Balázs Szalkai* and Vince Grolmusz*

SCARF: a biomedical association rule finding webserver

https://doi.org/10.1515/jib-2021-0035
Received October 17, 2021; accepted January 10, 2022; published online February 4, 2022

Abstract: The analysis of enormous datasets with missing data entries is a standard task in biological and medical data processing. Large-scale, multi-institution clinical studies are the typical examples of such datasets. These sets make possible the search for multi-parametric relations since from the plenty of the data one is likely to find a satisfying number of subjects with the required parameter ensembles. Specifically, finding combinatorial biomarkers for some given condition also needs a very large dataset to analyze. For fast and automatic multi-parametric relation discovery association-rule finding tools are used for more than two decades in the data-mining community. Here we present the SCARF webserver for *generalized* association rule mining. Association rules are of the form: a AND b AND ... AND $x \rightarrow y$, meaning that the presence of properties a AND b AND ... AND x implies property y; our algorithm finds generalized association rules, since it also finds logical disjunctions (i.e., ORs) at the left-hand side, allowing the discovery of more complex rules in a more compressed form in the database. This feature also helps reducing the typically very large result-tables of such studies, since allowing ORs in the left-hand side of a single rule could include dozens of classical rules. The capabilities of the SCARF algorithm were demonstrated in mining the Alzheimer's database of the Coalition Against Major Diseases (CAMD) in our recent publication (Archives of Gerontology and Geriatrics Vol. 73, pp. 300–307, 2017). Here we describe the webserver implementation of the algorithm.

Keywords: association rules; big data; data mining.

1 Introduction and motivation

An enormous amount of data is generated every day in biological experiments and clinical investigations. These data may yield deep and very useful relations between parameters of interest if analyzed properly. Data mining techniques [1, 2], which were first used in analyzing commercial transactions, are increasingly applied for biomedical data sources today [3–5]. Association rule mining is one of the areas that anticipated a massive development, beginning with its introduction in [6, 7]. Association rules are automatically found patterns in large databases, where, say, each human patient has a number of attributes or parameter values, and the association rules describe implication-like relations between these attributes, like this one:

(high cholesterol level) AND (high blood pressure) → (heart disease).

These rules have a left-hand side (abbreviated by LHS), left from the → symbol, and a right-hand side (RHS), right from the → symbol. There are several quality measures of these rules, we mention here only the three most important ones:

***Corresponding authors: Balázs Szalkai**, PIT Bioinformatics Group, Eötvös University, H-1117 Budapest, Hungary, E-mail: szalkai@pitgroup.org; and **Vince Grolmusz**, PIT Bioinformatics Group, Eötvös University, H-1117 Budapest, Hungary; and Uratim Ltd., H-1118 Budapest, Hungary, E-mail: grolmusz@pitgroup.org

Open Access. © 2022 Balázs Szalkai and Vince Grolmusz published by De Gruyter. This work is licensed under the Creative Commons Attribution 4.0 International License.

- Support: The number of data items (e.g., patients), where both the LHS and RHS are true. The LHS support is the number of the data items where LHS is true.
- Confidence: The value of the Support, divided by the LHS support. In our example it describes the fraction of patients with high cholesterol AND high blood pressure, also having heart disease.
- Lift: Describes the relative level of dependence between the LHS and the RHS, compared to the hypothesis that the LHS and the RHS are occurring independently; with probabilities: Pr(LHS AND RHS)/(Pr(LHS) · Pr(RHS)). The lift is 1 if the LHS does not affect the RHS. The lift is greater than 1 if the LHS increase the probability of the occurrence of the RHS.

In *association rule mining* the association rules with pre-defined minimum support, confidence and lift values need to be found [6, 7].

1.1 The simple combinatorial association rule finding (SCARF) webserver

Here we present the SCARF webserver that computes *generalized* association rules, where the LHS can also contain disjunctions (i.e., ORs), not only ANDs, as in the classical association rules. SCARF also computes some other statistical parameters of the rules. SCARF was successfully applied in mining the large CAMD Alzheimer's database [8], described in our work [9]. Because of the already successfully presented application of SCARF in [9], we do not present example applications in this note.

