An Appraisal of Automated Tools for FAIRness Evaluation

Leonardo Guerreiro Azevedo[0000−0002−2109−1285], Gabriel Banaggia[0000−0001−5555−894X], Julio Tesolin[0000−0002−0240−4506], and Renato Cerqueira[0000−0003−2829−7857]

IBM Research, Av. República do Chile, 330, 20031-170, Rio de Janeiro, RJ, Brazil
lga@br.ibm.com,{gbanaggia,julio.tesolin}@ibm.com, rcerq@br.ibm.com

Abstract. The FAIR Principles were introduced to address data challenges and improve the Findability, Accessibility, Interoperability, and Reusability of digital resources, following several Semantic Web standards. 'FAIRness' corresponds to a percentage grade indicating how close a digital object is to abiding by those principles. Several tools have been developed to assess the FAIRness of data digital objects in support of enacting the FAIR Principles. This work offers an appraisal of tools that evaluate the FAIRness of such objects, focusing on fully automated solutions. We conduct a literature review about existing tools, extract from it a set of requirements they aim to fulfill, and assess how each one fares considering this ensemble. Our results help researchers and data stewards with an overview of the tools, including an analysis of the fulfillment of the requirements and existing gaps.

Keywords: FAIR principles · FAIRness · FAIRness evaluation · Automated Tools for FAIRness Evaluation · Requirements for FAIRness Automated Tools.

1 Introduction

Research data can benefit from infrastructure improvements to support its use and the accrual of generated information and knowledge, which should be discovered and reused for investigations [43].

Enhancing knowledge discovery for human and computational agents is a challenge for data-intensive sciences, involving the access, integration, and analysis of task-appropriate data [43]. Wilkinson *et al.* [43] proposed the FAIR principles, a set of 15 recommendations for improving the **F**indability, **A**ccessibility, **I**nteroperability, and **R**eusability of digital resources [22]. The FAIR principles are supposed to be domain-independent, aim to facilitate the reuse of data by both humans and machines [39], and are related to several Semantic Web standards [2,37]. Applying the FAIR principles produces digital objects that ensure goals like transparency, reproducibility, and reusability [40].

There are several mechanisms to support data infrastructure in the task of designing FAIR data, such as guidelines, processes, questionnaires, and semi-

automated and automated tools that aim to evaluate a digital object's 'FAIRness' level. FAIRness corresponds to a percentage grade indicating how close a digital object is to abiding by the FAIR principles' characteristics considered relevant for a community or task. Manual and semi-automated mechanisms are essential to improve overall understanding and appreciation of the research life cycle [42]; nevertheless, assessing FAIRness with them is time-consuming, requires experience, carries difficulties when inspection is needed, and does not scale when considering several digital objects [18]. An automated tool is more appropriate to handle these issues. Automated tools perform the FAIRness evaluation without human intervention, resulting in a more objective evaluation and allowing the comparison of distinct digital objects.

This work analyzes tools for FAIRness assessment with a focus on automated solutions. We conducted a literature review to discover the tools and elicited the tools' requirements. None of the surveyed papers systematically present such a requirement list. They characterize and compare tools abstractly and do not use requirements. We examine the tools to verify how they fulfill the requirements and analyze their coverage and existing gaps.

Our main contributions are the aggregated requirements list and the evaluation of the tools. The requirements guide analyzing and developing tools that effectively and automatically evaluate FAIRness. The overview of how existing tools support the requirements aids one in choosing an existing tool to apply in one's scenario or engaging in new tool development.

The remainder of this work is divided as follows. Section 2 presents the literature review, the tools found, and the elicited requirements. Section 3 describes the automated tools for FAIRness evaluation and how they address the requirements. Finally, Section 4 presents our conclusions, an overall discussion about the appraisal, and proposals for future work.

2 Literature Review

Scientific research takes into account the existing literature on a topic as the best evidence for assessing the current state of any particular subject of inquiry. In computer science, evidence-based approaches were developed inspired by methods initially proposed for the social sciences and medicine research, with a strong, evidence-based empirical focus [24]. Evidence is constituted by a synthesis of the best available studies that help answer specific research questions. One of the most important ways of generating that synthesis is by doing a systematic literature review (SLR) [26]. However, SLRs are labor-intensive procedures, and for reduced teams with narrower questions, simplified literature reviews can already supply interesting results, although preliminary [8]. This work conducted this type of abbreviated literature review, which steps are detailed below.

We started with the definition of an abridged literature review protocol, which involved the following two research questions:

RQ1. *What are the existing automated tools for FAIRness evaluation?*

RQ2. *Which requirements do these tools meet?*

In order to find relevant academic texts to answer these questions, we established the chosen terms and synonyms for the five criteria according to the PICOC method [30]: (i) **P**opulation: Tool, Automated (Process/Method); (ii) **I**ntervention: Assessment, Evaluation; (iii) **C**omparison: (not included, as we are not searching against an established tool); (iv) **O**utcome: FAIRness, FAIRification; (v) **C**ontext: FAIR Principles, FAIR Data.

