

A peer-reviewed version of this preprint was published in PeerJ on 24 April 2017.

[View the peer-reviewed version](https://peerj.com/articles/cs-110) (peerj.com/articles/cs-110), which is the preferred citable publication unless you specifically need to cite this preprint.

Wilkinson MD, Verborgh R, Bonino da Silva Santos LO, Clark T, Swertz MA, Kelpin FDL, Gray AJG, Schultes EA, van Mulligen EM, Ciccarese P, Kuzniar A, Gavai A, Thompson M, Kaliyaperumal R, Bolleman JT, Dumontier M. 2017. Interoperability and FAIRness through a novel combination of Web technologies. PeerJ Computer Science 3:e110 <https://doi.org/10.7717/peerj-cs.110>

1

2

Interoperability and FAIRness through a novel combination of Web technologies

3

4

Authors:

5

6 **Mark D. Wilkinson** - Center for Plant Biotechnology and Genomics, UPM-INIA, Madrid, Spain

7 **Ruben Verborgh** – Ghent University – IMEC, Ghent, Belgium

8 **Luiz Olavo Bonino da Silva Santos** - Dutch Techcentre for Life Sciences, Utrecht, The Netherlands - Vrije Universiteit Amsterdam, Amsterdam, The Netherlands

9 **Tim Clark** - Department of Neurology, Massachusetts General Hospital Boston MA and Harvard Medical School, Boston, MA, USA

10 **Morris A. Swertz** - Genomics Coordination Center and Department of Genetics, University Medical Center Groningen, Groningen, The Netherlands

11 **Fleur D.L. Kelpin** - Genomics Coordination Center and Department of Genetics, University Medical Center Groningen, Groningen, The Netherlands

12 **Alasdair J. G. Gray** - Department of Computer Science, School of Mathematical and Computer Sciences, Heriot-Watt University, Edinburgh, UK

13 **Erik A. Schultes** - Department of Human Genetics, Leiden University Medical Center, Leiden, The Netherlands

14 **Erik M. van Mulligen** - Department of Medical Informatics, Erasmus University Medical Center Rotterdam, The Netherlands

15 **Paolo Ciccarese** - Perkin Elmer Innovation Lab, Cambridge MA and Harvard Medical School, Boston MA, USA

16 **Arnold Kuzniar**, Netherlands eScience Center, Amsterdam, The Netherlands

17 **Anand Gavai**, Netherlands eScience Center, Amsterdam, The Netherlands

18 **Mark Thompson** - Leiden University Medical Center, Leiden, The Netherlands

19 **Rajaram Kaliyaperumal** - Leiden University Medical Center, Leiden, The Netherlands

20 **Jerven T. Bolleman** - Swiss-Prot group, SIB Swiss Institute of Bioinformatics, Centre Medical Universitaire, Geneva, Switzerland

21 **Michel Dumontier** - Stanford Center for Biomedical Informatics Research, Stanford University, Stanford, California

22

Corresponding Author:

23

24 Mark D. Wilkinson

25 *markw@illuminae.com*, +34 622 784 026

26

40

41

42 Abstract

43 Data in the life sciences are extremely diverse and are stored in a broad spectrum of
44 repositories ranging from those designed for particular data types (such as KEGG for
45 pathway data or UniProt for protein data) to those that are general-purpose (such as
46 FigShare, Zenodo, Dataverse or EUDAT). These data have widely different levels of
47 sensitivity and security considerations. For example, clinical observations about genetic
48 mutations in patients are highly sensitive, while observations of species diversity are
49 generally not. The lack of uniformity in data models from one repository to another, and in
50 the richness and availability of metadata descriptions, makes integration and analysis of
51 these data a manual, time-consuming task with no scalability. Here we explore a set of
52 resource-oriented Web design patterns for data discovery, accessibility, transformation, and
53 integration that can be implemented by any general- or special-purpose repository as a
54 means to assist users in finding and reusing their data holdings. We show that by using off-
55 the-shelf technologies, interoperability can be achieved at the level of an individual
56 spreadsheet cell. We note that the behaviours of this architecture compare favourably to the
57 desiderata defined by the FAIR Data Principles, and can therefore represent an exemplar
58 implementation of those principles. The proposed interoperability design patterns may be
59 used to improve discovery and integration of both new and legacy data, maximizing the
60 utility of all scholarly outputs.

61

62

63

64 Introduction

65 Carefully-generated data are the foundation for scientific conclusions, new hypotheses,
66 discourse, disagreement and resolution of these disagreements, all of which drive scientific
67 discovery. Data must therefore be considered, and treated, as first-order scientific output,
68 upon which there may be many downstream derivative works, among these, the familiar
69 research article (Starr et al., 2015). But as the volume and complexity of data continue to
70 grow, a data publication and distribution infrastructure is beginning to emerge that is not *ad*
71 *hoc*, but rather explicitly designed to support discovery, accessibility, (re)coding to
72 standards, integration, machine-guided interpretation, and re-use.

73

74 In this text, we use the word “data” to mean all digital research artefacts, whether they be
75 data (in the traditional sense), research-oriented digital objects such as workflows, or
76 combinations/packages of these (i.e. the concept of a “research object”, (Bechhofer et al.,
77 2013)). Effectively, all digital entities in the research data ecosystem will be considered data
78 by this manuscript. Further, we intend “data” to include both data and metadata, and
79 recognize that the distinction between the two is often user-dependent. Data, of all types,
80 are often published online, where the practice of open data publication is being encouraged
81 by the scholarly community, and increasingly adopted as a requirement of funding agencies
82 (Stein et al., 2015). Such publications utilize either a special-purpose repository (e.g. model-
83 organism or molecular data repositories) or increasingly commonly will utilize general-
84 purpose repositories such as FigShare, Zenodo, Dataverse, EUDAT or even institutional
85 repositories. Special-purpose repositories generally receive dedicated funding to curate and
86 organize data, and have specific query interfaces and APIs to enable exploration of their
87 content. General-purpose repositories, on the other hand, allow publication of data in
88 arbitrary formats, with little or no curation and often very little structured metadata. Both of
89 these scenarios pose a problem with respect to interoperability. While APIs allow
90 mechanized access to the data holdings of a special-purpose repository, each repository has
91 its own API, thus requiring specialized software to be created for each cross-repository
92 query. Moreover, the ontological basis of the curated annotations are not always
93 transparent (neither to humans nor machines), which hampers automated integration.
94 General purpose repositories are less likely to have rich APIs, thus often requiring manual
95 discovery and download; however, more importantly, the frequent lack of harmonization of
96 the file types/formats and coding systems in the repository, and lack of curation, results in
97 much of their content being unusable (Roche et al., 2015).

98

99 Previous projects, specifically in the bio/medical domain, that have attempted to achieve
100 deep interoperability include caBIO (Covitz et al., 2003) and TAPIR (De Giovanni et al.,
101 2010). The former created a rich SOAP-based API, enforcing a common interface over all
102 repositories. The latter implemented a domain-specific query language that all participating
103 repositories should respond to. These initiatives successfully enabled powerful cross-
104 resource data exploration and integration; however, this was done at the expense of broad-
105 scale uptake, partly due to the complexity of implementation, and/or required the

106 unavoidable participation of individual data providers, who are generally resource-strained.
107 Moreover, in both cases, the interoperability was aimed at a specific field of study (cancer,
108 and biodiversity respectively), rather than a more generalized interoperability goal spanning
109 all domains.

110
111 With respect to more general-purpose approaches, and where 'lightweight' interoperability
112 was considered acceptable, myGrid (Stevens et al., 2003) facilitated discovery and
113 interoperability between Web Services through rich ontologically-based annotations of the
114 service interfaces, and BioMoby (Wilkinson et al., 2008) built on these myGrid annotations
115 by further defining a novel ontology-based service request/response structure to guarantee
116 data-level compatibility and thereby assist in workflow construction (Withers et al., 2010).
117 SADI (Wilkinson et al., 2011), and SSWAP (Gessler et al., 2009) used the emergent
118 Semantic Web technologies of RDF and OWL to enrich the machine-readability of Web
119 Service interface definitions and the data being passed - SADI through defining service
120 inputs and outputs as instances of OWL Classes, and SSWAP through passing data
121 embedded in OWL 'graphs' to assist both client and server in interpreting the meaning of the
122 messages. In addition, two Web Service interoperability initiatives emerged from the World
123 Wide Web Consortium - OWL-S (Martin et al., 2005) and SAWSDL (Martin et al., 2007), both
124 of which used semantic annotations to enhance the ability of machines to understand Web
125 Service interface definitions and operations. All of these Service-oriented projects enjoyed
126 success within the community that adopted their approach; however, the size of these
127 adopting communities have, to date, been quite limited and are in some cases highly
128 domain-specific. Moreover, each of these solutions is focused on Web Service functionality,
129 which represents only a small portion of the global data archive, where most data is
130 published as static records. Service-oriented approaches additionally require data
131 publishers to have considerable coding expertise and access to a server in order to utilize
132 the standard, which further limits their utility with respect to the 'lay' data publishers that
133 make-up the majority of the scholarly community. As such, these and numerous other
134 interoperability initiatives, spanning multiple decades, have yet to convincingly achieve a
135 lightweight, broadly domain-applicable solution that works over a wide variety of static and
136 dynamic source data resources, and can be implemented with minimal technical expertise.