The computed generalized association rules, with conjunctions and disjunctions in its LHS, have two remarkable properties: (i) any Boolean function can be represented as the ANDs of ORs of the variables and the negations of the variables, therefore, these generalized association rules are universal in describing Boolean functions, and (ii) short generalized association rules are capable of describing many non-generalized association rules in one formula, since, e.g., the LHS (a ORb) AND(c ORd) AND(e ORf) is equivalent to the OR of eight ternary conjunctions; consequently, this generalized LHS compresses the LHS of eight non-generalized rules.

1.2 Comparison with earlier works

There are several available webservers with association rule finding capabilities. For example, a recent publication describes the WebApriori server [10], a simple, but very useful implementation of the *apriori* algorithm [6, 7]. Another recent publication describes ARM [11], an interactive association rule miner with lots of selectable features. Both of these webservers are capable of finding association rules as described in [6, 7], but they lack the capability of our SCARF of finding generalized association rules. We repeat that the a single generalized association rule can be used for succinct and compact description of a large number of non-generalized, or simple association rules. Therefore, a large number of valid association rules can be presented in a few human-readable rules, instead of dozens of simple rules.

2 Materials and methods

The command-line SCARF program can be downloaded from http://pitgroup.org/apps/scarf/downloads/scarf.zip. This program takes a data table, a rule pattern and several numerical parameters (minimum confidence, etc.) as input, and produces association rules which have the given, pre-defined pattern.

The data table must be a comma or semicolon separated CSV file. The first line is the header, containing the column (or attribute or parameter) names. The subsequent lines are each a record in the data table, containing single ASCII characters in the cells. Empty cells are considered as N/A. Cells more than one character long are truncated to the very first character. Different characters represent different values.

The rule pattern is a logical expression, with blanks instead of variables. For example, □ AND (□ OR □) → □ is a valid pattern. The allowed tokens in a valid rule pattern are blanks (□), parentheses, operators (AND/OR) and the implies sign (→). For the sake of simplicity, the right-hand side of a rule must always consist of a single blank.

SCARF examines all the sensible possibilities for filling the blanks in the rule pattern with elementary equalities. An elementary equality states that a database column (attribute) equals to one of some given values. This way we get the rule candidates. For example, age $= AB$ AND (bread $= y$ OR onions $= n$) $\rightarrow$ butter $= y$ is a possible candidate for the pattern mentioned above. It states that if someone is in the age group A or B, and they buy bread or do not buy onions, then they will buy butter.

Numerical values like support, confidence, lift and leverage [1] are assigned to all the candidates, then compared against the corresponding parameters which were supplied to the program, and rules which fail the test are discarded.

A significant time would be required if SCARF were to do an exhaustive search. Instead, branches of the rule tree are pruned in advance. If the first few blanks are filled a certain way, then in many cases we can deduce that we cannot get good enough rules by carrying on, no matter how we assign attributes to the remaining blanks. In this case, a backtrack happens earlier than it would if we tried to fill in the remaining blanks as well. This speeds up operation considerably.

Another improvement is the pooling of bit operations. A rule is a logical combination of some elementary equalities which are either true or false. So when we evaluate a rule, we have to perform logical operations on bits. We can speed up this part if we process multiple rows simultaneously. If the word-length of a computer is 64 bits, then we can process 64 rows in parallel, yielding a 64-fold improvement in running time.

The majority of processors have multiple cores today. If a program can utilize all the cores efficiently, it can achieve an n-fold speedup in the ideal case, where n is the number of CPU cores. A commonly used approach is organizing a job pool and launch n worker threads, which will be distributed to separate cores by the operation system. The worker threads will then pick jobs from the queue, process them and write back the result to another shared pool. If there are a lot of jobs and they are similarly computationally intensive, then this approach results in almost ideal resource distribution. That is, the threads will finish about in the same time and do not have to wait for each other in the end.

Since the right-hand side of the rule consists of a single blank, and we want to examine all the possibilities, each possible assignment of this blank can be regarded as a job for the threads. This means that SCARF creates as many jobs for the threads as the number of possible assignments for the blank on the right-hand side.

3 Implementation and usage

The SCARF webserver, available at http://pitgroup.org/scarf/, allows uploading data tables and setting parameters on a web form. The job is then run on our high-performance 16-core server, the typical running time is several minutes. After completion, the user is notified in an email and can view the results again on a web interface, both in TXT (unformatted) and XML (formatted). The first screen (Figure 1) is for uploading the data table. There is a small example data table which can be used if the user just wants to test the service. It is possible to upload a custom data table which cannot exceed 3 MB.

