules
9:   Generate basis of approximate association rules
10: end
```

3.1 Phase 1: Preparing the Database

The first phase of FIST consists in the preparation of the *Item Table* (*IT*) and the *Sorted Frequent Database* (*SFD*) data structures used in the following phases of the algorithm. These data structures are stored in secondary memory for reuse. In the *SFD* database, each row is the list of *items*, each one representing an attribute value, contained in the corresponding row of the original dataset. An example source dataset D_2 containing 5 attributes and 5 objects is given in Figure 5. This preprocessing phase, which aims at optimizing the efficiency of the extraction and data accesses, is performed in two steps.

OID	C_1	C_2	C_3	A	A
O_1	-	v_2	-	v_5	-
O_2	v_1	v_2	v_3	v_5	-
O_3	v_1	-	v_3	v_4	-
O_4	-	v_2	v_3	v_5	-
O_5	v_1	v_2	v_3	v_5	v_6

Fig. 5: Example dataset D_2.

The first step consists in constructing the *IT* table by mapping attribute values in the dataset, which can be booleans, numerics, nominals or textuals, to *items* represented as discrete numbers. This data representation aims at optimizing the memory space required for data storage and the efficiency of comparison operations. This operation, which is performed only once and requires only one read of the dataset, can be omitted if the dataset contains uniquely discrete numbers. To create this table, a unique number is created for each pair {*attribute*, *value*} using a mapping function. During this operation, the *support* of each item in the dataset, corresponding to its number of occurrences, is counted. Then, using the minimum support threshold value *minsupport* provided by the user, the *infrequent items*, i.e., those with support less than the *minsupport* value, are discarded. Finally, the remaining *frequent items* are sorted in ascending order of their supports to optimize the size of the data structure used in the second phase of the algorithm. During the second step, the *SFD* database is created to reflect the occurrences of frequent items in rows of the original dataset. Rows of the original dataset containing only infrequent items are not represented in the *SFD* database. The example *SFD* database and the corresponding *IT* table for dataset D and *minsupport* = 2/5 are given in Figure 6. In this example, data values A = v_4 and A = v_6 with support 1/5 are infrequent and, given their support values, frequent items are ordered as: {$C_1 = v_1$, $C_2 = v_2$, A = v_5, $C_3 = v_3$}. Notice that these frequent items are ordered first on the support and then in order of appearance in the rows of the dataset. For example, $C_1 = v_1$ is in the first position in the *IT* table because it has a lower support (3), while A = v_5 appears before $C_3 = v_3$ because it is the second in order of appearence while $C_3 = v_3$ appears in the fifth place.

(A) *IT* TABLE

Data	Support	Item
$C_1 = v_1$	3	1
$C_2 = v_2$	4	2
A = v_5	4	3
$C_3 = v_3$	4	4

(B) *SFD* DATABASE

Items			
2	3		
1	2	3	4
1	4		
2	3	4	
1	2	3	4

Fig. 6: *IT* table and *SFD* database for D_2 and *minsupport* = 2/5.

3.2 Phase 2: Mining Frequent Closed Itemsets

During the second phase, which is the core of the FIST algorithm, the frequent closed itemsets are mined from the *SFD* database. This phase is carried out in two steps. The first step is the generation of the *frequent Generalized Itemset Suffix-Tree* (*fGIST*), which is a main memory data structure specific to the FIST algorithm. In the *fGIST* tree, each internal node represents an item, each branch from the root to a leaf represents an itemset, and each leaf node represents the list of numbers of objects (rows) containing this itemset. The second step is the extraction of the frequent closed itemsets from the *fGIST* tree. This extraction is based on inclusion and intersection operations performed on the branches and the sub-branches of the *fGIST* tree.

Creating fGIST tree To create the *fGIST* data structure, each row of the *SFD* database is accessed once from the first to the last. Each row read is represented

as a vector of items associated with the identifier number of the row in the SFD database. Since items were ordered in ascending order of their supports during the construction of the SFD database, they are also sorted in this order in the vector. This vector is then inserted into the *fGIST* tree as a branch, starting from the root, with a leaf containing the identifier number of the row. If this vector of items is already represented as a branch in the tree, that is if an identical row was read before, then only the leaf is updated by adding the identifier number of the row. Then, this process is repeated for all *suffixes* of the vector of items that are sub-vectors obtained by deleting successively one item from the first to the last. In our example, the first branch to be inserted is {2, 3}, then the branch corresponding to its unique suffix {3}. The third branch to be inserted into the tree is {1, 2, 3, 4}, and then the ones corresponding to its suffixes {2, 3, 4}, {3, 4}, and {4}, and so on. The *fGIST* tree for database *SFD* is given in Figure 7.