137
138 There are many stakeholders who would benefit from progress in this endeavour. Scientists
139 themselves, acting as both producers and consumers of these public and private data; public
140 and private research-oriented agencies; journals and professional data publishers both
141 "general purpose" and "special purpose"; research funders who have paid for the underlying
142 research to be conducted; data centres (e.g. the EBI (Cook et al., 2016), and the SIB (SIB
143 Swiss Institute of Bioinformatics Members, 2016)) who curate and host these data on behalf
144 of the research community; research infrastructures such as BBMRI-ERIC (van Ommen et
145 al., 2015) and ELIXIR (Crosswell & Thornton, 2012), and diverse others. All of these
146 stakeholders have distinct needs with respect to the behaviours of the scholarly data
147 infrastructure. Scientists, for example, need to access research datasets in order to initiate
148 integrative analyses, while funding agencies and review panels may be more interested in
149 the metadata associated with a data deposition - for example, the number of views or

150 downloads, and the selected license. Due to the diversity of stakeholders; the size,
151 nature/format, and distribution of data assets; the need to support freedom-of-choice of all
152 stakeholders; respect for privacy; acknowledgment of data ownership; and recognition of the
153 limited resources available to both data producers and data hosts, we see this endeavour as
154 one of the *Grand Challenges of eScience*.

155

156 In January 2014, representatives of a range of stakeholders came together at the request of
157 the Netherlands eScience Centre and the Dutch Techcentre for Life Sciences (DTL) at the
158 Lorentz Centre in Leiden, the Netherlands, to brainstorm and debate about how to further
159 enhance infrastructures to support a data ecosystem for eScience. From these discussions
160 emerged the notion that the definition and widespread support of a minimal set of
161 community-agreed guiding principles and practices could enable data providers and
162 consumers - machines and humans alike - to more easily find, access, interoperate, and
163 sensibly re-use the vast quantities of information being generated by contemporary data-
164 intensive science. These principles and practices should enable a broad range of integrative
165 and exploratory behaviours, and support a wide range of technology choices and
166 implementations, just as the Internet Protocol (IP) provides a minimal layer that enables the
167 creation of a vast array of data provision, consumption, and visualisation tools on the
168 Internet. The main outcome of the workshop was the definition of the so-called FAIR guiding
169 principles aimed at publishing data in a format that is **Findable, Accessible, Interoperable**
170 and **Reusable** by both machines and human users. The FAIR Principles underwent a period
171 of public discussion and elaboration, and were recently published (Wilkinson et al., 2016).
172 Briefly, the principles state:

173

174

175 **Findable** - data should be identified using globally unique, resolvable, and persistent
176 identifiers, and should include machine-actionable contextual information that can be
177 indexed to support human and machine discovery of that data.

178

179 **Accessible** - identified data should be accessible, optimally by both humans and
180 machines, using a clearly-defined protocol and, if necessary, with clearly-defined
181 rules for authorization/authentication.

182

183 **Interoperable** - data becomes interoperable when it is machine-actionable, using
184 shared vocabularies and/or ontologies, inside of a syntactically and semantically
185 machine-accessible format.

186

187 **Reusable** - Reusable data will first be compliant with the F, A, and I principles, but
188 further, will be sufficiently well-described with, for example, contextual information, so
189 it can be accurately linked or integrated, like-with-like, with other data sources.
190 Moreover, there should be sufficiently rich provenance information so reused data
191 can be properly cited.

192

193 While the principles describe the desired features that data publications should exhibit to
194 encourage maximal, automated discovery and reuse, they provide little guidance regarding
195 how to achieve these goals. This poses a problem when key organizations are already
196 endorsing, or even requiring adherence to the FAIR principles. For example, a biological
197 research group has conducted an experiment to examine polyadenylation site usage in the
198 pathogenic fungus *Magnaporthe oryzae*, recording, by high-throughput 3'-end sequencing,
199 the preference of alternative polyadenylation site selection under a variety of growth
200 conditions, and during infection of the host plant. The resulting data take the form of study-
201 specific Excel spreadsheets, BED alignment graphs, and pie charts of protein functional
202 annotations. Unlike genome or protein sequences and microarray outputs, there is no public
203 curated repository for these types of data, yet the data are useful to other researchers, and
204 should be (at a minimum) easily discovered and interpreted by reviewers or third-party
205 research groups attempting to replicate their results. Moreover, their funding agency, and
206 their preferred scientific journal, both require that they publish their source data in an open
207 public archive according to the FAIR principles. At this time, the commonly used general-
208 purpose data archival resources in this domain do not explicitly provide support for FAIR, nor
209 do they provide tooling or even guidance for how to use their archival facilities in a FAIR-
210 compliant manner. As such, the biological research team, with little or no experience in
211 formal data publishing, must nevertheless self-direct their data archival in a FAIR manner.
212 We believe that this scenario will be extremely common throughout all domains of research,
213 and thus this use-case was the initial focus for this interoperability infrastructure and FAIR
214 data publication prototype.

215
216 Here we describe a novel interoperability architecture that combines three pre-existing Web
217 technologies to enhance the discovery, integration, and reuse of data in repositories that
218 lack or have incompatible APIs; data in formats that normally would not be considered
219 interoperable such as Excel spreadsheets and flat-files; or even data that would normally be
220 considered interoperable, but do not use the desired vocabulary standards. We examine the
221 extent to which the features of this architecture comply with the FAIR Principles, and suggest
222 that this might be considered a “reference implementation” for the FAIR Principles, in
223 particular as applied to non-interoperable data in any general- or special-purpose repository.
224 We provide two exemplars of usage. The first is focused on a use-case similar to that
225 presented above, where we use our proposed infrastructure to create a FAIR, self-archived
226 scholarly deposit of biological data to the general-purpose Zenodo repository. The second,
227 more complex example has two objectives - first to use the infrastructure to improve
228 transparency and FAIRness of metadata describing the inclusion criterion for a dataset,
229 representing a subset of a special-purpose, curated resource (UniProt); and second, to show
230 how even the already FAIR data within UniProt may be transformed to increase its FAIRness
231 even more by making it interoperable with alternative ontologies and vocabularies, and more
232 explicitly connecting it to citation information. Finally, we place this work in the context of
233 other initiatives and demonstrate that it is complementary to, rather than in competition with,
234 other initiatives.

235 Methods

236 Implementation

237 Overview of technical decisions and their justification

238

239 The World Wide Web Consortium's (W3C) Resource Description Framework (RDF) offers
240 the ability to describe entities, their attributes, and their relationships with explicit semantics
241 in a standardized manner compatible with widely used Web application formats such as
242 JSON and XML. The Linked Data Principles (Berners-Lee, 2006) mandate that data items
243 and schema elements are identified by HTTP-resolvable URIs, so the HTTP protocol can be
244 used to obtain the data. Within an RDF description, using shared public ontology terms for
245 metadata annotations supports search and large scale integration. Given all of these
246 features, we opted to use RDF as the basis of this interoperability infrastructure, as it was
247 designed to share data on the Web.

248

249 Beyond this, there was a general feeling that any implementation that required a novel data
250 discovery/sharing "Platform", "Bus", or API, was beyond the minimal design that we had
251 committed to; it would require the invention of a technology that all participants in the data
252 ecosystem would then be required to implement, and this was considered a non-starter.
253 However, there needed to be some form of coalescence around the mechanism for finding
254 and retrieving data. Our initial target-community - that is, the biomedical sciences - have
255 embraced lightweight HTTP interfaces. We propose to continue this direction with an
256 implementation based on REST (Fielding & Taylor, 2002), as several of the FAIR principles
257 map convincingly onto the objectives of the REST architectural style for distributed
258 hypermedia systems, such as having resolvable identifiers for all entities, and a common
259 machine-accessible approach to discovering and retrieving different representations of those
260 entities. The implementation we describe here is largely based on the HTTP GET method,
261 and utilizes rich metadata and hypermedia controls. We use widely-accepted vocabularies
262 not only to describe the data in an interoperable way, but also to describe its nature (e.g. the
263 context of the experiment and how the data was processed) and how to access it. These
264 choices help maximize uptake by our initial target-community, maximize interoperability
265 between resources, and simplify construction of the wide (not pre-defined) range of client
266 behaviours we intend to support.