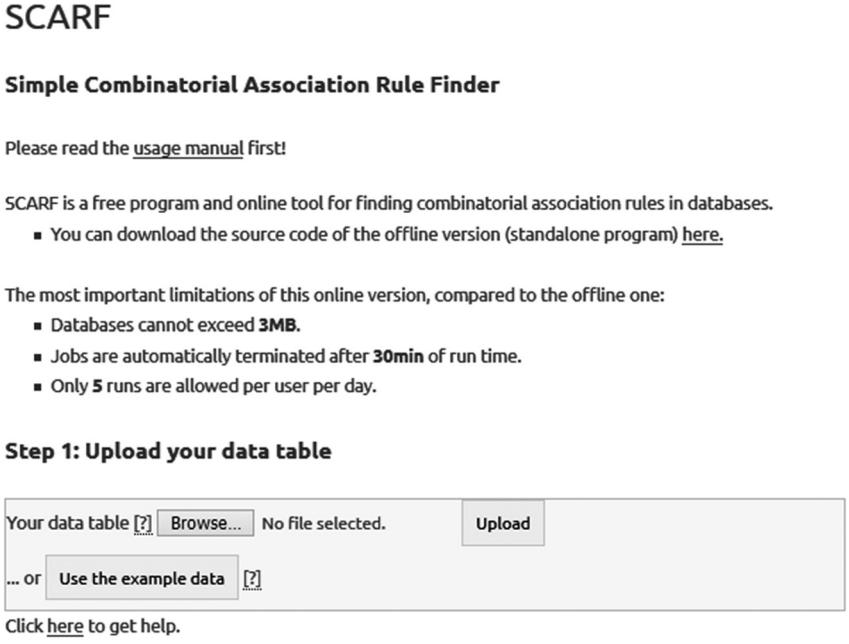

Figure 1: The input screen of the SCARF webserver.

The next screen (Figure 2) allows specifying the rule pattern, setting the parameters and constraining which columns (attributes) should appear on the left and right-hand side, respectively. It is advised to set stricter parameters to reduce the number of rules examined. The user can then schedule the job after checking the input once more. Note that the specified rule structure is the most complex the webserver may return; SCARF tries to find the simplest rules possible in the following sense: it deletes elementary clauses from the LHS as long as the lift of the rule did not decrease by more than 2%. In other words, SCARF will not necessarily return such complex rules that we specified as input if simpler rules have higher lift (i.e., quality) (Figure 3).

The job is allowed 30 min of running time on our server. In our experiments, this was more than enough for a data table with 170 columns and 6100 rows. A user may submit five jobs daily, allowing two and a half hours of server-time per day. If a user would like to overcome these limitations, it is suggested downloading and running the off-line version.