Subsequently, we used those terms to create the following search string: *("Tool" OR "Automated") AND ("Assessment" OR "Evaluation") AND ("FAIRness" OR "FAIRification") AND ("FAIR Principles" OR "FAIR Data").* This string was used to create queries in three different repositories (Scopus[1], IEEE Digital Library[2], and ACM Digital Library[3]), restricting the search options to titles, keywords and abstracts as much as possible to achieve a manageable and meaningful total number of results.

There were 32 unique hits initially, with the majority coming from Scopus. Inclusion and exclusion criteria were enacted so that we could privilege texts that already dealt with comparative assessment of different automated FAIRness evaluation tools, which are [25,29,36,38]. These works characterize and compare tools abstractly and do not propose or use requirements in their analysis. From these works, we found the tools listed in Table 1, where FAIR-Checker, F-UJI, and FAIR Evaluator are pointed out as automated. We confirmed that by reading the tools documentation. Looking for a reference to FAIRenough listed by [25], we found a tool with a similar name (FAIR Enough) that is automated and based on F-UJI and FAIR Evaluator. We included this tool in the analysis.

To appraise the tools, we elicited the requirements presented in Table 2 by reading the papers found in the literature review and the papers presenting the automated tools. The former papers elected different parameters and focal points for appraising the tools for FAIRness assessment, but none of them present or use a list of requirements in their analysis. We created a requirement for any mention of a desirable feature for an automated tool. The purpose of the requirements is to guide the appraisal and development of tools that effectively and automatically evaluate FAIRness. This is crucial for making objective FAIRness evaluations and improving digital objects.

[1] `https://www.scopus.com`
[2] `https://ieeexplore.ieee.org`
[3] `https://dl.acm.org/dl.cfm`
[4] FAIR Enough was not found in the surveyed papers. It was included in the evaluation due to be a free automated tool based on F-UJI and FAIR Evaluator.
[5] `http://oznome.csiro.au/5star/` (link is broken)
[6] A survey-based tool that is not accessible since 2021 [25].
[7] `https://bigdata.cgiar.ocrg/resources/gardian/` (link is broken)
[8] The FAIRness badge is a generated image that presents an overview of the results of a FAIRness evaluation, *e.g.*, it may be a pie, bar, or radar chart.

Table 1. Tools found during the literature review.

Tool name and reference for the tool	**Automated?**	**Found at**
F-UJI [13,14]	Yes	[29,38]
FAIR Evaluator [41,42]	Yes	[25,29,38]
FAIR Enough [20]	Yes	[4]
FAIR-Checker [18,34]	Yes	[38]
ARDC's FAIR Data Self Assessment Tool [5]	No	[25,36]
Checklist for Evaluation of Dataset Fitness for Use [4]	No	[29]
CSIRO's 5✻ OzNome Data tool[5]	No	[36]
DANS's SATIFYD [11]	No	[25,36]
Data Stewardship Wizard [27,28]	No	[25]
EUDAT's Checklist [23]	No	[36]
FAIRdat [1]	No	[25]
FAIRenough[6]	No	[25]
FAIRshake [9,10]	No	[25,29]
GARDIAN[7]	No	[25]
RDA's Simple Grid [12]	No	[25,36]
Semi-automated workflow for FAIR maturity indicators [6,7]	No	[25]

To facilitate the requirements presentation, we grouped them into categories using a bottom-up approach, according to their similarities. We present the categories' descriptions and why each requirement matters.

- **General**: Requirements that deal with basic needs the tools have to fulfill. *Reasons*: R1: Without supplying a fully automated solution, FAIRness evaluation does not scale. R2: Without a numeric score, it is difficult to objectively compare results of FAIRness evaluation executed in the same contexts (*e.g.*, FAIR aspects considered, and configurations of metrics and tests used in the evaluation). A maturity level is another kind of FAIRness grade, *e.g.*, the RDA FAIR Data Maturity Model [31] propose maturity levels from 0 to 5 to indicate "how a resource under evaluation performs on meeting the indicators across the FAIR areas". R3: Without being fully autonomous, evaluation is limited in scope to specific repositories and their restrictions. R4: Without referencing the FAIR principles, users cannot follow how each one was interpreted and implemented. R5: With no mention to the types of digital objects that can be evaluated, the scope of the principles is diminished.
- **Deployment**: Requirements that deal with decisions taken by developers. *Reasons*: R6 and R7: Without offering the tool both as a Web page and service, it does not meet the FAIR principles of simultaneous human and machine actionability. R12: Without a badge, the user does have the whole assessment in a visual representation. R8: Without specifying its use license, different use communities cannot adequately employ the tool. R9: Without specifying its development stage, users do not know whether results can vary after new versions of the tool come out. R10: Without the ability to customize the tool, evaluation is limited to agnostic parameters, *i.e.*, digital

Table 2. List of requirements elicited from analyzing the papers returned by the literature review, or through reading the evaluated tools' papers. The latter ones have their IDs marked with a '*'.