The insertion of a vector of items in the *fGIST* tree is a recursive procedure starting from the root node of the tree. Each item of the vector is processed sequentially from the first to last. For each item, we test if there is a sub-node, of the current node, representing this item. If this is the case, then we go to this node and repeat the process for the next item of the vector. Otherwise, a new sub-node is created to represent this item as a child of the current node. When the last item of the vector was processed, we test if there is a leaf sub-node of the current node. If this is the case, the row number corresponding to the vector processed is added to the list of row numbers in this leaf. Otherwise, a new leaf sub-node is created with a list of row numbers initialized with the row number corresponding to the vector.

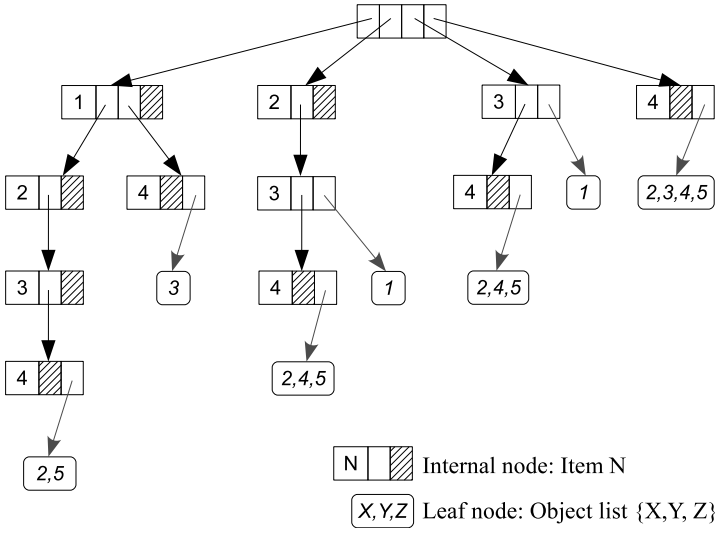

Fig. 7: Frequent Generalized Itemset Suffix-Tree for *SFD* database.

During this process, the whole SFD database is accessed only once. At the end of the process, the *fGIST* tree contains a condensed representation of the frequent itemsets in the dataset. This data structure is optimized for the following phases of the process as the most frequent itemsets resulting of intersections of dataset rows, which are in majority closed itemsets, are represented as branches. This property is ensured by the fact that items are ordered in ascending order of their supports.

Creating FCI Table The second step consists in extracting the FCIs, with the list of objects containing each of them, from the *fGIST* tree. Each entry in the *FCI* table contains two elements: A list of items and the list of numbers of objects containing that itemset in the database.
First, each branch of the *fGIST* tree from the root to a leaf is traversed and a new entry in the *FCI* table is created for the itemset corresponding to that branch. The associated list of numbers of objects is initialized using the leaf node of that branch. The size of this object list corresponds to the support of the itemset in the database.
Then, the non-closed itemsets in the *FCI* table are identified using associated object lists as follows. If an itemset is included in another itemset and both have identical object lists, then the included itemset is not closed and is dropped from the table.
Finally, the frequent closed itemsets not already found are identified by performing intersections between two closed itemsets in the *FCI* table and verifying if the resulting itemset is not infrequent or already present in the table. If this is not the case, that itemset is a new FCI and it is inserted into the table. The associated object list is the result of the union of the object lists of the two intersected itemsets. If at least one new frequent closed itemset is generated in such a way, then the process is repeated for the new generated itemsets. This iterative process ends when no new frequent closed itemset is generated. At the end, the *FCI* table contains all frequent closed itemsets with associated list of objects containing each of them as shown in Figure 8.

Itemset	Object list
{4}	{2, 3, 4, 5}
{1, 4}	{2, 3, 4}
{2, 3}	{1, 2, 4, 5}
{2, 3, 4}	{2, 4, 5}
{1, 2, 3, 4}	{2, 4}

Fig. 8: *FCI* Table for *SFD* database.

3.3 Phase 3: Generating Knowledge Patterns

During the third phase, the conceptual bi-clusters, the generators of frequent closed itemsets and the association rules are extracted from the *FCI* table. The association rules are generated in two distinct sets: a minimal cover for exact association rules and a minimal cover for approximate association rules. These minimal covers, or *bases*, contain, respectively, the non-redundant exact and approximate association rules with minimal antecedent (predictor itemset) and maximal consequent (predicted items) [18]. Minimality is defined here according to the inclusion relation. The pseudo-code of the extraction of these knowledge patterns is given in Algorithm 2.
