

FCA4J: A Java Library for Relational Concept Analysis and Formal Concept Analysis

Alain Gutierrez, Marianne Huchard, Pierre Martin

▶ **To cite this version:**

Alain Gutierrez, Marianne Huchard, Pierre Martin. FCA4J: A Java Library for Relational Concept Analysis and Formal Concept Analysis. ETAFCA 2022 - ExistingTools and Applications for Formal Conceptual Analysis Workshop@CLA2022, Jun 2022, Tallinn, Estonia. pp.207-212. hal-03702242

HAL Id: hal-03702242
https://hal.science/hal-03702242

Submitted on 22 Jun 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

FCA4J: A Java Library for Relational Concept Analysis and Formal Concept Analysis*

Alain Gutierrez[1], Marianne Huchard[1], and Pierre Martin[2]

[1] LIRMM, Univ Montpellier, CNRS, Montpellier, France
{alain.gutierrez@lirmm.fr,marianne.huchard@lirmm.fr}
[2] CIRAD, UPR AIDA, F-34398 Montpellier, France
AIDA, Univ Montpellier, CIRAD, Montpellier, France
pierre.martin@cirad.fr

Abstract. Formal Concept Analysis (FCA) and its extensions have shown their efficacy and relevance in various application domains. We recently conducted a set of experiments using Relational Concept Analysis and FCA in the domain of agro-ecology. This motivated the development of a library named FCA4J, which includes Java implementation of algorithms to build structures and implications and to manage data, in particular relational context families. This paper presents the main features of FCA4J, its ecosystem, and a few use cases.

Keywords: Formal Concept Analysis · Relational Concept Analysis · software library · Java .

1 Introduction

The development of the Knomana knowledge-based system [18], dealing with pesticidal plant use, required to use Relational Concept Analysis (RCA) to consider the conforming entity-relationship model [7] of its data model with the aim to perform knowledge discovery and Data mining [4]. The first results obtained using RCAexplore were very relevant [11], but led to raising new questions to be tackled through the computing of implication sets. As RCAexplore was not proposing such calculations, and no software was computing Duquennes-Guigues bases of implications [6] on a relational dataset, we initiated the development of a library named FCA4J. The latter was developed in Java to be cross-platform, and to facilitate its distribution. Since then, various types of analysis, including Formal Concept Analysis (FCA) computations, have been produced for Knomana [12, 17], that went enriching this library. FCA4J is actually available[3] and distributed with the licence BSD3.

The objective of this paper is to introduce FCA4J. Section 2 presents the existing libraries on FCA. The functionalities of FCA4J are presented in Section 3, and examples of usage in Section 4. Section 5 concludes and provides perspectives on FCA4J enrichment.

* This work was partly supported by the ANR SmartFCA project, grant ANR-21-CE23-0023 of the French Agence Nationale de la Recherche.

[3] https://www.lirmm.fr/fca4j/

© Pablo Cordero, Ondrej Krídlo (Eds.): CLA 2022, pp. 209–214.

2 Existing FCA libraries

Multiple concrete applications have been developed using FCA [5, 14, 2]. Most of the software that supports these applications are listed in [15], including the following libraries.

Colibri and qdfca[4] aim both to compute a lattice from a formal context. Colibri was initially developed in C (to be compiled before being executed on a computer), but a Java release[5] was further developed to facilitate its distribution. To run qdfca, it is required the standard Ruby library. Galicia v3 [6] is a java library distributed as a jar file, which computes various FCA structures and various rule bases. fcaR [3] is a R package that also computes fuzzy FCA. FCAlib[7] is a Java library distributed as a jar file. FCAlib includes basic FCA algorithms such as implicational closure and implements an attribute exploration algorithm that can be customized according to expert needs. FcaStone[8] aims to convert between the file formats of commonly-used FCA tools and between FCA formats and other graphical formats. To run, it requires the PERL files. The library presented above considers a formal context exclusively in the computation. Both Galicia v1 [9] and RCAexplore[10] consider RCA. They are developed in Java and distributed as a jar file. They aim to compute a family of relational conceptual structures.

3 FCA4J Features

Figure 1 shows the FCA4J features organized along six high-level categories: (1) input file format, (2) output file format, (3) pre-processing commands, (4) computed structures command, (5) Java data structure used to compute structures, and (6) Duquennes-Guigues Basis of Implication (DGBI) computation parameters. This section introduces each of these categories.