ID	Requirement: The tool should...	Source(s)
R1	...be fully **automated**.	[38]
R2	...give a FAIRness **score**/grade.	[2,25,29,36]
R3	...should work autonomously, *i.e.*, it should work **independently** from a specific domain, digital objects, or framework.	[36]
R4	...present **which principles** are being evaluated with each FAIRness test.	[25]
R5*	...specify the **types of digital object** it assesses (*e.g.*, data objects, data repositories, workflows, software).	[14,42]
R6	...expose their services as APIs (*e.g.*, as **RESTful services** [32]).	[38]
R7	...be available as a standalone **web application**	[29,38]
R8	...indicate its usage **license**.	[25]
R9	...state its **development stage** (*e.g.*, in development, in beta, in production).	[25]
R10	...be **customizable** according to the type of digital object and community.	[2,29]
R11	...allow the user supply **authentication** credentials so that (meta)data can be accessed in their repositories for proper evaluation.	[29]
R12*	...provide a visual representation (*e.g.*, a **badge**[8]) of the FAIR assessment results.	[14,18]
R13*	...support all phases of the **data lifecycle**.	[14,3]
R14*	...rely on **FAIR-enabling services** (*e.g.*, FAIRsharing.org, identifiers.org) to perform the assessment.	[14,42]
R15	...offer **guidance** on how it is used (*e.g.*, providing user manual, help, publications, or explanatory tips).	[25]
R16	...require **little expertise** to use.	[25]
R17	...export the result in a **machine-actionable** way.	[29,38]
R18	...disclose its **rating system** (*e.g.*, presenting evidences concerning the algorithm used and the rationale for the scale chosen).	[36]
R19	...be informative, *i.e.*, it should **teach** the user about the FAIR principles.	[36]
R20	...give **recommendations** on how to improve the FAIRness of the evaluated resource.	[25,36]
R21	...export the result fully readable in **natural language**.	[36]
R22	...store the evaluation results in a **searchable** resource.	[29]
R23*	...support **versioning** of FAIRness assessment.	[42]

objects being evaluated using the same metrics, tests, and weights independent from the community-specific characteristics. R11: Without allowing authentication, tools cannot evaluate protected data. R13: Without specifying the data lifecycle, users do not know in which data management step the tool can be used. R14: Without recourse to FAIR-enabling services, the tools would not benefit from standardized, open, and well-governed services that support long-term data assessment [14]. As an example, the FAIRsharing.org[9] is an initiative that provides such kind of services [35].

- **User experience**: Requirements that relate to how users employ the tool. *Reasons*: R15: Without receiving guidance, users are left to interpret design decisions by themselves, which could lead to a misreading of its results. R16: Without being open to users with little technical expertise, entire sections of scientific communities would be cast aside from using the tool.
- **Reporting**: Requirements about how evaluation results are reported. *Reasons*: R17: Without machine-actionable, the FAIR principles are not fully realized since one of their main goals is allowing programmatic access to scientific data. R18: Without explaining the rating system, design decisions are made opaque and the score becomes uninterpretable. R19 and R20: Without teaching users about the FAIR principles (*e.g.*, explaining the principles or referencing documents about them) or giving recommendations (*e.g.*, suggesting ways to improve the FAIRness like referencing recipes or standard schemas to be used), one misses the opportunity to increase the FAIRness of data. R21: Without reports supplied in natural language, non-technical users are left out of the end result. R22 and R23: Without being able to store and search for past results, it becomes impossible to gauge how FAIRness improved over time.

3 Appraisal of the tools

Table 3 presents the appraisal of the automated tools for FAIRness evaluation. We do not present in the table the requirements that are fulfilled by all the tools, which are R1 to R8, R16, and R18.

This section describes the tools and analyzes how they meet the requirements. The requirements were evaluated by reading the documentation of the tools (*i.e.*, web pages, GitHub pages, and papers). The score "Yes" (✓) means the requirement is totally supported. "Partially" (◗) is a requirement not entirely fulfilled. "No" (✗) means it is ignored by the tool.

3.1 F-UJI

F-UJI automatically (**R1**:✓) evaluates the FAIRness (**R2**:✓) of research data objects (**R5**:✓) according to FAIRsFAIR metrics[10] [14]. The evaluation can be applied to data digital objects belonging to any domain (**R3**:✓).

[9] `https://fairsharing.org/`

[10] The version V0.4 of the metrics are available at Zenodo [15].

Table 3. Tools' appraisal considering the elicited requirements, using symbols to represent entire (✓ = Yes), partial (◗ = Partially) or lack of (✗ = No) fulfillment.