267

268 Confidential and privacy-sensitive data was also an important consideration, and it was
269 recognized early on that it must be possible, within our implementation, to identify and richly
270 describe data and/or datasets without necessarily allowing direct access to them, or by
271 allowing access through existing regulatory frameworks or security infrastructures. For
272 example, many resources within the International Rare Disease Research Consortium
273 participate in the RD Connect platform (Thompson et al., 2014) which has defined the
274 "disease card" - a metadata object that gives overall information about the individual disease
275 registries, which is then incorporated into a "disease matrix". The disease matrix provides
276 aggregate data about what disease variants are in the registry, how many individuals

277 represent each disease, and other high-level descriptive data that allows, for example,
278 researchers to determine if they should approach the registry to request full data access.

279

280 Finally, it was important that the data host/provider is not *necessarily* a participant in making
281 their data interoperable - rather, the interoperability solution should be capable of adapting
282 existing data with or without the source provider's participation. This ensures that the
283 interoperability objectives can be pursued for projects with limited resourcing, that
284 'abandoned' datasets may still participate in the interoperability framework, but most
285 importantly, that those with the needs and the resources should adopt the responsibility for
286 making their data-of-interest interoperable, even if it is not owned by them. This distributes
287 the problem of migrating data to interoperable formats over the maximum number of
288 stakeholders, and ensures that the most crucial resources - those with the most demand for
289 interoperability - become the earliest targets for migration.

290

291 With these considerations in mind, we were inspired by three existing technologies whose
292 features were used in a novel combination to create an interoperability infrastructure for both
293 data and metadata, that is intended to also addresses the full range of FAIR requirements.

294 Briefly, the selected technologies are:

295

- 296 1) The W3C's Linked Data Platform (Speicher, Arwe & Malhotra, 2015). We generated
297 a model for hierarchical dataset containers that is inspired by the concept of a Linked
298 Data Platform (LDP) Container, and the LDP's use of the Data Catalogue Vocabulary
299 (DCAT, (Maali, Erickson & Archer, 2014)) for describing datasets, data elements, and
300 distributions of those data elements. We also adopt the DCAT's use of Simple
301 Knowledge Organization System (SKOS, (Miles & Bechhofer, 18 August, 2009))
302 Concept Schemes as a way to ontologically describe the content of a dataset or data
303 record.
- 304 2) The RDF MappingLanguage (RML, (Dimou et al., 2014). RML allows us to describe
305 one or more possible RDF representations for any given dataset, and do so in a
306 manner that is, itself, FAIR: every sub-component of an RML model is Findable,
307 Accessible, Interoperable, and Reusable. Moreover, for many common semi-
308 structured data, there are generic tools that utilize RML models to dynamically drive
309 the transformation of data from these opaque representations into interoperable
310 representations (<https://github.com/RMLio/RML-Mapper>).
- 311 3) Triple Pattern Fragments (TPF - (Verborgh et al., 2016)). A TPF interface is a REST
312 Web API to retrieve RDF data from data sources in any native format. A TPF server
313 accepts URLs that represent triple patterns [Subject, Predicate, Object], where any of
314 these three elements may be constant or variable, and returns RDF triples from its
315 data source that match those patterns. Such patterns can be used to obtain entire
316 datasets, slices through datasets, or individual data points even down to a single
317 triple (essentially a single cell in a spreadsheet table). Instead of relying on a
318 standardized contract between servers and clients, a TPF interface is self-describing
319 such that automated clients can discover the interface and its data.

320

321 We will now describe in detail how we have applied key features of these technologies, in
322 combination, to provide a novel data discoverability architecture. We will later demonstrate
323 that this combination of technologies also enables both metadata and data-level
324 interoperability even between opaque objects such as flat-files, allowing the data within
325 these objects to be queried in parallel with other data on the Semantic Web.
326

327 Metadata Interoperability - The “FAIR Accessor” and the Linked Data Platform

328

329 The Linked Data Platform “*defines a set of rules for HTTP operations on Web resources... to*
330 *provide an architecture for read-write Linked Data on the Web*” (<https://www.w3.org/TR/ldp/>).
331 All entities and concepts are identified by URLs, with machine-readable metadata describing
332 the function or purpose of each URL and the nature of the resource that will be returned
333 when that URL is resolved.
334

335 Within the LDP specification is the concept of an LDP Container. A basic implementation of
336 LDP containers involves two “kinds” of resources, as diagrammed in Figure 1. The first type
337 of resource represents the container - a metadata document that describes the shared
338 features of a collection of resources, and (optionally) the membership of that collection. This
339 is analogous to, for example, a metadata document describing a data repository, where the
340 repository itself has features (ownership, curation policy, etc.) that are independent from the
341 individual data records within that repository (i.e. the members of the collection). The second
342 type of resource describes a member of the contained collection and (optionally) provides
343 ways to access the record itself.
344

345 Our implementation, which we refer to as the “FAIR Accessor”, utilizes the container concept
346 described by the LDP, however, it does not require a full implementation of LDP, as we only
347 require read functionality. In addition, other requirements of LDP would have added
348 complexity without notable benefit. Our implementation, therefore, has two resource types
349 based on the LDP Container described above, with the following specific features:
350

351 **Container resource:** This is a composite research object (of any kind - repository,
352 repository-record, database, dataset, data-slice, workflow, etc.). Its representation could
353 include scope or knowledge-domain covered, authorship/ownership of the object, latest
354 update, version number, curation policy, and so forth. This metadata may or may not include
355 URLs representing MetaRecord resources (described below) that comprise the individual
356 elements within the composite object. Notably, the Container URL provides a resolvable
357 identifier independent from the identifier of the dataset being described; in fact, the dataset
358 may not have an identifier, as would be the case, for example, where the container
359 represents a dynamically-generated data-slice. In addition, Containers may be published by
360 anyone - that is, the publisher of a Container may be independent from the publisher of the
361 research object it is describing. This enables one of the objectives of our interoperability
362 layer implementation - that anyone can publish metadata about any research object, thus
363 making those objects more FAIR.
364

365 **MetaRecord resource:** This is a specific element within a collection (data point, record,
 366 study, service, etc.). Its representation should include information regarding licensing and
 367 accessibility, access protocols, rich citation information, and other descriptive metadata. It
 368 also includes a reference to the container(s) of which it is a member (the Container URL).
 369 Finally, the MetaRecord may include further URLs that provide direct access to the data
 370 itself, with an explicit reference to the associated data format by its MIME type (e.g.
 371 text/html, application/json, application/vnd.ms-excel, text/csv, etc.). This is achieved using
 372 constructs from the Data Catalogue Vocabulary (DCAT; W3C, 2014), which defines the
 373 concept of a data "Distribution", which includes metadata facets such as the data source
 374 URL and its format. The lower part of Figure 1 diagrams how multiple DCAT Distributions
 375 may be a part of a single MetaRecord. As with Container resources, MetaRecords may be
 376 published by anyone, and independently of the original data publisher.
 377