First, rows in the *FCI* table are sorted in increasing order of itemset sizes (step 1) and output sets BIC, GEN, and AR are initialized with the empty set (step 2). Then, each entry FCI[i] in the *FCI* table is processed successively (steps 3 to 25) for creating hierarchical bi-clusters (step 4) and identifying generators and association rules (steps 6 to 24) as follows. All subsets S of itemset FCI[i].Itemset are generated, sorted in increasing order of their sizes (steps 7 and 8) and processed one by one (steps 10 to 22). For instance, for itemset {2, 3, 4}, the generated

Algorithm 2 Generating knowledge patterns.

Input: FCI table (FCI), *minconfidence* value, Item table (IT)
Output: Bi-clusters (BIC), generators (GEN), association rules (AR)

```
 1: sort FCI in increasing size of itemsets
 2: GEN, BIC, AR ← ∅
 3: for all row FCI[i] in FCI do
 4:    BIC ← {FCI[i].Itemset, FCI[i].Object_list}
 5:    M ← FCI[i].Itemset.size()
 6:    if (M ≥ 2) then
 7:       SUB ← list of subsets of FCI[i].Itemset
 8:       sort SUB in increasing size of subsets
 9:       K ← SUB.size()
10:       for all subset S in SUB do
11:          if (S ∉ GEN) and (S ∉ FCI.Itemset) then
12:             GEN[i] ← S
13:          end if
14:          for j = 1 to K do
15:             if (S.size() + SUB[j].size() = M) and (S ≠ SUB[j]) then
16:                create rule R : {S ⟹ SUB[j]}
17:                if (confidence(R) ≥ minconfidence) and (R ∉ AR) then
18:                   AR ← {R, support(R), confidence(R), FCI[i].Object_list}
19:                end if
20:             end if
21:          end for
22:       end for
23:       SUB ← ∅
24:    end if
25: end for
26: map patterns in BIC, GEN, AR to dataset values
```

subsets are {2}, {3}, {4}, {2, 3}, {2, 4} and {3, 4}. The algorithm first determines if S is a generator of FCI[i].Itemset (steps 11 to 13). Then, all association rules with S as antecedent are generated if their confidence is greater than or equal to the *minconfidence* threshold (steps 14 to 21). Considering itemset {2, 3, 4}, generators {2, 4} and {3, 4} are identified, as they are the only minimal itemsets contained in exactly the same objects as {2, 3, 4}. From these itemsets, rules {2, 4}$\Longrightarrow${3} and {3, 4}$\Longrightarrow${2} are generated. Finally, knowledge patterns in the BIC, GEN, and AR sets are mapped to data values using the *Item Table*, and object numbers are mapped to object identifiers (e.g., gene or protein names) if the source dataset contained such information, in order to simplify their interpretation by the end-user (step 26).

4 Experiments and Discussion

The FIST algorithm has been implemented in Java for portability. Experiments were conducted on a PC with an Intel Core 2 Duo - T5670 Series processor at 1.80 GHz and 4 GB DDR2 of RAM, running under the 32 bits Windows 7 Professional Edition operating system. The PPI dataset used for performance experiments was constructed from the HIV-1-Human Protein Protein Interaction Database of the NIAID [8, 19] available at http://www.ncbi.nlm.nih.gov/RefSeq/HIVInteractions/. This dataset is a matrix of 19 columns corresponding to the different HIV-1 proteins and 1433 rows corresponding to the human proteins. Each cell of the matrix

contains a 1 if there is a positive interaction between the corresponding pair of proteins and a question mark if no interaction is reported. To assess the scalability of FIST when the number of columns increases, a second dataset was constructed by integrating biological and bibliographical annotations of human proteins with these interaction data. These two datasets can be downloaded at *address removed for double-blind paper evaluation.*

4.1 Algorithmic Performance

Figure 9 compares the execution times of FIST (blue curves) and the Java implementation of Apriori (red curves) in Weka available at http://www.cs.waikato.ac.nz/ml/weka/. Three *minsupport* values, 0.001 (0.1%), 0.006 (0.6%) and 0.01 (10%), were used and *minconfidence* was varied between 0.001 (0.1%) and 0.6 (60%). We can see that execution times of FIST are always significantly smaller than those of Apriori even if, as it should be noted, FIST generates more information than Apriori: bi-clusters and object lists supporting each association rule are also generated by FIST, bringing to the end-user more information on extracted relationship patterns. With object lists supporting each association rule, the end-user can see precisely the list of objects (human proteins) concerned by the rule.