Req.	Req. Keyword	F-UJI	FAIR Evaluator	FAIR Enough	FAIR Checker
R1	Automated	✓	✓	✓	✓
R2	Score	✓	✓	✓	✓
R3	Independent	✓	✓	✓	✓
R4	Refer principles	✓	✓	✓	✓
R5	Types of digital object	✓	✓	✓	✓
R6	RESTful services	✓	✓	✓	✓
R7	Web application	✓	✓	✓	✓
R8	License	✓	✓	✓	✓
R9	Development stage	◗	✗	✗	◗
R10	Customizable	◗	◗	◗	◗
R11	Allow authentication	✓	✗	✗	✗
R12	Badge	✓	✗	✓	✓
R13	Data lifecycle	✗	✗	✗	✗
R14	FAIR-enabling services	✓	✓	✓	◗
R15	Guidance	✓	✓	✗	✓
R16	Little expertise	✓	✓	✓	✓
R17	Machine-actionable	✓	✓	✓	✗
R18	Rating system	✓	✓	✓	✓
R19	Teach	◗	◗	◗	◗
R20	Recommendations	✗	✗	✗	✓
R21	Natural language	◗	◗	◗	✓
R22	Searchable	✗	✗	✗	✗
R23	Versioning	✗	✗	✗	✗

The tool's primary entities are Principles, Metrics, and Tests. Each FAIR principle has one or more metrics that are automatically evaluated by tests (**R4**:✓). Each test is executed using a pass-or-fail approach and returns both a score and a maturity level to represent an overview of the digital object's fitness to each FAIR principle. The score represents the FAIRness of the digital object. The maturity levels are Incomplete, Initial, Moderate, and Advanced. They are based on CMMI (Capability Maturity Model Integration) maturity levels [21].

F-UJI is available as a RESTful Web service [32] (**R6**:✓), implemented in Python using the Flask framework, and it is described using the OpenAPI[11] specification. The service accepts two inputs: the unique identifier of the data object to be evaluated and, if available, the repository that provisions the digital object's metadata. The F-UJI Web Client (**R7**:✓) renders a FAIRness badge to give a visual representation of the measurements of the evaluated object (**R12**:✓). The service[12] and the Web Client[13] source codes are available on GitHub under the MIT License (**R8**:✓). There is no information in the F-UJI documentation about in what development stage the tool is (**R9**:◗).

The tool is only customizable by implementing new tests or deactivating some of them by altering the source code, which requires software development skills. There is no configuration option for non-experts (**R10**:◗). F-UJI's documentation does not detail to which data lifecycle stage the tool applies (**R13**:✗).

F-UJI employs FAIR-enabling services to perform the evaluation (**R14**:✓). F-UJI implements content negotiation to access metadata that may be available through proprietary Web service (**R11**:✓).

The tool has plenty of documentation, including papers and GitHub pages (**R15**:✓). The Web Client is easy to use since only the digital object identifier must be passed to execute the FAIRness evaluation. However, Web service development skills are required to make use of the RESTful Web API (**R16**:✓).

The evaluation results are returned in JSON with scores, practical tests, inputs and outputs, and the evaluation context for each metric. The results can be processed by computing agents (**R17**:✓). The Web Client presents the results as an HTML page. This report presents evidence on how the digital object is evaluated (**R18**:✓) and helps the user learn about FAIR concepts. Although the latter is very useful, it requires technical skills to be fully understood (**R19**:◗). The tool does not present explicit recommendations to improve the FAIRness of digital objects (**R20**:✗). The results are not so easy to understand by non-IT people, even though they are readable in natural language (**R21**:◗). Besides, they are not stored in a searchable engine (**R22**:✗) and, consequently, F-UJI does not support storing versions of the evaluation results (**R23**:✗).

3.2 FAIR Evaluator

FAIR Evaluator is a framework to automatically (**R1**:✓) evaluate digital objects and return its FAIRness (**R2**:✓). The framework has three main components:

[11] `https://www.openapis.org/`
[12] `https://github.com/pangaea-data-publisher/fuji`
[13] `https://github.com/MaastrichtU-IDS/fairificator`

(a) Maturity Indicators (MI): community-authored specifications delimiting a specific automatically-measurable FAIR behavior; (b) Compliance Tests: services that test digital resources against MIs; (c) Evaluator: a Web application that registers, assembles, and applies community-relevant sets of Compliance Tests against a digital resource, providing a report about what a machine 'sees' when it visits that resource. The framework is not tied to a specific domain (**R3**:✓) and can be adapted to fit community needs, *i.e.*, the stakeholders may participate in the creation of community-specific MIs, develop their own compliance tests, and define which tests to use in an evaluation. The tests also refer to the principles they evaluate (**R4**:✓). It can evaluate not only data digital objects but other types as well, like software digital objects, as long as the user configures the MI Collection with the options that make sense for the evaluation (**R5**:✓).