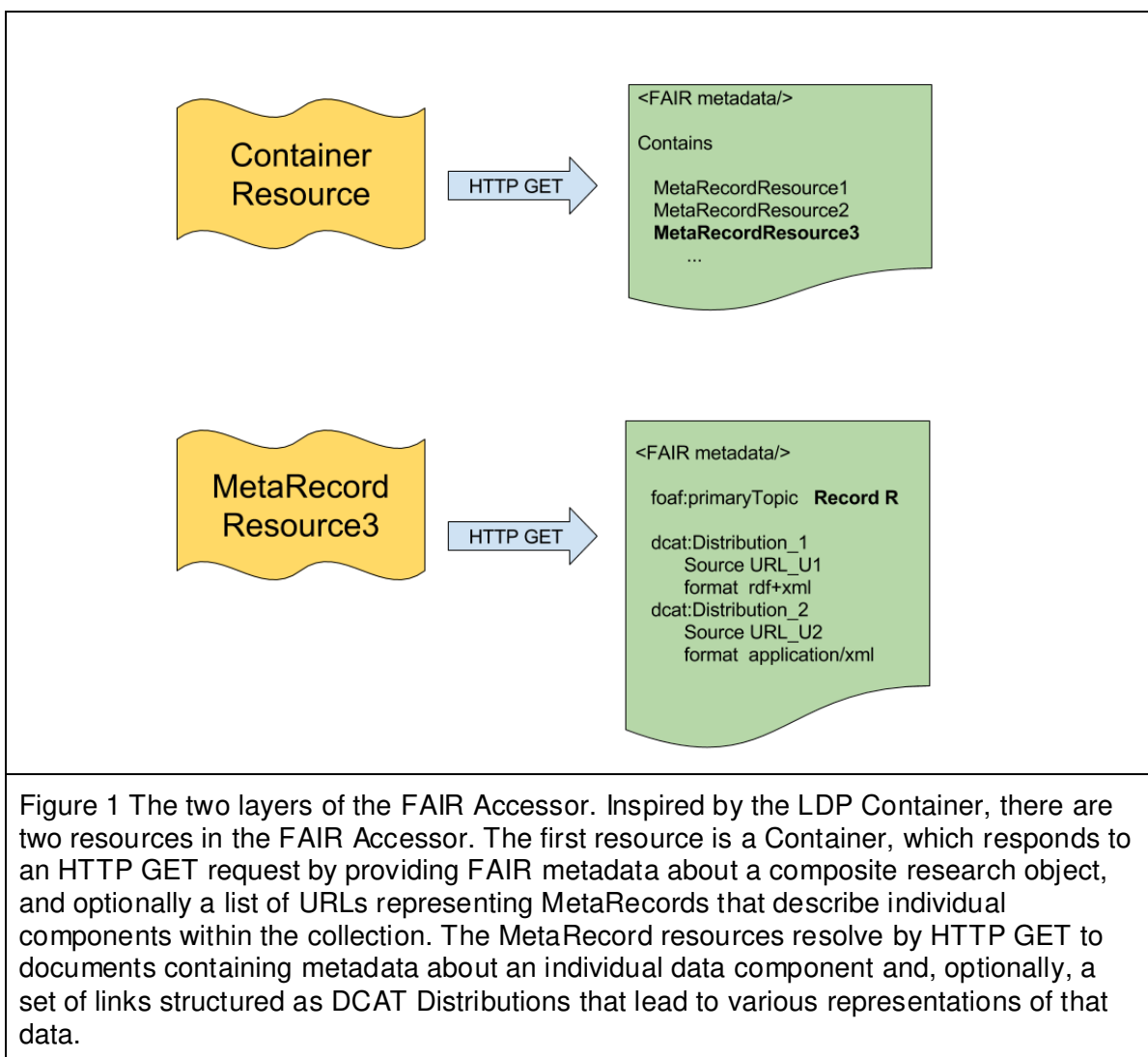


Figure 1 The two layers of the FAIR Accessor. Inspired by the LDP Container, there are two resources in the FAIR Accessor. The first resource is a Container, which responds to an HTTP GET request by providing FAIR metadata about a composite research object, and optionally a list of URLs representing MetaRecords that describe individual components within the collection. The MetaRecord resources resolve by HTTP GET to documents containing metadata about an individual data component and, optionally, a set of links structured as DCAT Distributions that lead to various representations of that data.

378

379 In summary, the FAIR Accessor shares commonalities with the Linked Data Platform, but
 380 additionally recommends the inclusion of rich contextual metadata, based on the FAIR

381 Principles, that facilitate discovery and interoperability of repository and record-level
382 information. The FAIR Accessor is read-only, utilizing only HTTP GET together with widely-
383 used semantic frameworks to guide both human and machine exploration. Importantly, the
384 lack of a novel API means that the information is accessible to generic Web-crawling agents,
385 and may also be processed if that agent “understands” the vocabularies used. Thus, in
386 simplistic terms, the Accessor can be envisioned as a series of Web pages, each containing
387 metadata, and hyperlinks to more detailed metadata and/or data, where the metadata
388 elements and relationships between the pages are explicitly explained to Web crawlers.

389

390 To help clarify this component prior to presenting the more complex components of our
391 interoperability proposal, we will now explore our first use case - data self-archival. A simple
392 FAIR Accessor has been published online (Rodriguez Iglesias et al., 2016) in the Zenodo
393 general-purpose repository. The data self-archival in this citation represents a scenario
394 similar to the polyadenylation use-case described in the Introduction section. In this case,
395 the data describes the evolutionary conservation of components of the RNA Metabolism
396 pathway in fungi as a series of heatmap images. The data deposit, includes a file
397 'RNAME_Accessor.rdf' which acts as the Container Resource. This document includes
398 metadata about the deposit (authorship, topic, etc.), together with a series of 'contains'
399 relationships, referring to MetaRecords inside of the file
400 'RNAME_Accessor_Metarecords.rdf'. Each MetaRecord is about one of the heatmaps, and
401 in addition to metadata about the image, includes a link to the associated image (datatype
402 image/png) and a link to an RDF representation of the same information represented by that
403 image (datatype application/rdf+xml). It should be noted that much of the content of those
404 Accessor files was created using a text editor, based on template RDF documents. The
405 structure of these two documents are described in more detail in the Results section, which
406 includes a full walk-through of a more complex exemplar Accessor.

407

408 At the metadata level, therefore, this portion of the interoperability architecture provides a
409 high degree of FAIRness by allowing machines to discover and interpret useful metadata,
410 and link it with the associated data deposits, even in the case of a repository that provides
411 no FAIR-support. Nevertheless, these components do not significantly enhance the
412 FAIRness and interoperability of the data itself, which was a key goal for this project. We will
413 now describe the application of two recently-published Web technologies - Triple Pattern
414 Fragments and RML - to the problem of data-level interoperability. We will show that these
415 two technologies can be combined to provide an API-free common interface that may be
416 used to serve, in a machine-readable way, FAIR data transformations (either from non-FAIR
417 data, or transformations of FAIR data into novel ontological frameworks). We will also
418 demonstrate how this FAIR data republishing layer can be integrated into the FAIR Accessor
419 to provide a machine-traversable path for incremental drill-down from high-level repository
420 metadata all the way through to individual data points within a record, and back.

421

422 **Data Interoperability: Discovery of compatible data through RML-based FAIR**
423 **Profiles**

424
425 In our approach to data-level interoperability, we first identified a number of desiderata that
426 the solution should exhibit:

- 427
- 428 1. View-harmonization over dissimilar datatypes, allowing discovery of *potentially*
429 integrable data within non-integrable formats.
 - 430 2. Support for a multitude of source data formats (XML, Excel, CSV, JSON, binary,
431 etc.)
 - 432 3. “Cell-level” discovery and interoperability (referring to a “cell” in a spreadsheet)
 - 433 4. Modularity, such that a user can make interoperable only the data component of
434 interest to them
 - 435 5. Reusability, avoiding “one-solution-per-record” and minimizing effort/waste
 - 436 6. Must use standard technologies, and reuse existing vocabularies
 - 437 7. Should not require the participation of the data host (for public data)
- 438

439 The approach we selected was based on the premise that data, in any format, could be
440 modelled as a first step towards interoperability; i.e., the salient data-types and
441 relationships within an opaque data “blob” could be described in a machine-readable
442 manner. The metamodels of two data sources could then be compared to determine if their
443 contained data was, in principle, integrable.

444

445 We referred to these metamodels as “FAIR Profiles”, and we further noted that we should
446 support multiple metamodels of the same data, differing in structure or ontological/semantic
447 framework, within a FAIR Profile. For example, a data record containing blood pressure
448 information might have a FAIR Profile where this facet is modelled using both the SNOMED
449 vocabulary and the ICD10 vocabulary, since the data facet can be understood using either.
450 We acknowledge that these meta-modelling concepts are not novel, and have been
451 suggested by a variety of other projects such as DCAT and Dublin Core (the DC Application
452 Profile (Heery & Patel, 2000), and have been extensively described by the ISO 11179
453 standard for “metadata registries”. It was then necessary to select a modelling framework
454 for FAIR Profiles capable of representing arbitrary, and possibly redundant, semantic
455 models.