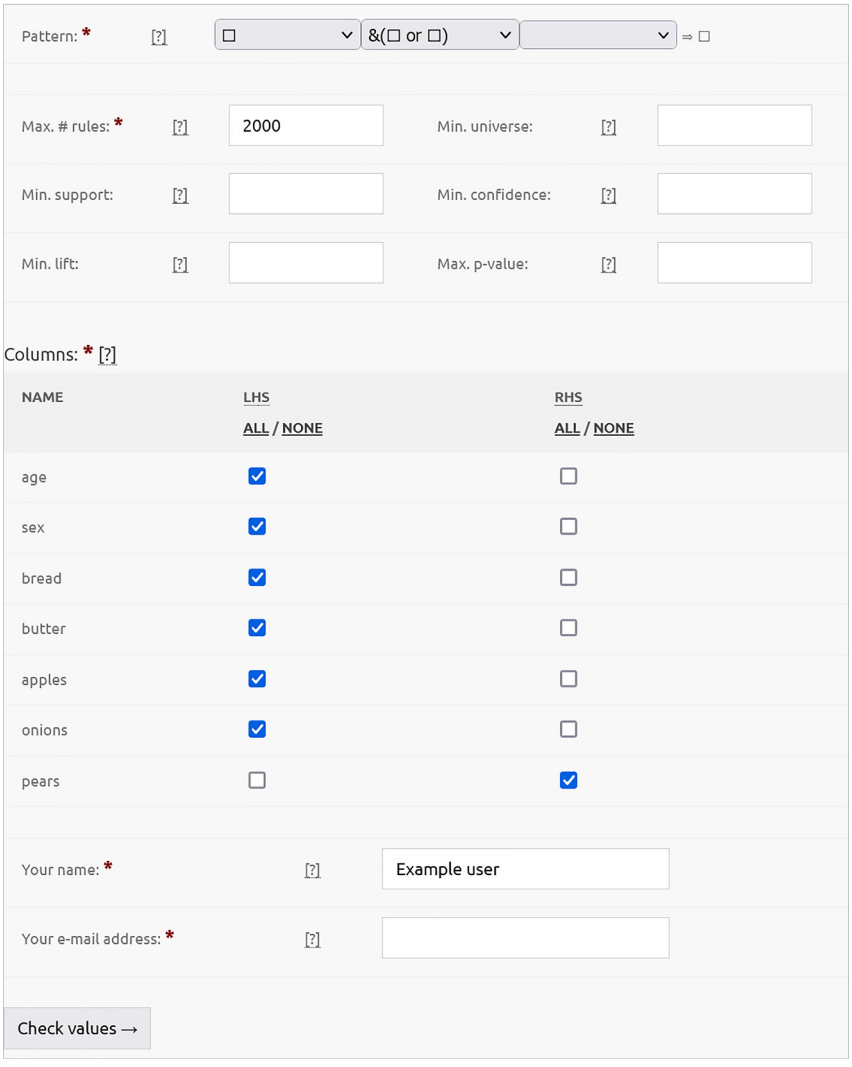

Figure 2: The parameter screen of the SCARF webserver.