The FAIR Evaluator is primarily designed for mechanized interaction through its RESTful Web Service API (**R6**:✓) and provides a demonstrative user interface [41] for form-based access (**R7**:✓). The client-side interface is implemented in JavaScript using the AngularJS framework. The service is a Ruby on Rails application with four primary components: (a) a registry of known MI Tests and their annotations; (b) a registry of collections of MI Tests ("Collections") and their annotations; (c) an invocation function that creates a pipeline of MI Tests and applies them to a resource; and (d) a registry of evaluation results. The codes of the framework[14] and the front-end[15] are available on GitHub under the MIT license (**R8**:✓). The framework does not present a summary of the evaluation as a visual representation, like a badge (**R12**:✗). There is no information in the documentation about the tool's development stage (**R9**:✗) nor about in which phases of the data lifecycle the framework could be used (**R13**:✗).

The FAIR Evaluator has 15 defined MIs, evaluated by 22 Compliance Tests, that the user can group in a collection according to priorities and needs. This feature allows the user to indicate what is the context of a evaluation, *i.e.*, which metrics should be considered in a evaluation. For deeper customization, the user should have software development skills, *e.g.*, to create new tests or change existing ones by refactoring the source files (**R10**:◗). We did not find information about authentication features to access restricted data (**R11**:✗).

A set of FAIR standards is used in the documentation and implementation of the framework's tools, mainly based on FAIRsharing services (**R14**:✓). The framework has plenty of documentation, including papers and GitHub pages (**R15**:✓). The front-end Web Client is easy to use since one has to fill a MIs Collection, the digital object identifier, a title for the evaluation, and the user's unique identifier (*e.g.*, ORCID) to execute the FAIRness evaluation. However, Web service development skills are required to make use of the RESTful Web API, which is expected (**R16**✓).

The results generated by the framework are machine-actionable (**R17**:✓). The user can scan the output for details (**R18**:✓) and learn about FAIR concepts, although it is technical information (**R19**:◗). They give no explicit recommenda-

[14] `https://github.com/FAIRMetrics/Metrics`
[15] `https://github.com/FAIRsharing/FAIR-Evaluator-FrontEnd`

tions to improve the FAIRness of resources (**R20**:✗). Although all this information is human-readable, it is technical and not so easily understood by non-IT people (**R21**:◗). The results are not stored in a searchable engine (**R22**:✗), and although results versioning is a crucial highlighted feature [42], we did not find implementation for that (**R23**:✗).

3.3 FAIR Enough

FAIR Enough [20] evaluates resources (**R5**:✓) automatically (**R1**:✓) and returns their FAIRness level (**R2**:✓). Its implementation is based on FAIR Evaluator and F-UJI tools. The FAIR metrics test APIs can be registered and used as in the FAIR Evaluator. To run an evaluation, the user provides a URL, but DOI and handles are also accepted for some collections of tests. The tool is not domain specific (**R3**:✓). The tests' results refer to the principles (**R4**:✓).

It is available as a RESTful Web service (**R6**:✓) implemented in Python using FastAPI[16] and provides a frontend Web interface (**R7**:✓) implemented using React[17]. The code is available in GitHub[18] under MIT license (**R8**:✓).

We did not find information about the tool's development stage (**R9**:✗), in which phases of the data lifecycle the framework could be used (**R13**:✗), and authentication features to access restricted data (**R11**:✗).

The tool and the tests are customizable but require technical skills (**R10**:◗). Tests implementation uses FAIRsharing services (**R14**:✓).

There is a few documentation available in GitHub (**R15**:✗). The Web Client is easy to use in the same way as the FAIR Evaluator. However, Web service development skills are required to make use of the RESTful Web API (**R16**✓).

Results are in JSON format when calling the RESTful API or can be downloaded when using the Web interface (**R17**:✓). They contain details about how tests are executed (**R18**:✓), which can be used to learn FAIR concepts, but it is very technical (**R19**:◗). There are no explicit recommendations for FAIRness improvements (**R20**:✗). Technical skills are required to understand the results (**R21**:◗). We did not find documentation about the storage of results in a searchable engine (**R22**:✗) and about versioning (**R23**:✗).

3.4 FAIR-Checker

FAIR-Checker automatically (**R1**:✓) evaluates the FAIRness (**R2**:✓) of digital objects focusing on domain-agnostic metadata (**R3**:✓). The tool can evaluate any digital object, as long as they are described by metadata available on a landing page (**R5**:✓). Its goal is to promote the use of embedded metadata in Web pages to facilitate the findability and reuse of digital scientific resources. The tool helps data producers and curators identify improvement areas and take steps toward enhancing the FAIRness of their resources [18].