456

457 Our investigation into relevant existing technologies and implementations revealed a
458 relatively new, unofficial specification for a generic mapping language called “RDF Mapping
459 Language” (RML (Dimou et al., 2014)). RML is an extension of R2RML (Das, Sundara &
460 Cyganiak, 27 September, 2012), a W3C Recommendation for mapping relational databases
461 to RDF, and is described as “*a uniform mapping formalization for data in different format,*
462 *which [enables] reuse and exchange between tools and applied data*” (Dimou et al., 2014).
463 An RML map describes the triple structure (subject, predicate, object, abbreviated as
464 [S,P,O]), the semantic types of the subject and object, and their constituent URI structures,
465 that would result from a transformation of non-RDF data (of any kind) into RDF data. RML

466 maps are modular documents where each component describes the schema for a single-
467 resource-centric graph (i.e. a graph with all triples that share the same subject). The “object”
468 position in each of these map modules may be mapped to a literal, or may be mapped to
469 another RML module, thus allowing linkages between maps in much the same way that the
470 object of an RDF triple may become the subject of another triple. RML modules therefore
471 may then be assembled into a complete map representing both the structure and the
472 semantics of an RDF representation of a data source. RML maps themselves take the form
473 of RDF documents, and can be published on the Web, discovered, and reused, via standard
474 Web technologies and protocols. RML therefore fulfils each of the desiderata for FAIR
475 Profiles, and as such, we selected this technology as the candidate for their implementation.
476 Comparing with related technologies, this portion of our interoperability prototype serves a
477 similar purpose to the XML Schema (XSD; Fallside & Walmsley, 2004) definitions within the
478 output component of a Web Services Description Language (WSDL) document, but unlike
479 XSD, is capable of describing the structure and semantics of RDF graphs.

480
481 Of particular interest to us was the modularity of RML - its ability to model individual triples.
482 This speaks directly to our desiderata 4, where we do not wish to require (and should not
483 expect) a modeller to invest the time and effort required to fully model every facet of a
484 potentially very complex dataset. Far more often, individuals will have an interest in only one
485 or a few facets of a dataset. As such, we chose to utilize RML models at their highest level
486 of granularity - that is, we require a distinct RML model for each triple pattern (subject+type,
487 predicate, object+type) of interest. We call these small RML models "Triple Descriptors".
488 An exemplar Triple Descriptor is diagrammed in Figure 2. There may be many Triple
489 Descriptors associated with a single data resource. Moreover, multiple Triple Descriptors
490 may model the same facet within that data resource, using different URI structures,
491 subject/object semantic types, or predicates, thus acting as different "views" of that data
492 facet. Finally, then, the aggregation of all Triple Descriptors associated with a specific data
493 resource produces a FAIR Profile of that data. Note that FAIR Profiles are not necessarily
494 comprehensive; however, by aggregating the efforts of all modellers, FAIR Profiles model
495 only the data facets that are most important to the community.

496
497 FAIR Profiles enable view harmonization over compatible but structurally non-integrable
498 data, possibly in distinct repositories. The Profiles of one data resource can be compared to
499 the Profiles of another data resource to identify commonalities between their Triple
500 Descriptors at the semantic level, even if the underlying data is semantically opaque and/or
501 structurally distinct - a key step toward Interoperability. FAIR Profiles, therefore, have utility,
502 independent of any *actuated* transformation of the underlying data, in that they facilitate
503 compatible data discovery. Moreover, with respect to desiderata 5, Triple Descriptors, and
504 sometimes entire FAIR Profiles, are RDF documents published on the Web, and therefore
505 may be reused to describe new data resources, anywhere on the Web, that contain similar
506 data elements, regardless of the native representation of that new resource, further
507 simplifying the goal of data harmonization.

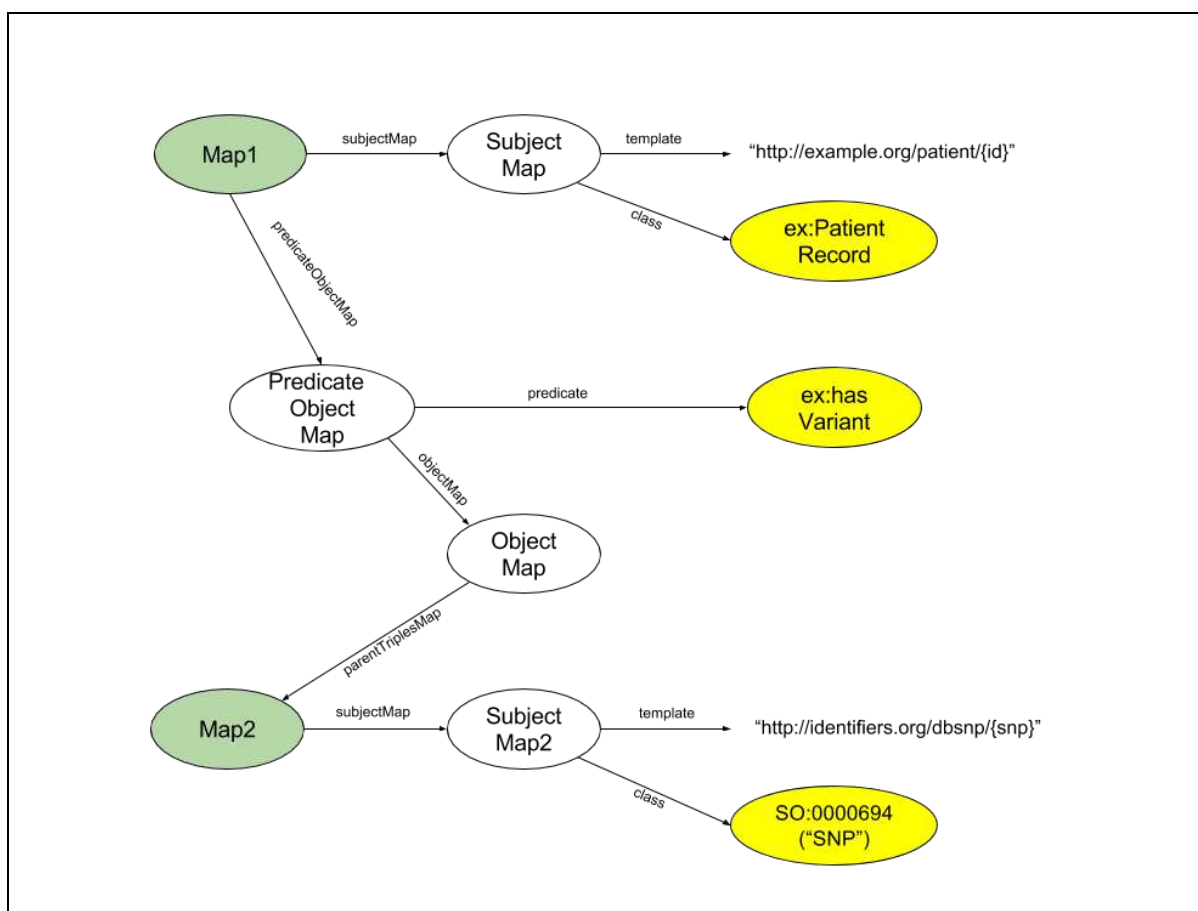


Figure 2: Diagram of the structure of an exemplar Triple Descriptor representing a hypothetical record of a SNP in a patient's genome. In this descriptor, the Subject will have the URL structure *http://example.org/patient/{id}*, and the Subject is of type PatientRecord. The Predicate is hasVariant, and the Object will have URL structure *http://identifiers.org/dbsnp/{snp}* with the *rdf:type* from the sequence ontology "0000694" (which is the concept of a "SNP"). The two nodes shaded green are of the same ontological type, showing the iterative nature of RML, and how individual RML Triple Descriptors will be concatenated into full FAIR Profiles. The three nodes shaded yellow are the nodes that define the subject type, predicate and object type of the triple being described.

508

509

510 Data Interoperability: Data transformation with FAIR Projectors and Triple 511 Pattern Fragments

512

513 The ability to identify *potentially* integrable data within opaque file formats is, itself, a notable
514 achievement compared to the *status quo*. Nevertheless, beyond just discovery of relevant
515 data, our interoperability layer aims to support and facilitate cross-resource data integration
516 and query answering. This requires that the data is not only semantically described, but is
517 also semantically and syntactically transformed into a common structure.

518

519 Having just presented a mechanism to describe the structure and semantics of data - Triple
520 Descriptors in RML - what remains lacking is a way to retrieve data consistent with those
521 Triple Descriptors. We require a means to expose transformed data without worsening the
522 existing critical barrier to interoperability - opaque, non-machine-readable interfaces and API
523 proliferation (Verborgh & Dumontier, 2016). What is required is a universally-applicable way
524 of retrieving data generated by a (user-defined) data extraction or transformation process,
525 that does not result in yet another API.

526

527 The Triple Pattern Fragments (TPF) specification (Verborgh et al., 2016) defines a REST
528 interface for publishing triples. The server receives HTTP GET calls on URLs that contain a
529 triple pattern [S,P,O], where any component of that pattern is either a constant or a variable.
530 In response, a TPF server returns pages with all triples from its data source that match the
531 incoming pattern. As such, any given triple pattern has a distinct URL.