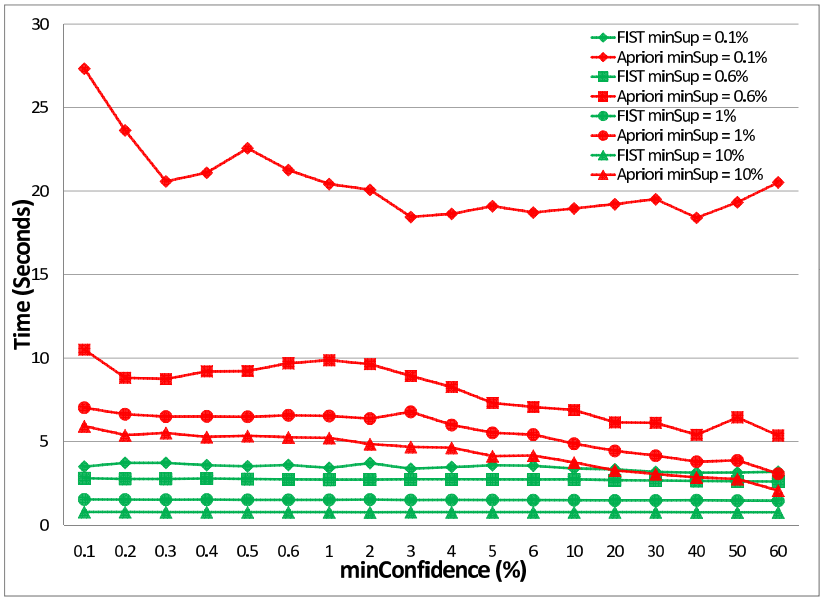

Fig. 9: Execution times.

The number of association rules generated by Apriori and FIST is shown in Figure 10. We can see that FIST reduces this number by a factor up to several tens, allowing the end-user to concentrate on the most relevant rules. In Figure 11, the number of association rules generated by FIST for different *minsupport* and *minconfidence* values is shown. The number of bi-clusters extracted by FIST for *minsupport* values ranging from 0.001 (0.1%) to 0.5 (50%) is shown in Table 1.

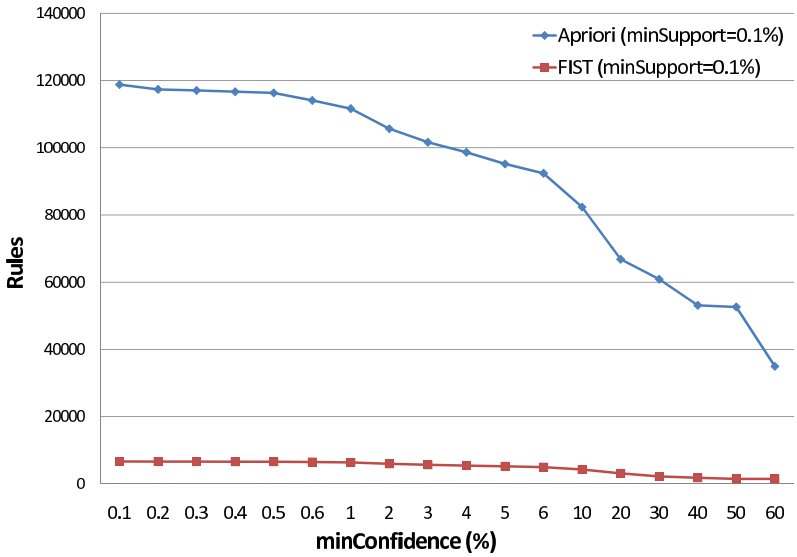

Fig. 10: Number of association rules.

Table 1: Number of bi-clusters extracted by FIST.