[16] `https://fastapi.tiangolo.com/`
[17] `https://react.dev/`
[18] `https://github.com/MaastrichtU-IDS/fair-enough`

The tool execution process starts with the user supplying a URL as input. Then, it extracts semantic annotations from the Web page to create the first version of a Knowledge Graph (KG)[19]. Public KGs are queried using SPARQL during the FAIRness evaluation against metrics tailored to the evaluation of computation ontologies, according to the FAIR principles (**R4**:✓).

FAIR-Checker is a Web application (**R7**:✓) developed in Python based on the Flask Web Framework. Recently, a RESTful API (**R6**:✓) has been made available to provide FAIRness evaluation to online client tools. The tool is available on GitHub[20] under the MIT license (**R8**:✓) and can be used on the Web [34]. We did not find explicit information about the tool's development stage (**R9**:◗). In addition to that, there is no information about in which phases of the data lifecycle the tool can be used (**R13**:✗).

The tool customization requires software development skills (**R10**:◗). Currently, the tool does not handle content negotiation (**R11**:✗). The tool uses state-of-the-art semantic Web technologies and standards to perform the evaluation but does not employ explicit FAIR-enabling services (**R14**:◗).

The tool has plenty of documentation, including papers and GitHub pages (**R15**:✓). To use the tool, the user has to submit selected dataset identifiers to FAIR-Checker and compare the returning reports. Regular usage does not require software development skills (**R16**:✓).

The execution log can be downloaded as a CSV file, but it is not machine-readable (**R17**:✗). The results include: (i) descriptions about the running tests and comments (**R18**:✓); (ii) FAIR learning resources; however, they are tied to the implementation and not to the principle's definitions (**R19**:◗); (iii) a set of recommendations for FAIRness improvements with links to training resources, such as FAIR-CookBook [33] (**R20**:✓); (iv) a visual representation summarizing the tests (**R12**:✓). Results are intended to be understandable by non-IT experts (**R21**:✓). They are not stored in a searchable engine (**R22**:✗), and there is no support for storing versions of them either (**R23**:✗).

4 Conclusion

This work analyzed automated tools found in a literature review on FAIRness assessment. We also compiled a list of 23 desirable requirements that tools for automated FAIRness evaluation could fulfill, grouping them under four categories. We offered an analysis of if and how each of the tools under appraisal meets each of the elicited requirements.

The tools are very similar regarding overall requirement fulfillment (R1 to R8, R13, R16, R18, R19, R22, and R23) but diverge on supporting the other requirements. Considering 1 and 0.5 for complete and partial fulfillment, their requirements coverage would be F-UJI (74%), FAIR-Checker (70%), FAIR-Evaluator (63%), and FAIR Enough (63%). They all follow good software development

[19] "A knowledge graph acquires and integrates information into an ontology and applies a reasoner to derive new knowledge" [16].

[20] `https://github.com/IFB-ElixirFr/FAIR-checker`

practices and use state-of-the-art technologies in software engineering and the Semantic Web. The reporting features should be improved, considering user-experience techniques and user evaluations. The storage of results and versioning are complex features not implemented by the appraised tools.

The automated tools vary in how they support the main requirements found in the literature. No tool meets all requirements and stands out as state-of-the-art. Choosing the best tool is challenging, and there is room for evolving existing tools or developing a new one to overcome the existing gaps. To choose a tool, one should consider not only the percentage of fulfillment but also: (i) the more critical requirements for their specific scenario; (ii) the details of the appraisal presented in this work; (iii) the difficulties to customize each tool's implementations in case it becomes vital to support requirements not adequately addressed.

The results of our work can aid other researchers and developers by providing insights into the available automated tools for FAIRness assessment. The consolidated requirements can form the basis for creating a benchmark for the appraisal of FAIRness assessment tools, including enhancing existing options or even designing new approaches.

One limitation of our work is the search scope. We considered tools cited by academic works of Scopus, IEEE, and ACM. These digital libraries include papers from the most relevant journals and conference in Computer Science.

As a future work, we propose extending the search scope to other digital libraries and to the gray literature (which concerns, *e.g.*, blog posts, videos, and white papers), *i.e.*, works outside of the published (formal) literature (*e.g.*, journal and conference papers) [19]. One example is the FAIRassist.org Web page[17]. It collects and describes resources aimed at helping developers and stakeholders to make their digital objects FAIR. It lists manual questionnaires, checklists, and semi-automated and automated tools, which include the ones we evaluated. FAIRassist.org is provided by FAIRsharing.org.

We also plan to engage with users who intend to use automated tools to verify our proposal of requirements. Users could also help us by indicating weights for the requirements and organizing them as mandatory, recommended and optional since the requirements may have distinct importance for different communities.

Another future work is the development of benchmarks, considering our proposal of requirements. We also intend to appraise the automated tools in practice against those benchmarks and existing ones like the "Apples-to-Apples" (A2A) [44] benchmarks, which were developed to evaluate, in a standard way, the tools compliance with (meta)data publishing based on well-established Web standards in use by communities. They were developed because each tool usually produces different results due to distinct metrics, the use of metadata publishing paradigms, and the context considered of the FAIRness evaluation.