532

533 We propose, therefore, to combine three elements - data transformed into RDF, which is
534 described by Triple Descriptors, and served via TPF-compliant URLs. We call this
535 combination of technologies a "FAIR Projector". A FAIR Projector, therefore, is a Web
536 resource (i.e., something identified by a URL) that is associated with both a particular data
537 source, and a particular Triple Descriptor. Calling HTTP GET on the URL of the FAIR
538 Projector produces RDF triples from the data source that match the format defined by that
539 Projector's Triple Descriptor. The originating data source behind a Projector may be a
540 database, a data transformation script, an analytical web service, another FAIR Projector, or
541 any other static or dynamic data-source. Note that we do not include a transformation
542 methodology in this proposal; however, we address this issue and provide suggestions in the
543 Discussion section. There may, of course, be multiple projectors associated with any given
544 data source, serving a variety of triples representing different facets of that data.

545

546 **Linking the Components: FAIR Projectors and the FAIR Accessor**

547

548 At this point, we have a means for requesting triples with a particular structure - TPF Servers
549 - and we have a means of describing the structure and semantics of those triples - Triple
550 Descriptors. Together with a source of RDF data, these define a FAIR Projector. However,
551 we still lack a formal mechanism for linking TPF-compliant URLs with their associated Triple
552 Descriptors, such that the discovery of a Triple Descriptor with the desired semantics for a
553 particular data resource, also provides its associated Projector URL.

554

555 We propose that this association can be accomplished, without defining any novel API or
556 standard, if the output of a FAIR Projector is considered a DCAT Distribution of a particular
557 data source, and included within the MetaRecord of a FAIR Accessor. The URL of the
558 Projector, and its Triple Descriptor, become metadata facets of a new `dcat:Distribution`
559 element in the MetaRecord. This is diagrammed in Figure 3, where `Distribution_3` and
560 `Distribution_4` include Triple Pattern Fragment-formatted URLs representing the FAIR
561 Projector, and the Triple Descriptor RML model describing the structure and semantics of
562 the data returned by calling that Projector.

563

564 Thus, all components of this interoperability system - from the top level repository metadata,
 565 to the individual data cell - are now associated with one another in a manner that allows
 566 mechanized data discovery, harmonization, and retrieval, including relevant citation
 567 information. No novel technology or API was required, thus allowing this rich combination of
 568 data and metadata to be explored using existing Web tools and crawlers.

569

570

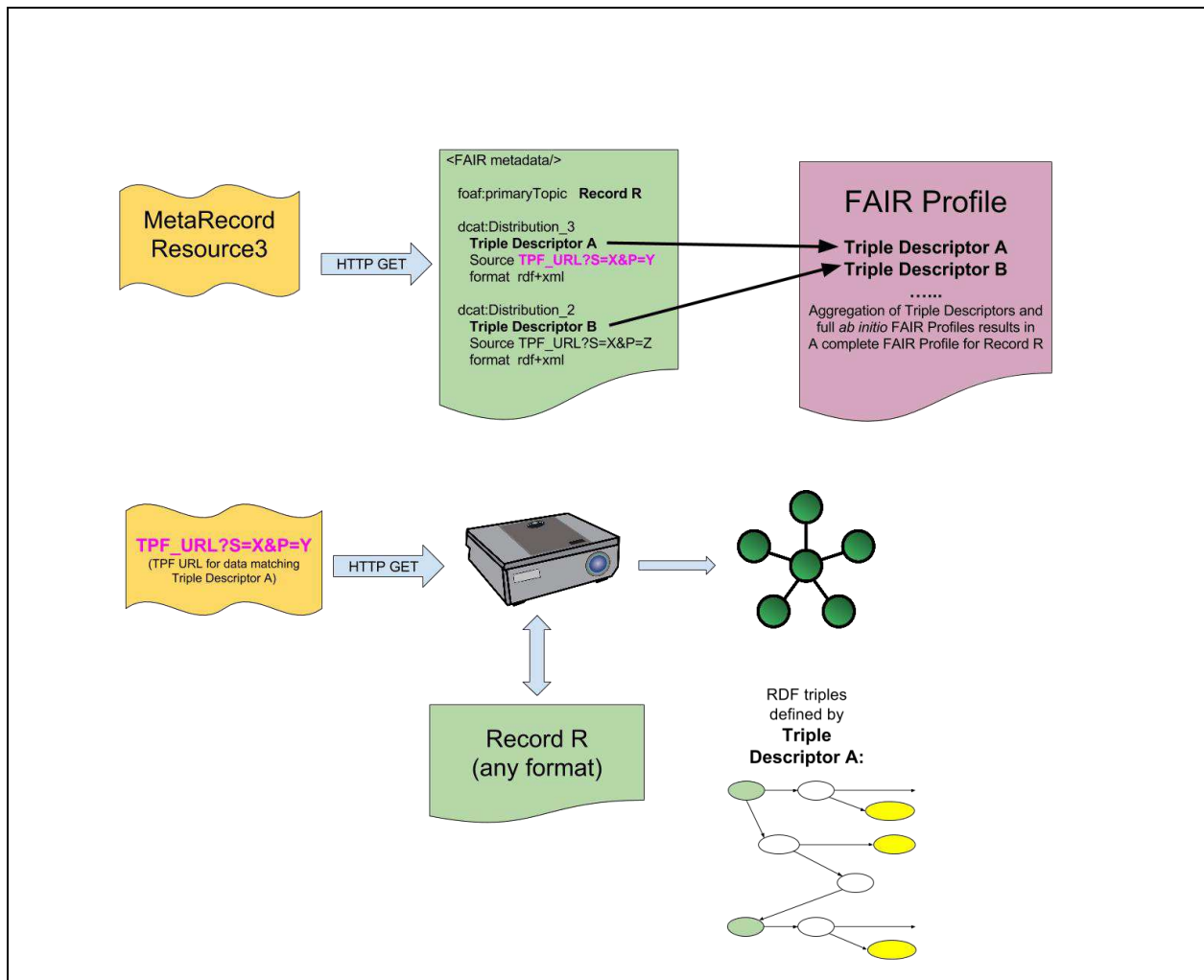


Figure 3. Integration of FAIR Projectors into the FAIR Accessor. Resolving the MetaRecord resource returns a metadata document containing multiple DCAT Distributions for a given record, as in Figure 1. When a FAIR Projector is available, additional DCAT Distributions are included in this metadata document. These Distributions contain a URL (purple text) representing a Projector, and a Triple Descriptor that describes, in RML, the structure and semantics of the Triple(s) that will be obtained from that Projector resource if it is resolved. These Triple Descriptors may be aggregated into FAIR Profiles, based on the Record that they are associated with (Record R, in the figure) to give a full mapping of all available representations of the data present in Record R.

571

572

573 Results

574

575 In the previous section, we provided the URL to a simple exemplar FAIR Accessor published
576 on Zenodo. To demonstrate the interoperability system in its entirety - including both the
577 Accessor and the Projector components - we will now proceed through a second exemplar
578 involving the special-purpose repository for protein sequence information, UniProt. In this
579 example, we examine a FAIR Accessor to a dataset, created through a database query, that
580 consists of a specific “slice” of the Protein records within the UniProt database - that is, the
581 set of proteins in *Aspergillus nidulans* FGSC A4 (NCBI Taxonomy ID 227321) that are
582 annotated as being involved in mRNA Processing (Gene Ontology Accession GO:0006397).
583 We first demonstrate the functionality of the two layers of the FAIR Accessor in detail. We
584 then demonstrate a FAIR Projector, and show how its metadata integrates into the FAIR
585 Accessor. In this example, the Projector modifies the ontological framework of the UniProt
586 data such that the ontological terms used by UniProt are replaced by the terms specified in
587 EDAM - an ontology of bioinformatics operations, datatypes, and formats (Ison et al., 2013).
588 We will demonstrate that this transformation is specified, in a machine-readable way, by the
589 FAIR Triple Descriptor that accompanies each Projector’s metadata.

590

591 The two-step FAIR Accessor

592

593 The example FAIR Accessor accesses a database of RDF hosted by UniProt, and issues
594 the following query over that database (expressed in the standard RDF query language
595 SPARQL):

596

```
597 PREFIX up:<http://purl.uniprot.org/core/>  
598 PREFIX taxon:<http://purl.uniprot.org/taxonomy/>  
599 PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#>  
600 PREFIX GO:<http://purl.obolibrary.org/obo/GO_>  
601 SELECT DISTINCT ?id  
602  
603 WHERE  
604 {  
605     ?protein a up:Protein ;  
606     up:organism taxon:227321 ;  
607     up:classifiedWith/rdfs:subClassOf GO:0006397 .  
608     BIND(substr(str(?protein), 33) as ?id)  
609 }
```

610

611 Accessor output is retrieved from the Container Resource URL:

612

```
613 http://linkeddata.systems/Accessors/UniProtAccessor
```

614

615 The result of calling GET on the Container Resource URL is visualized in Figure 4, where
 616 Tabulator (Tim Berners-lee et al., 2006) is used to render the output as HTML for human-
 617 readability.