Input and output formats Several external formats are supported by FCA4J to facilitate interoperability with other software. As FCA input, it supports XML data files, i.e. XML format from Galicia v3 and CEX from ConExp, and text files, such as SLF (a HTK standard lattice format), CXT (Burmeisters ConImp), and CSV. For the latter, it allows data provided as Boolean and multi-valued attributes. As FCA output, it generates DOT, XML, JSON, TXT, and CSV format files. The conversion of a file to another format can be realized using the command Convert. Regarding RCA, it supports RCFT (the relational context family file format of RCAexplore, RCFTGZ (a compressed released of RCFT), and RCFAL (a JSON file containing an adjacency list).

[4] https://rubygems.org/gems/qdfca
[5] https://code.google.com/archive/p/colibri-java/
[6] v3 at https://sourceforge.net/projects/galicia/files/galicia/
[7] https://github.com/julianmendez/fcalib
[8] https://github.com/upriss/fcastone
[9] v1 at http://www.iro.umontreal.ca/ galicia/
[10] http://dataqual.engees.unistra.fr/logiciels/rcaExplore

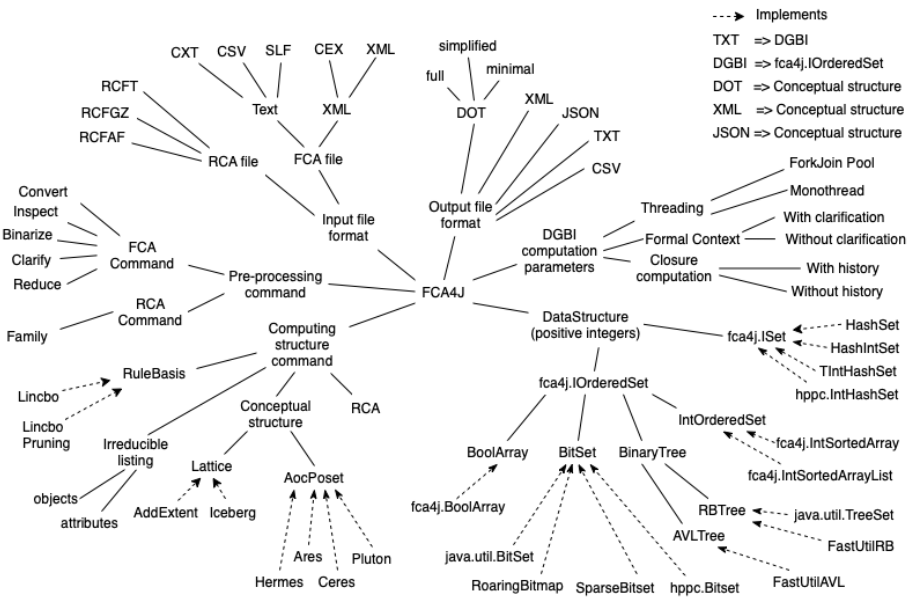


Fig. 1. An overview of all features of FCA4J

Pre-processing command FCA4J provides advanced commands required for the data pre-processing. Applied to a formal context, the command *Inspect* displays information (size, density, etc.), *Binarize* applies scaling operations to binarizing multi-valued data, and *Clarify* eliminates any duplicated objects and attributes. *Reduce* keeps only irreducible objects and attributes. *Family*, applied to a relational context family, enables the creation and the management of a relational context family using a set of formal contexts and relations.

Computing structure command FCA4J computes various structures. The command *Lattice* computes the concept lattice (*AddExtent option*) or an Iceberg lattice (*Iceberg option*). Command *AocPoset* builds the subset of the concept lattice considering only concepts introducing an object or an attribute. Various algorithms are available, i.e. Hermes, Ares, Ceres, and Pluton. *Irreducible* lists irreducible objects and attributes. *RuleBasis* computes the canonical basis of implications (Duquenne-Guigues). Two algorithms are available, Lincbo [9] and LinCbO with pruning [10, 8]. *RCA* applies Relational Concept Analysis and creates a family of conceptual structures (AOCposets, Iceberg lattices or concept lattices) from a relational context family.

Java data structures FCA4J integrates most of them through a design pattern in order to enable the user to select the most appropriate one to perform FCA or RCA computation. The available data structure types are BoolArray, BitSet, AVLtree, RBTree, IntOrderedSet, and fca4j.ISet. These data structure types present various implementations, such as RBTree, which uses the

Java.util.TreeSet collection or the FastUtil framework[11]. Other data structures are provided by Koloboke[12] and hppc[13] collections.