References

1. FAIR data assessment tool. `https://www.surveymonkey.com/r/%20fairdat`
2. Amdouni, E., Jonquet, C.: FAIR or FAIRer? an integrated quantitative fairness assessment grid for semantic resources and ontologies. In: Communications

in Computer and Information Science. vol. 1537 CCIS, pp. 67 – 80 (2022). https://doi.org/10.1007/978-3-030-98876-0_6
3. Aronsen, J.M., Beyan, O., Harrower, N., Holl, A., et al.: Recommendations on FAIR metrics for EOSC. Publications Office of the European Union, LU (2021). https://doi.org/10.2777/70791
4. Austin, C., Cousjin, H., Diepenbroek, M., Petters, J., Soares E Silva, M., Alliance, W.D.S.R.D.: WDS/RDA Assessment of Data Fitness for Use WG Outputs and Recommendations (Apr 2019). https://doi.org/10.15497/rda00034
5. Australian Research Data Commons: FAIR Data Self Assessment Tool. https://ardc.edu.au/resource/fair-data-self-assessment-tool/
6. Bonaretti, S., Willighagen, E.: Two real use cases of FAIR maturity indicators in the life sciences (Aug 2019). https://doi.org/10.1101/739334
7. Bonaretti, S., Willighagen, E.: BiGCAT-UM/FAIR_metrics. Department of Bioinformatics - BiGCaT (Jul 2021)
8. Carrera-Rivera, A., Ochoa, W., Larrinaga, F., Lasa, G.: How-to conduct a systematic literature review: A quick guide for computer science research. MethodsX **9**, 101895 (Nov 2022). https://doi.org/10.1016/j.mex.2022.101895
9. CFDE-CC: FAIRshake: A System to Evaluate the FAIRness of Digital Objects. https://fairshake.cloud/
10. Clarke, D.J., Wang, L., Jones, A., Wojciechowicz, M.L., et al.: FAIRshake: Toolkit to Evaluate the FAIRness of Research Digital Resources. Cell Systems **9**(5), 417–421 (Nov 2019). https://doi.org/10.1016/j.cels.2019.09.011
11. Data Archiving and Networked Service (DANS): FAIR self-assessment tool. https://satifyd.dans.knaw.nl/
12. David, R., Mabile, L., Yahia, M., Cambon-Thomsen, A., et al.: Comment opérationnaliser et évaluer la prise en compte du concept 'FAIR' dans le partage des données: Vers une grille simplifiée d'évaluation du respect des critères FAIR. (Dec 2018). https://doi.org/10.5281/zenodo.2551500
13. Devaraju, A., Huber, R.: F-UJI - An Automated FAIR Data Assessment Tool. Zenodo (Oct 2020). https://doi.org/10.5281/zenodo.4063720
14. Devaraju, A., Huber, R.: An automated solution for measuring the progress toward FAIR research data. Patterns **2**(11), 100370 (2021). https://doi.org/10.1016/j.patter.2021.100370
15. Devaraju, A., Huber, R., Mokrane, M., Cepinskas, L., Davidson, J., Herterich, P., L'Hours, H., de Vries, J., White, A.: FAIRsFAIR data object assessment metrics. Tech. rep., Zenodo (2020). https://doi.org/10.5281/zenodo.3934401
16. Ehrlinger, L., Wöß, W.: Towards a definition of knowledge graph. SEMANTICS 2016: Posters and Demos Track **48**(1-4), 2 (2016)
17. fairsharing: Fairassist.org. https://fairassist.org/, accessed in 2024–04-12
18. Gaignard, A., Rosnet, T., De Lamotte, F., Lefort, V., Devignes, M.D.: FAIR-Checker: supporting digital resource findability and reuse with Knowledge Graphs and Semantic Web standards. Journal of Biomedical Semantics **14**(1), 7 (Jul 2023). https://doi.org/10.1186/s13326-023-00289-5
19. Garousi, V., Felderer, M., Mäntylä, M.V.: Guidelines for including grey literature and conducting multivocal literature reviews in software engineering. Information and Software Technology **106**, 101–121 (Feb 2019). https://doi.org/10.1016/j.infsof.2018.09.006
20. Institute of Data Science: FAIR Enough. https://fair-enough.semanticscience.org/
21. ISACA: CMMI Institute - CMMI. https://cmmiinstitute.com/cmmi