minsupport (%)	0.1	0.5	1	5	10	20	30	40	50
Bi-clusters	342	187	104	22	7	2	2	1	1

4.2 Scalability

To assess the scalability of FIST when the number of attributes increases, a second dataset integrating biological annotations and related publications for human proteins was constructed. GO biological annotations of human proteins from the UniProtKB-GOA (GO Annotation@EBI) database were collected from the Gene Ontology web site at http://www.geneontology.org/GO.downloads.annotations.shtml and GO annotations with evidence code TAS, i.e., annotations manually validated by biologists and cited in a published biological reference that are the most reliable biological annotations, were integrated in the data. Publication annotations were collected from the NCBI web site at http://www.ncbi.nlm.nih.gov/sites/entrez and Pubmed and Reactome publications related to the GO biological annotations of human proteins were also integrated in the dataset as new attributes (columns). This dataset contains overall 1149 distinct GO annotations and 2670 distinct publication annotations, and up to 40 GO annotations and 88 publication annotations for each protein. We were unable to run Apriori on this dataset, even for *minsupport* and *minconfidence* values as high as 90% and with a maximum java heap size parameter set to its maximal value, that is 1.5 GB, due to the memory consumption of the approach that requires to identify all frequent itemsets and not only frequent closed itemsets. Execution times of the execution of FIST on this second dataset for *minsupport* values of 0.1 (10%), 0.01 (1%) and 0.006 (0.6%) and for *minconfidence* varying between 0.001 (0.1%) and 0.6 (60%) are depicted in Figure 12. We can see that even for this very large dataset, execution times remain reasonable for all threshold values, ranging from a few seconds to a few minutes. We can also see slight execution time variations for different *minconfidence* values due to other running operating system processes.

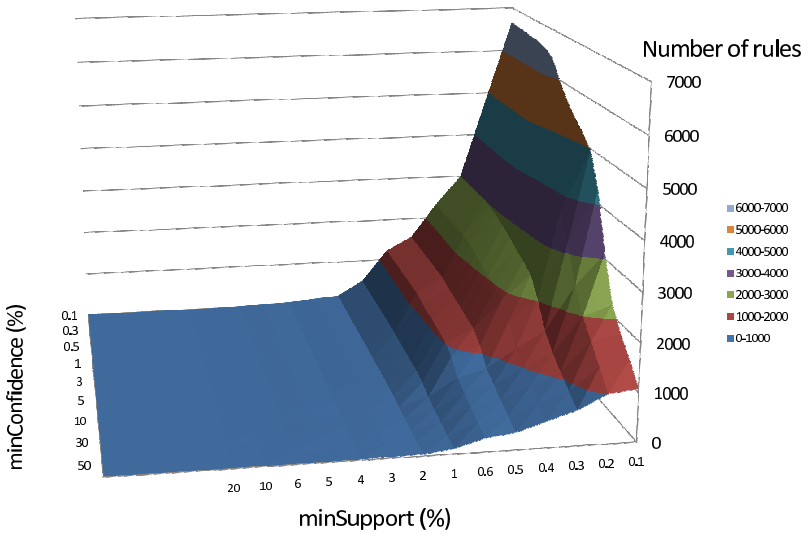

Fig. 11: Number of minimal non-redundant rules.

4.3 Discussion

It is interesting to compare our results to the results obtained by Tastan *et al.* [22], which are the most comprehensive HIV-Human PPI results available to date. We focus on the results generated by FIST for *minsupport* = 0.1% and *minconfidence* = 0.1%, which are the lowest threshold values tested and thus contain maximal information.

For each protein pair interaction predicted in [22], we counted the number of FCIs (and association rules) generated by FIST covering it. Of the 3372 interactions predicted in [22] by a random forest classifier, 895 are covered by at least one FCI generated by FIST. This is 26.5% of their predicted pairs.

Now, the random forest classifier is reported to achieve a mean average precision (MAP) of 0.23 on this problem, meaning that around 23% of the predicted interacting pairs should be expected to be true positives. This is just a little below the percentage of predicted pairs that are "confirmed" by FIST. Since the random forest classifier has little in common with FIST, we believe the two techniques should be regarded as complementary to one another. By the same argument, there are good chances that the interacting pairs predicted by [22] and confirmed by FIST are indeed true interactions.

In general, it appears that proteins pairs predicted by the random forest classifier with a high score are mostly confirmed by a large number of FCIs, although exceptions exist, like the novel high-score predicted pair $\langle \text{env\_gp120}, \text{CALM1} \rangle$, which is not covered by any FCI, indicating perhaps that it is a false positive. Likewise, most low-score predictions are not confirmed by FIST with some exceptions, like $\langle \text{env\_gp120}, \text{EP300} \rangle$ which, however, were known to be indirectly interacting (the human gene is reported in the siRNA screen in [13]). All in all, exceedingly few (28) of the 2100 novel predictions by [22], or 1.3%, are confirmed by FIST. An exhaustive list thereof is given in Table 2, along with the number of covering FCIs, approximate, and exact rules. For rules, two separate counts are provided for rules that have the viral protein in the antecedent (IF part) and in the consequent (THEN part).