618
 619

UniProt Slice FAIR Accessor - Aspergillus RNA Processing proteins	creator	wilkinsonlab.info/
	language	eng
	license	4.0/
	title	UniProt Slice FAIR Accessor - Aspergillus RNA Processing proteins
	authored By	0000 0002 9699 485X
	version	UniProt release 2016_09
	entities	82
	term has Principal Investigator	Dr. Mark Wilkinson
	type	Dataset Basic Container Collection
	contact Point	Wilkinson.rdf
	description	Takes a SPARQL query of the UniProt database specific to proteins and their GO annotations related to RNA Processing proteins in Aspergillus and makes it a FAIR Accessor source. The query being executed is: <pre>PREFIX up:<http://purl.uniprot.org/core/> PREFIX taxon:<http://purl.uniprot.org/taxonomy/> PREFIX rdf:<http://www.w3.org/1999/02/22-rdf-syntax-ns#> PREFIX GO:<http://purl.obolibrary.org/obo/GO_> SELECT DISTINCT ?id WHERE { ?protein a up:Protein ; up:organism taxon:227321 ; up:classifiedWith/rdfs:subClassOf GO:0006397 . BIND(substr(str(?protein), 33) as ?id) }</pre>
	identifier	Uni Prot Accessor
	keyword	Aspergillus nidulans Aspergillus Proteins RNA Processing
	landing Page	uniprot.org/
	language	en
	publisher	wilkinsonlab.info/
	theme	RNA Processing conceptscheme.rdf
	contains	C8V1L6 C8V2B3 C8V609 C8V6T1 C8V8F9 C8V9C2 C8V9G4 C8VBS9 C8VBV2

Figure 4. A representative portion of the output from resolving the Container Resource of the FAIR Accessor, rendered into HTML by the Tabulator Firefox plugin. The three columns show the label of the Subject node of all RDF Triples (left), the label of the URI in the predicate position of each Triple (middle), and the value of the Object position (right), where blue text indicates that the value is a Resource, and black text indicates that the value is a literal.

620
 621
 622
 623
 624

Of particular note are the following metadata elements:

http://purl.org/dc/elements/1.1/license	https://creativecommons.org/licenses/by-nd/4.0/
http://purl.org/pav/authoredBy	http://orcid.org/0000-0002-9699-485X
http://rdfs.org/ns/void#entities	82
a	http://purl.org/dc/dcmitype/Dataset http://www.w3.org/ns/ldp#BasicContainer http://www.w3.org/ns/prov#Collection
http://www.w3.org/ns/dcat#contactPoint	http://biordf.org/DataFairPort/MiscRDF/Wilkinson.rdf
http://www.w3.org/ns/dcat#keyword	"Aspergillus nidulans", "Aspergillus", "Proteins", "RNA Processing";
http://www.w3.org/ns/dcat#theme	http://linkeddata.systems/ConceptSchemes/RNA_Processing_conceptscheme.rdf
http://www.w3.org/ns/ldp#contains	http://linkeddata.systems/cgi-bin/Accessors/UniProtAccessor/C8VIL6 http://linkeddata.systems/cgi-bin/Accessors/UniProtAccessor/C8V2B3 ...

625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645

- License information is provided as an HTML + RDFa document, following one of the primary standard license forms published by Creative Commons. This allows the license to be unambiguously interpreted by both machines and people prior to accessing any data elements, an important feature that will be discussed later.
- Authorship is provided by name, using the Academic Research Project Funding Ontology (ARPFO), but is also unambiguously provided by a link to the author's ORCID, using the Provenance Authoring and Versioning (PAV; Ciccarese et al, 2013) ontology.
- The repository descriptor is typed as being a Dublin Core Dataset, a Linked Data Platform container, and a Provenance Collection, allowing it to be interpreted by a variety of client agents, and conforming to several best-practices, such as the Healthcare and Life Science Dataset Description guidelines (Gray et al., 2015; Dumontier et al., 2016)
- Contact information is provided in a machine-readable manner via the Friend of a Friend (Foaf) record of the author, and the DCAT ontology "contactPoint" property.
- Human readable keywords, using DCAT, are mirrored and/or enhanced by a machine-readable RDF document which is the value of the DCAT "theme" property. This RDF document follows the structure determined by the Simple Knowledge Organization System (SKOS) ontology, and lists the ontological terms that describe the repository for machine-processing.

- 646 • Finally, individual records within the dataset are represented as the value of the
 647 Linked Data Platform “contains” property, and provided as a possibly paginated list of
 648 URLs (a discussion of machine-actionable pagination will not be included here).
 649 These URLs are the MetaRecord Resource URLs shown in Figure 1.

650
 651

652 Following the flow in Figure 1, the next step in the FAIR Accessor is to resolve a
 653 MetaRecord Resource URL. For clarity, we will first show the metadata document that is
 654 returned if there are no FAIR Projectors for that dataset. This will be similar to the document
 655 returned by calling a FAIR MetaRecord URL in the Zenodo use case discussed in the earlier
 656 Methods section.

657

658 Calling HTTP GET on a MetaRecord Resource URL returns a document that include
 659 metadata elements and structure shown in Figure 5. Note that Figure 5 is not the complete
 660 MetaRecord; rather it has been edited to include only those elements relevant to the aspects
 661 of the interoperability infrastructure that have been discussed so far. More complete
 662 examples of the MetaRecord RDF, including the elements describing a Projector, are
 663 described in Figures 7, 8, and 9.

664
 665

UniProt Protein C8V1L6	bibliographic Citation	The UniProt Consortium (2015). UniProt: a hub for protein information. Nucleic Acids Res. 43: D204-D212
	creator	UniProt Consortium
	language	eng
	license	3.0/
	title	UniProt Protein C8V1L6
	Version	UniProt release 2016_09
	in dataset	Uni Prot Accessor/
	contact point	contact
	description	Splicing factor u2af large subunit (AFU_orthologue AFUA_7G05310)
	distribution	Distribution D7566F52 C143 11E6 897C 26245D07C3DD Distribution D75682F8 C143 11E6 897C 26245D07C3DD
	identifier	C8V1L6
	keyword	Annotation Aspergillus nidulans Aspergillus Functional Annotation GO Gene Ontology Proteins RNA Processing
	landing page	uniprot.org
	language	en
	publisher	uniprot.org
	page	sparql uniprot.org/
	primary topic	C8V1L6
C8V1L6	...	
Distribution D7566F52 C143 11E6 897C 26245D07C3DD	format	text/html
	type	Dataset
	download URL	Distribution C8V1L6.html
Distribution D75682F8 C143 11E6 897C 26245D07C3DD	format	application/rdf+xml
	type	Dataset
	download URL	dataset Distribution C8V1L6.rdf

Figure 5. A representative (incomplete) portion of the output from resolving the MetaRecord Resource

of the FAIR Accessor for record C8V1L6 (at <http://linkeddata.systems/Accessors/UniProtAccessor/C8V1L6>), rendered into HTML by the Tabulator Firefox plugin. The columns have the same meaning as in Figure 4.

666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684

Many properties in this metadata document are similar to those at the higher level of the FAIR Accessor. Notably, however, the primary topic of this document is the UniProt record, indicating a shift in the focus of the document from the provider of the Accessor to the provider of the originating Data. Therefore, the values of these facets now reflect the authorship and contact information for that record. We do, recognize that MetaRecords are themselves scholarly works and should be properly cited. The MetaRecord includes the “in dataset” predicate, which refers back to the first level of the FAIR Accessor, thus this provides one avenue for capturing the provenance information for the MetaRecord. If additional provenance detail is required, we propose (but do not describe further here) that this information could be contained in a separate named graph, in a manner akin to that used by NanoPublications (Kuhn et al., 2016).