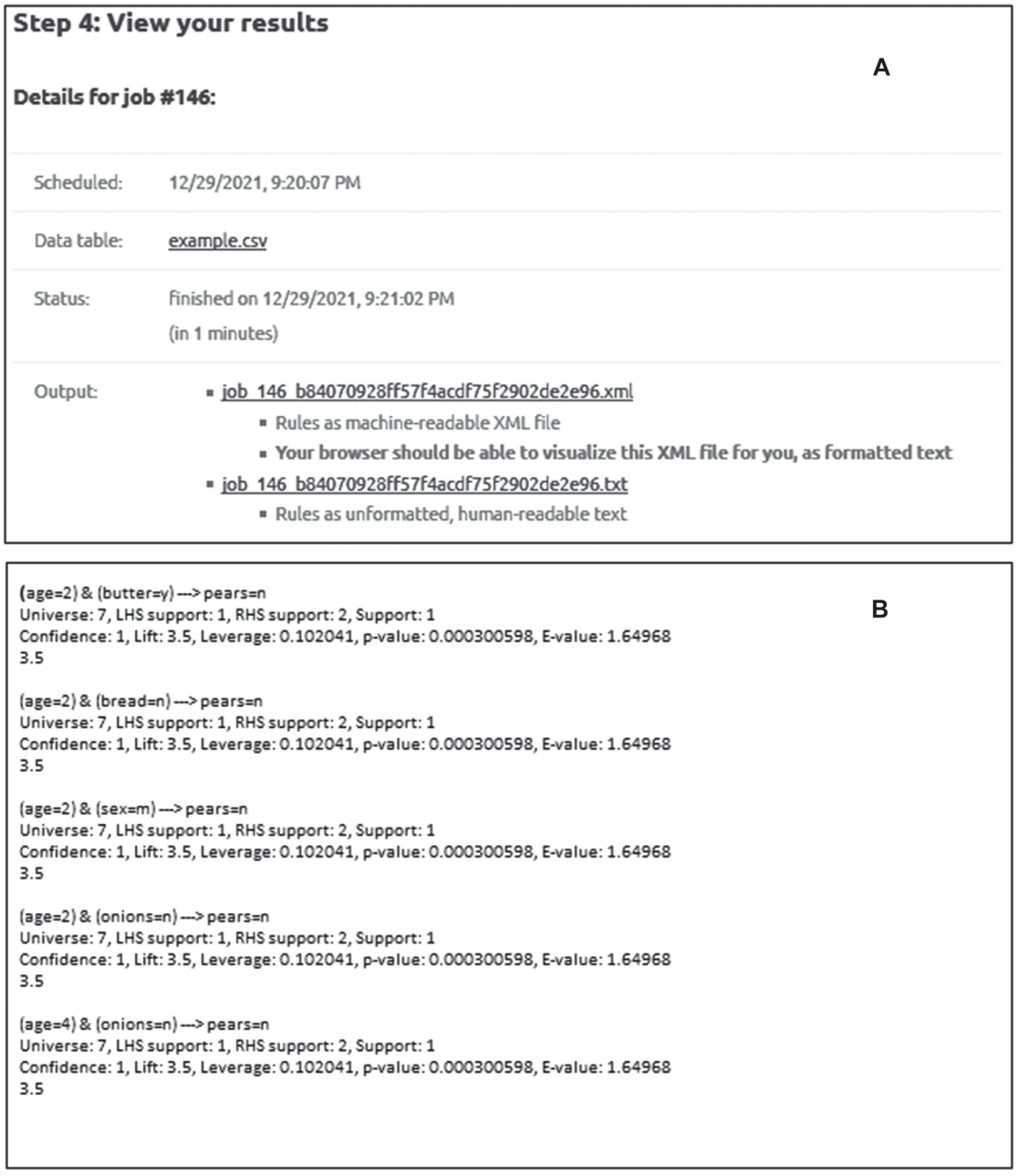

Figure 3: The output screen of the webserver (panel A) and a partial output with five discovered rules found in the example dataset (panel B).

Author contribution: BS and VG wrote the main manuscript text; BS prepared the SCARF webserver and developed the SCARF program; VG initiated the study.

Research funding: VG was partially supported by the VEKOP-2.3.2-16-2017-00 014 and the EFOP-3.6.3-VEKOP-16-2017-00 002 programs, supported by the European Union and the State of Hungary, co-financed by the European Regional Development Fund, and by the NKFI-127 909 grant of the National Research, Development and Innovation Office of Hungary.

Conflict of interest statement: The authors declare no conflicts of interest regarding this article.

Availability and implementation: The stand-alone SCARF (Simple Combinatorial Association Rule Finder) program is written in C++, and is downloadable from http://pitgroup.org/apps/scarf/downloads/scarf.zip. The webserver can be found at the address http://pitgroup.org/scarf/.

References

1. Han J, Kamber M. Data Mining: Concepts and Techniques. San Francisco: Morgan Kaufmann Publishers; 2000.
2. Hand DJ, Mannila H, Smyth P. Principles of Data Mining. Cambridge, MA: MIT Press; 2001.
3. Ivan G, Szabadka Z, Grolmusz V. Being a binding site: characterizing residue composition of binding sites on proteins. Bioinformation 2007;2:216—21.
4. Ivan G, Szabadka Z, Ordog R, Grolmusz V, Naray-Szabo G. Four spatial points that define enzyme families. Biochem Biophys Res Commun 2009;383:417—20.
5. Ivan G, Szabadka Z, Grolmusz V. A hybrid clustering of protein binding sites. FEBS J 2010;277:1494—502.
6. Agrawal R, Imielinski T, Swami AN. Mining association rules between sets of items in large databases. In: Buneman P, Jajodia S, editors Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data, Washington, D.C., May 26—28, 1993. ACM Press; 1993:207—16 pp.
7. Agrawal R, Srikant R. Fast algorithms for mining association rules in large databases. In: Bocca JB, Jarke M, Zaniolo C, editors Proc. of the 20th International Conference on Very Large Data Bases (VLDB '94), vol 1215. Kaufmann Publishers Inc.; 1994:487—99 pp.
8. Romero K, de Mars M, Frank D, Anthony M, Neville J, Kirby L, et al. The coalition against major diseases: developing tools for an integrated drug development process for alzheimer's and Parkinson's diseases. Clin Pharmacol Ther 2009;86:365—7.
9. Szalkai B, Grolmusz VK, Grolmusz VI, Diseases CAM. Identifying combinatorial biomarkers by association rule mining in the CAMD Alzheimer's database. Arch Gerontol Geriatr 2017;73:300—7.
10. Malliaridis K, Ougiaroglou S, Dervos DA. WebApriori: a web application for association rules mining. In: Kumar V, Troussas C, editors. Intelligent Tutoring Systems. Cham: Springer International Publishing; 2020:371—7 pp.
11. Percin I, Yagin FH, Guldogan E, Yologlu S. ARM: an interactive web software for association rules mining and an application in medicine. In: 2019 International Artificial Intelligence and Data Processing Symposium (IDAP); 2019:1—5 pp.