On the other hand, FIST finds 451 protein pairs that are covered by at least one FCI among those not included in [22], i.e., for which no explicit indication

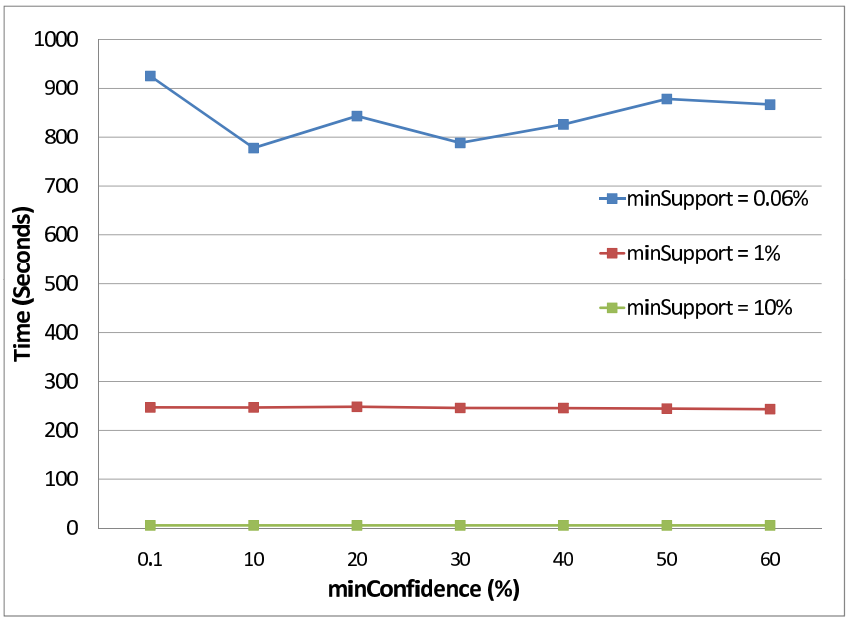

Fig. 12: Execution times for integrated dataset.

of possible interaction was pointed out. This is 2.2% of the pairs not included in [22].

The most covered of these protein pairs is $\langle \text{NEF}, \text{IFNG} \rangle$, covered by 70 FCIs. The NEF protein occurs in the antecedent of 755 approximate rules, in the consequent of 779 approximate rules, in the antecedent of 28 exact rules and in the consequent of 30 exact rules. Lagging far behind this pair, we find the four pairs $\langle \text{TAT}, \text{ACTG1} \rangle$, covered by 45 FCIs. $\langle \text{NEF}, \text{IL6} \rangle$, covered by 45 FCIs, $\langle \text{TAT}, \text{IL2} \rangle$, covered by 44 FCIs, and $\langle \text{TAT}, \text{IL6} \rangle$, covered by 44 FCIs. There are a number of other pairs covered by 35 or fewer FCIs.

The $\langle \text{NEF}, \text{IFNG} \rangle$ pair, to begin by the most covered novel suggestion, although not previously signaled, looks like a promising candidate for further investigation: NEF is the viral negative regulatory factor, associated with the early stages of HIV infection, and the IFNG gene encodes for the interferon-γ protein, an important immune response stimulator and modulator; the suggestion of some kind of relationships between these two proteins may be corroborated by recent research on HIV vaccines [9].

The same negative regulatory factor is involved in the $\langle \text{NEF}, \text{IL6} \rangle$ pair: IL6 is the gene encoding for interleukin-6, a pro-inflammatory cytokine secreted by T-cells and macrophages to stimulate immune response. Indeed, the interaction between NEF and interleukin-6 has been recognized quite early in the study of AIDS [7].

Other two novel pairs suggested by FIST, namely $\langle \text{TAT}, \text{IL2} \rangle$ and $\langle \text{TAT}, \text{IL6} \rangle$, involve interleukins. IL2 is the gene of interleukin-2, a signaling molecule normally produced during an immune response: an antigen binding to a T-cell receptor stimulates the secretion of interleukin-2, which in turn stimulates the growth of antigen-selected cytotoxic T-cells. TAT, for trans-activator of transcription, is a key protein of HIV-1, the first to be transcribed, causing the subsequent massive increase in the transcription levels of the HIV dsRNA. Both interactions are mentioned in the literature: the interaction between TAT and interleukin-6 in [21] and the one between TAT and interleukin-2 in [23].

As for pair $\langle \text{TAT}, \text{ACTG1} \rangle$ suggested by FIST, we are not aware of any work in the literature mentioning it. However, the suggestion does not look completely

Table 2: Novel predicted interacting pairs confirmed by FIST.