DGBI computation parameters Various configurations of the DGBI computation are available. The formal context can be clarified, the closure computation can consider the history, and the calculation can be realized using a monothread or multithreads (fork Join Pool method).

4 FCA4J ecosystem and use case

The FCA4J website fully describes the commands presented in Section 3. In addition, its *Getting started* page provides a dataset and a set of command lines to produce artifacts, e.g. concept lattice, Iceberg50, AOCposet, Duquenne-Guigues set of implications. This section presents the actual ecosystem of FCA4J and then a few use cases.

Ecosystem FCA4J does not propose any viewer. To navigate within a conceptual structure family represented as a set of lattices, we developed the online visualiser RCAviz[14] [13]. Figure 2 presents the process that combines FCA4J and RCAviz. The RCFT file containing the data is provided as input to FCA4J that generates the JSON file (the input file format required by RCAviz) using the command *RCA* via the terminal. The user has then to open this file with RCAviz to navigate within the lattices and from one to another. FCA4J is not managed by RCAviz in order to keep these applications independent. Moreover, FCA4J was integrated into Cogui[15] platform to reason on knowledge through the representation of the inference rules, expressed as predicates sets, as an AOC-poset. The aim is to perform a lazy evaluation of the predicates, by favouring the easiest to compute.

Use cases FCA4J is currently used by students of the Software engineering Master at Montpellier University in several laboratory work as the practical support to learn FCA and RCA, to study Java class refactoring (using FCA) and the UML class model refactoring (using RCA).

FCA4J is also used by students during their master internship applied to Knomana. For instance, L. Mahrach used FCA4J to explore the combination of RCA and DGBI with the aim of rendering knowledge suitable to experts [12]. The results showed that the way of splitting data between tables could have a positive or negative impact on the implications' readability by the experts. Moreover, J. Saoud identified various types of knowledge elements and patterns that constitute the implications from the DGBI [17]. The results show that a

[11] https://fastutil.di.unimi.it
[12] https://github.com/leventov/Koloboke
[13] https://github.com/carrotsearch/hppc
[14] https://rcaviz.lirmm.fr
[15] https://www.lirmm.fr/cogui/

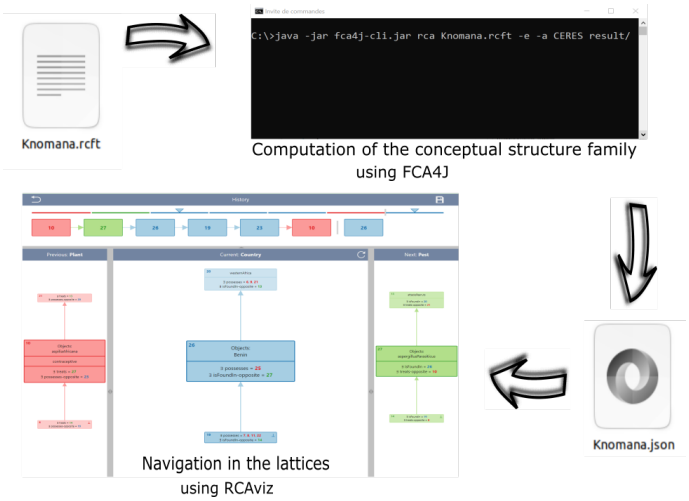


Fig. 2. Process for navigating within a Relational context family, provided as a RCFT format file, using FCA4J and RCAviz.

post-process can be conducted to remove tacit knowledge elements from implications, to improve their readability by the experts. An on-going work by N. Saab evaluates the use of FCA4J to detect and correct various types of anomalies in the Knomana dataset and the associated sets of implications [16].

5 Conclusion

Two releases of FCA4J are currently distributed as a jar file. One contains all the elements presented in this document. The light release does not contain the libraries FastUtil, hppc, and Koloboke as they are very large. This removal reduces the number of data structures available in FCA4J as well as the size of the distributed jar file, and thus facilitates the integration of FCA4J in an Apache Maven repository.

In the context of a Master student project, we are currently investigating the use of FCA4J commands in a Jupyter notebook [1]. One investigated scenario is to design a workflow involving tools such as Scikitlearn[16], weka[17] or Orange[18], that includes association rule computation and classification of the rules in a concept lattice based on the premise attributes.

Acknowledgements The authors would like to thank Huaxi (Yulin) Zhang for her co-supervision of the Master project [1]. Part of this work was supported by the French National Research Agency under the Investments for the Future Program (ANR-16-CONV-0004), and through the project SmartFCA (ANR-21-CE23-0023).