22. Jacobsen, A., de Miranda Azevedo, R., Juty, N., *et al.*, B.: FAIR Principles: Interpretations and Implementation Considerations. Data Intelligence **2**(1-2), 10–29 (2020). https://doi.org/10.1162/dint_r_00024
23. Jones, S., Grootveld, M.: How FAIR are your data? (Nov 2017). https://doi.org/10.5281/zenodo.3405141
24. Kitchenham, B., Charters, S.: Guidelines for performing Systematic Literature Reviews in Software Engineering. Tech. rep., Durham University Evidence-Based Software Engineering (2007)
25. Krans, N., Ammar, A., Nymark, P., Willighagen, E., Bakker, M., Quik, J.: FAIR assessment tools: evaluating use and performance. NanoImpact **27** (2022). https://doi.org/10.1016/j.impact.2022.100402
26. Machado, M.d.O.C., Bravo, N.F.S., Martins, A.F., Bernardino, H.S., Barrere, E., Souza, J.F.d.: Metaheuristic-based adaptive curriculum sequencing approaches: a systematic review and mapping of the literature. Artificial Intelligence Review **54**(1), 711–754 (Jan 2020). https://doi.org/10.1007/s10462-020-09864-z, https://doi.org/10.1007/s10462-020-09864-z
27. Pergl, R., Hooft, R., Suchánek, M., Knaisl, V., Slifka, J.: "Data Stewardship Wizard": A Tool Bringing Together Researchers, Data Stewards, and Data Experts around Data Management Planning. Data Science Journal **18**(1), 59 (Dec 2019). https://doi.org/10.5334/dsj-2019-059
28. Pergl, R., Hooft, R., Vondrášek, J., et al.: Data Stewardship Wizard. https://ds-wizard.org/
29. Peters-Von Gehlen, K., Höck, H., Fast, A., Heydebreck, D., Lammert, A., Thiemann, H.: Recommendations for Discipline-Specific FAIRness Evaluation Derived from Applying an Ensemble of Evaluation Tools. Data Science Journal **21**(1) (2022). https://doi.org/10.5334/dsj-2022-007
30. Petticrew, M., Roberts, H.: Systematic reviews in the social sciences: A practical guide. Blackwell Publishing, Malden (2006). https://doi.org/10.1002/9780470754887
31. RDA: FAIR Data Maturity Model: specification and guidelines (Jun 2020). https://doi.org/10.15497/rda00050
32. Richardson, L., Amundsen, M., Ruby, S.: RESTful web APIs: services for a changing world. O'Reilly Media Inc., (2013)
33. Rocca-Serra, P., Gu, W., Ioannidis, V., Abbassi-Daloii, T., et al.: The FAIR Cookbook - the essential resource for and by FAIR doers. Scientific Data **10**(1), 292 (May 2023). https://doi.org/10.1038/s41597-023-02166-3
34. Rosnet, T., Gaignard, A., Devignes, M.D.: FAIR-checker. https://fair-checker.france-bioinformatique.fr/
35. Sansone, S.A., McQuilton, P., Rocca-Serra, P., Gonzalez-Beltran, A., Izzo, M., Lister, A.L., Thurston, M.: FAIRsharing as a community approach to standards, repositories and policies. Nature Biotechnology **37**(4), 358–367 (Apr 2019). https://doi.org/10.1038/s41587-019-0080-8
36. Slamkov, D., Stojanov, V., Koteska, B., Mishev, A.: A comparison of data fairness evaluation tools. In: CEUR Workshop Proceedings. vol. 3237. CEUR-WS (2022)
37. van Soest, J., Choudhury, A., Gaikwad, N., Sloep, M., Dekker, A.: Annotation of existing databases using Semantic Web technologies: making data more FAIR. In: 12th International Conference on Semantic Web Applications and Tools for Health Care and Life Sciences (2019)
38. Sun, C., Emonet, V., Dumontier, M.: A comprehensive comparison of automated FAIRness evaluation tools. In: CEUR Workshop Proceedings. vol. 3127 (2022)

39. Trojahn, C., Kamel, M., Annane, A., et al.: A FAIR core semantic metadata model for FAIR multidimensional tabular datasets. In: Knowledge Engineering and Knowledge Management. pp. 174–181. Springer International Publishing (2022). https://doi.org/10.1007/978-3-031-17105-5_13
40. Wilkinson, M., Dumontier, M., Durbin, P.: DataFairPort: The Perl libraries version 0.231 (2015). https://doi.org/10.5281/zenodo.33584
41. Wilkinson, M., et al.: The FAIR Maturity Evaluation Service. https://fairsharing.github.io/FAIR-Evaluator-FrontEnd
42. Wilkinson, M.D., Dumontier, M., Sansone, S.A., Bonino da Silva Santos, L.O., et al.: Evaluating fair maturity through a scalable, automated, community-governed framework. Scientific data **6**(1), 1–12 (2019)
43. Wilkinson, M.D., Dumontier, M., et al.: The fair guiding principles for scientific data management and stewardship. Scientific data **3**(1), 1–9 (2016)
44. Wilkinson, M.D., Sansone, S.A., Marjan, G., Nordling, J., Dennis, R., Hecker, D.: FAIR Assessment Tools: Towards an "Apples to Apples" comparisons (Dec 2022). https://doi.org/10.5281/zenodo.7463421