HIV-1	Human	#FCI	#approx rules		#exact rules	
			IF	THEN	IF	THEN
ENV_GP160	APOBEC3G	1	0	0	0	0
REV	CXCR4	4	5	5	0	0
ENV_GP120	FURIN	1	0	0	0	0
VPR	MAPK3	34	186	197	7	0
ENV_GP120	PAK1	1	0	0	0	0
TAT	PAK2	2	1	1	0	0
NEF	PIK3R2	8	19	19	0	0
TAT	PPARG	1	0	0	0	0
NEF	PRKCD	34	275	300	17	5
NEF	PRKCG	34	275	300	17	5
NEF	PRKCZ	18	77	83	4	0
TAT	RAF1	3	2	2	0	0
VPR	RAF1	3	2	2	0	0
ENV_GP120	RAN	2	1	1	0	0
TAT	RPA2	4	5	5	0	0
TAT	SDCBP	2	1	1	0	0
GAG_PR55	SHC1	1	0	0	0	0
ENV_GP120	SLC3A2	1	0	0	0	0
TAT	SREBF2	2	1	1	0	0
NEF	STAT5A	4	5	5	0	0
NEF	SUMO1	1	0	0	0	0
TAT	TCEB1	1	0	0	0	0
ENV_GP120	TUBB1	1	0	0	0	0
ENV_GP120	UBB	2	1	1	0	0
NEF	UBB	2	1	1	0	0
TAT	UBE2I	1	0	0	0	0
TAT	WT1	1	0	0	0	0
REV	XRCC5	3	2	2	0	0

implausible, for TAT functions also as a cell-penetrating peptide that acts as a toxine, causing the apoptosis of uninfected T-cells, and the γ-actin 1, encoded for by gene ACTG1, is a component of the cytoskeleton of T-cells.

5 Conclusion

We presented the new FIST algorithm for mining association rules and conceptual bi-clusters that is based on the concept of closure. The main advantages of FIST are that it generates:

- A minimal non-redundant cover for association rules, from which all rules generated by Apriori can be deduced if required, that is much smaller;
- For each association rule, the list of objects (rows) supporting the rule instead of the support (corresponding to the number of such rows) of the rule only;
- The bi-clusters, which are concepts (intension and extension) and form a dual lattice structure defined by inclusion relation;
- For each frequent closed itemset, the generators, which are the minimal sets of properties required to construct the closed itemset.

The method has been validated by applying it to the prediction of HIV-1 protein-human protein interactions. Besides proving faster than traditional association mining methods, the results obtained by FIST confirm and improve the predictions by existing methods, and suggest novel possible interactions to be further investigated.
In the future, we plan to apply the FIST method to data integrating additional information about proteins, like structural and sequential similarities, with protein-protein interactions to improve the results. Indeed, the integration of different kinds of biological information is an essential consideration to fully understand the underlying biological processes [4].

References

1. R. Agrawal, H. Mannila, R. Srikant, H. Toivonen and A.I. Verkamo, *Fast Discovery of Association Rules.* Advances in Knowledge Discovery and Data Mining, pp. 307-328, AAAI/MIT Press, 1996.
2. M.R. Arkin and J.A. Wells, *Small-molecule inhibitors of protein-protein interactions: Progressing towards the dream.* Nature Reviews Drug Discovery, 3:301-317, 2004.
3. Y. Bastide, N. Pasquier, R. Taouil, G. Stumme and L. Lakhal, *Mining minimal non-redundant association rules using frequent closed itemsets.* Proceedings of the International Conference on Computational Logic (CL), LNCS 1861, pp. 972-986, 2000.
4. L. Bell, R. Chowdhary, J.S. Liu, X. Niu and J. Zhang, *Integrated Bio-Entity Network: A system for biological knowledge discovery.* PLoS ONE 6(6):e21474, 2011.
5. A. Ben-Hur and W.S. Noble, *Kernel methods for predicting protein-protein interactions.* Bioinformatics, vol. 21, Suppl 1, pp. 38-46, 2005.
6. A. Ceglar and J. Roddick, *Association mining.* ACM Computing Surveys, 38(2), Article No. 5, 2006.
7. N. Chirmule, N. Oyaizu, C. Saxinger and S. Pahwa, *Nef protein of HIV-1 has B-cell stimulatory activity.* AIDS, 8(6):733–4, 1994.
8. W. Fu, B.E. Sanders-Beer, K.S. Katz, D.R. Maglott, K.D. Pruitt and R.G. Ptak, *Human immunodeficiency virus type 1, human protein interaction database at NCBI.* Nucleic Acids Research, 37:417-422, 2009.