[16] https://scikit-learn.org/stable/
[17] https://www.cs.waikato.ac.nz/ml/weka/
[18] https://orangedatamining.com/

References

1. Bros, T., Haoulani, R., Alla, J.C., Seoudi, N.: FCA Notebook - Master 1st year - Research initiation report. Master's thesis, Université de Montpellier, France (May 2022), in French, supervision: Huaxi (Yulin) Zhang, Marianne Huchard
2. Cordero, P., Enciso, M., López, D., Mora, A.: A conversational recommender system for diagnosis using fuzzy rules. Exp. Syst. App. **154**, 113449 (2020)
3. Cordero, P., Enciso, M., López-Rodríguez, D., , Ángel Mora: fcaR, Formal Concept Analysis with R. The R Journal p. Accepted (2022)
4. Fayyad, U.M., Piatetsky-Shapiro, G., Smyth, P.: The KDD process for extracting useful knowledge from volumes of data. Commun. ACM **39**(11), 27–34 (1996)
5. Ferré, S., Kaytoue, M., Huchard, M., Kuznetsov, S.O., Napoli, A.: Formal concept analysis: From Knowledge discovery to Knowledge processing. In: A Guided Tour of Artificial Intelligence Research, vol. II. Springer (2018)
6. Guigues, J.L., Duquenne, V.: Famille minimale d'implications informatives résultant d'un tableau de données binaires. Math. et Sci. Hum. **24**(95), 5–18 (1986)
7. Hacene, M.R., Huchard, M., Napoli, A., Valtchev, P.: Relational concept analysis: mining concept lattices from multi-relational data. Ann. Math. Artif. Intell. **67**(1), 81–108 (2013)
8. Janostik, R., Konecny, J., Krajca, P.: LinCbO: Fast algorithm for computation of the Duquenne-Guigues basis. Inf. Sci. **572**, 223–240 (2021)
9. Janostik, R., Konecny, J., Krajča, P.: LinCbO: Fast algorithm for computation of the Duquenne-Guigues basis. arXiv:2011.04928 (2020)
10. Janostik, R., Konecny, J., Krajča, P.: Pruning techniques in LinCbO for computation of the Duquenne-Guigues basis. In: The 16th Int. Conf. on Formal Concept Analysis (ICFCA) 2021. No. 12733 in LNCS/LNAI, Springer (2021)
11. Keip, P., Gutierrez, A., Huchard, M., Le Ber, F., Sarter, S., Silvie, P., Martin, P.: Effects of Input Data Formalisation in Relational Concept Analysis for a Data Model with a Ternary Relation. In: Formal Concept Analysis. pp. 191–207. LNCS, Springer International Publishing (2019)
12. Mahrach, L., Gutierrez, A., Huchard, M., Keip, P., Marnotte, P., Silvie, P., Martin, P.: Combining implications and conceptual analysis to learn from a pesticidal plant knowledge base. LNCS, vol. 12879, pp. 57–72. Springer (2021)
13. Muller, E., Huchard, M., Martin, P., Poncelet, P., Sallaberry, A.: RCAviz: Visualizing and Exploring Relational Conceptual Structures. In: Proc. of CLA 2022 - 16th Int. Conf. on Concept Lattices and Their Applications (Jun 2022)
14. Poelmans, J., Ignatov, D.I., Kuznetsov, S.O., Dedene, G.: Formal concept analysis in knowledge processing: A survey on applications. Expert Syst. Appl. **40**(16), 6538–6560 (2013)
15. Priss, U.: Formal Concept Analysis Home page: FCA Software. https://upriss.github.io/fca/fcasoftware.html, [Last access 30-April-2022]
16. Saab, N., Huchard, M., Martin, P.: Evaluating Formal Concept Analysis Software for Anomaly Detection and Correction. In: ETAFCA 2022 : Existing Tools and Applications for Formal Concept Analysis, Workshop @ CLA 2022
17. Saoud, J., Gutierrez, A., Huchard, M., Marnotte, P., Silvie, P., Martin, P.: Explicit versus Tacit Knowledge in Duquenne-Guigues Basis of Implications: Preliminary Results. In: Workshop – Analyzing Real Data with Formal Concept Analysis (RealDataFCA'2021) (Jun 2021)
18. Silvie, P.J., Martin, P., Huchard, M., Keip, P., Gutierrez, A., Sarter, S.: Prototyping a knowledge-based system to identify botanical extracts for plant health in subsaharan africa. Plants **10**(5) (2021). https://doi.org/10.3390/plants10050896