9. H. Gahery, S. Figueiredo, C. Texier, *et al.*, *HLA-DR-restricted peptides identified in the Nef protein can induce HIV type 1-specific IL-2/IFN-gamma-secreting CD4+ and CD4+ /CD8+ T cells in humans after lipopeptide vaccination.* AIDS Research and Human Retroviruses, 23(3):427-37, 2007.
10. B. Ganter and R. Wille, *Formal Concept Analysis: Mathematical foundations.* Springer, 1999.
11. T. Hamrouni, S. Ben Yahia and E. Mephu Nguifo, *Succinct system of minimal generators: A thorough study, limitations and new definitions.* Proceedings of the International Conference on Concept Lattices and their Applications (CLA), LNCS 4923, pp. 80-95, 2006.
12. R. Jansen, H. Yu, D. Greenbaum, Y. Kluger, N.J. Krogan, S. Chung, A. Emili, M. Snyder, J.F. Greenblatt and M. Gerstein, *A Bayesian networks approach for predicting protein-protein interactions from genomic data.* Science, 302:449-453, 2003.
13. R. Konig, Y. Zhou, D. Elleder, *et al.*, *Global analysis of host-pathogen interactions that regulate early-stage HIV-1 replication.* Cell, 135:49-60, 2008.
14. N. Lin, B. Wu, R. Jansen, M. Gerstein and H. Zhao, *Information assessment on predicting protein-protein interactions.* BMC Bioinformatics, 5(154), 2004.

15. S.C. Madeira and A.L. Oliveira, *Biclustering algorithms for biological data analysis: A survey.* IEEE/ACM Transactions on Computational Biology and Bioinformatics, 1:24-45, IEEE Computer Society, 2004.
16. A. Mukhopadhyay, U. Maulik, S. Bandyopadhyay and R. Eils, *Mining association rules from HIV-Human protein interaction.* Proceedings of the International Conference on Systems in Medicine and Biology (ICSMB), pp. 344-348, 2010.
17. N. Pasquier, Y. Bastide, R. Taouil and L. Lakhal, *Efficient mining of association rules using closed itemset lattices.* Information Systems, 24(1):25-46, 1999.
18. N. Pasquier, R. Taouil, Y. Bastide, G. Stumme and L. Lakhal, *Generating a condensed representation for association rules.* Journal of Intelligent Information Systems, 24(1):29-60, 2005.
19. R.G. Ptak, W. Fu, B.E. Sanders-Beer, J.E. Dickerson, J.W. Pinney, D.L. Robertson, M.N. Rozanov, K.S. Katz, D.R. Maglott, K.D. Pruitt, C.W. Dieffenbach, *Cataloguing the HIV-1 Human protein interaction network.* AIDS Research and Human Retroviruses, 4(12):1497-1502, 2008.
20. Y. Qi, J. Klein-Seetharaman and Z. Bar-Joseph, *A mixture of feature experts approach for protein-protein interaction prediction.* BMC Bioinformatics, vol. 8, Suppl 10, 2007.
21. G. Scala *et al.*, *The expression of the interleukin 6 gene is induced by the human immunodeficiency virus 1 TAT protein.* Journal of Experimental Medicine, 179(3):961–971, 1994.
22. O. Tastan, Y. Qi, J. Carbonell and J. Klein-Seetharaman, *Prediction of interactions between HIV-1 and human proteins by information integration.* Proceedings of the Pacific Symposium on Biocomputing (PSB), pp. 516-527, 2009.
23. M.O. Westendorp, M. Li-Weber, R.W. Frank and P.H. Krammer, *Human immunodeficiency virus type 1 Tat upregulates interleukin-2 secretion in activated T cells.* Journal of Virology, 68(7):4177–4185, 1994.
24. Y. Yamanishi, J.P. Vert and M. Kanehisa, *Protein network inference from multiple genomic data: A supervised approach.* Bioinformatics, 20:363-370, 2004.
25. Y. Yang, G. Webb and X. Wu, *Discretization Methods.* Data Mining and Knowledge Discovery Handbook, Part 1, pp. 101-116, 2010.
26. M.J. Zaki, *Generating non-redundant association rules.* Proceedings of the ACM SIGKDD Conference on Knowledge Discovery from Data (KDD), pp. 34-43, 2000.
27. M.J. Zaki, *Mining non-redundant association rules.* Data Mining and Knowledge Discovery, 23(9):223-248, 2004.
28. L. Zhang, S. Wong, O. King and F. Roth, *Predicting co-complexed protein pairs using genomic and proteomic data integration.* BMC Bioinformatics, 5(38), 2004.