The important distinctive property in this document is the “distribution” property, from the DCAT ontology. For clarity, an abbreviated document in Turtle format is shown in Figure 6, containing only the “distribution” elements and their values.

```
@prefix dc: <http://purl.org/dc/elements/1.1/>.
@prefix dcat: <http://www.w3.org/ns/dcat#>.
@prefix Uni: <http://linkeddata.systems/Accessors/UniProtAccessor/>.

Uni:C8V1L6
  dcat:distribution
    <#DistributionD7566F52-C143-11E6-897C-26245D07C3DD>,
    <#DistributionD75682F8-C143-11E6-897C-26245D07C3DD>;
<#DistributionD7566F52-C143-11E6-897C-26245D07C3DD>
  dc:format
    "text/html";
  a dc:Dataset, dcat:Distribution;
  dcat:downloadURL
    <http://www.uniprot.org/uniprot/C8V1L6.html>.
<#DistributionD75682F8-C143-11E6-897C-26245D07C3DD>
  dc:format
    "application/rdf+xml";
  a dc:Dataset, void:Dataset, dcat:Distribution;
  dcat:downloadURL
    <http://www.uniprot.org/uniprot/C8V1L6.rdf>.
```

Figure 6. Turtle representation of the subset of triples from the MetaRecord metadata pertaining to the two DCAT Distributions. Each distribution specifies an available representation (media type), and a URL from which that representation can be downloaded.

685

686

687 There are two DCAT Distributions in this document. The first is described as being in format
688 “application/rdf+xml”, with its associated download URL. The second is described as being
689 in format “text/html”, again with the correct URL for that representation. Both are typed as
690 Distributions from the DCAT ontology. These distributions are published by UniProt
691 themselves, and the UniProt URLs are used. Additional metadata in the FAIR Accessor (not
692 shown in Figure 6) describes the keywords that relate to that record in both machine and
693 human-readable formats, access policy, and license, allowing machines to more accurately
694 determine the utility of this record prior to retrieving it.

695

696 Several things are important to note before moving to a discussion of FAIR Projectors. First,
697 the two levels of the FAIR Accessor are not interdependent. The Container layer can
698 describe relevant information about the scope and nature of a repository, but might not
699 provide any further links to MetaRecords. Similarly, whether or not to provide a distribution
700 within a MetaRecord is entirely at the discretion of the data owner. For sensitive data, an
701 owner may chose to simply provide (even limited) metadata, but not provide any direct link to
702 the data itself, and this is perfectly conformant with the FAIR guidelines. Further, when
703 publishing a single data record, it is not obligatory to publish the Container level of the FAIR
704 Accessor; one could simply provide the MetaRecord document describing that data file,
705 together with an optional link to that file as a Distribution. Finally, it is also possible to publish
706 containers of containers, to any depth, if such is required to describe a multi-resource
707 scenario (e.g. an institution hosting multiple distinct databases).

708

709 **The FAIR Projector**

710

711 FAIR Projectors can be used for many purposes, including (but not limited to) publishing
712 transformed Linked Data from non-Linked Data; publishing transformed data from a Linked
713 Data source into a distinct structure or ontological framework; load-management/query-
714 management; or as a means to explicitly describe the ontological structure of an underlying
715 data source in a searchable manner. In this demonstration, the FAIR Projector publishes
716 dynamically transformed data, where the transformation involves altering the semantics of
717 RDF provided by UniProt into a different ontological framework (EDAM).

718

719 This FAIR Projector’s TPF interface is available at:

720

721 `http://linkeddata.systems:3001/fragments`

722

723 Data exposed as a TPF-compliant Resource require a subject and/or predicate and/or object
 724 value to be specified in the URL; a request for the all-variable pattern (blank, as above) will
 725 return nothing. How can a software agent know what URLs are valid, and what will be
 726 returned from such a request?

727

728 In this interoperability infrastructure, we propose that Projectors should be considered as
 729 DCAT Distributions, and thus TPF URLs, with appropriate parameters bound, are included in
 730 the distribution section of the MetaRecord metadata. An example is shown in Figure 7, again
 731 rendered using Tabulator.

732

733

UniProt Protein C8V1L6	bibliographic Citation	The UniProt Consortium (2015). UniProt: a hub for protein information. Nucleic Acids Res. 43: D204-D212
	creator	UniProt Consortium
	language	eng
	license	3.0/
	title	UniProt Protein C8V1L6
	Version	UniProt release 2016_09
	in dataset	Uni Prot Accessor/
	contact point	contact
	description	Splicing factor u2af large subunit (AFU_orthologue AFUA_7G05310)
	distribution	Distribution9E275EC2 C1F6 11E6 8812 3E445D07C3DD Distribution9E2771E6 C1F6 11E6 8812 3E445D07C3DD Distribution9EFD1238 C1F6 11E6 8812 3E445D07C3DD Distribution9EFD2458 C1F6 11E6 8812 3E445D07C3DD
	identifier	C8V1L6
	keyword	Annotation Aspergillus nidulans Aspergillus Funcinal Annotation GO Gene Ontology Proteins RNA Processing
	landing page	uniprot.org
	language	en
	publisher	uniprot.org
	page	sparql
	primary topic	uniprot.org/ C8V1L6

Figure 7. A portion of the output from resolving the MetaRecord Resource of the FAIR Accessor for record C8UZX9, rendered into HTML by the Tabulator Firefox plugin. The columns have the same meaning as in Figure 4. Comparing the structure of this document to that in Figure 5 shows that there are now four values for the “distribution” predicate. An RDF and HTML representation, as in Figure 5, and two additional distributions with URLs conforming to the TPF design pattern (highlighted).

734

735

736 Note that there are now four distributions - two of them are the html and rdf distributions
 737 discussed above (Figure 5). The two new distributions are those provided by a FAIR
 738 Projector. Again, looking at an abbreviated and simplified Turtle document for clarity (Figure
 739 8) we can see the metadata structure of one of these two new distributions.

740

```
@prefix dc: <http://purl.org/dc/elements/1.1/>.
@prefix dcat: <http://www.w3.org/ns/dcat#>.
```



```
rr:class ed:data_1176; rr:template "http://identifiers.org/taxon/{TAX}" .
```

Figure 8. Turtle representation of the subset of triples from the MetaRecord metadata pertaining to one of the FAIR Projector DCAT Distributions of the MetaRecord shown in Figure 7. The text is colour-coded to assist in visual exploration of the RDF. The DCAT Distribution blocks of the two Projector distributions (black bold) have multiple media-type representations (red), and are connected to an RML Map (Dark blue) by the hasMapping predicate, which is a block of RML that semantically describes the subject, predicate, and object (green, orange, and purple respectively) of the Triple Descriptor for that Projector. This block of RML is schematically diagrammed in Figure 2. The three media-types (red) indicate that the URL will respond to HTTP Content Negotiation, and may return any of those three formats.

741
742 Following the Triple Pattern Fragments behaviour, requesting the downloadURL with HTTP
743 GET will trigger the Projector to restrict its output to only those data from UniProt where the
744 subject is UniProt record C8V1L6, and the property of interest is “classifiedWith” from the
745 UniProt Core ontology. The triples returned in response to this call, however, will not match
746 the native semantics of UniProt, but rather will match the semantics and structure defined in
747 the RML Mappings block. The schematic structure of this Mapping RML is diagrammed in
748 Figure 2. The Mappings describes a Triple where the subject will be of type
749 `edam:data_0896` (“Protein record”), the predicate will be “classifiedWith” from the
750 UniProt Core ontology, and the object will be of type `edam:data_1176` (“GO Concept ID”).

751
752 Specifically, the triples returned are:

```
753  
754 @prefix uni: <http://identifiers.org/uniprot/>.  
755 @prefix obo: <http://purl.obolibrary.org/obo/>.  
756 uni:C8V1L6 core:classifiedWith obo:GO_0000245, obo:GO_0045292 .
```

757
758 This is accompanied by a block of hypermedia controls (not shown) using the Hydra
759 vocabulary (Lanthaler & Gütl; Das, Sundara & Cyganiak, 27 September, 2012) that provide
760 machine-readable instructions for how to navigate the remainder of that dataset - for
761 example, how to get the entire row, or the entire column for the current data-point.

762
763 Though the subject and object are not explicitly typed in the output from this call to the
764 Projector, further exploration of the Projector’s output, via those TPF’s hypermedia controls,
765 would reveal that the Subject and Object are in fact typed according to the EDAM ontology,
766 as declared in the RML Mapping. Thus, this FAIR Projector served data transformed from
767 UniProt Core semantic types, to the equivalent data represented within the EDAM semantic
768 framework, as shown in Figure 9. Also note that the URI structure for the UniProt entity has
769 been changed from the UniProt URI scheme to a URI following the Identifiers.org scheme.

770
771 The FAIR Projector, in this case, is a script that dynamically transforms data from a query of
772 UniProt into the appropriately formatted triples; however, this is opaque to the client. The
773 Projector’s TPF interface, from the perspective of the client, would be identical if the
774 Projector was serving pre-transformed data from a static document, or even generating
775

776 novel data from an analytical service. Thus, FAIR Projectors harmonize the interface to
 777 retrieving RDF data in a desired semantic/structure, regardless of the underlying mechanism
 778 for generating that data.

779

780 This example was chosen for a number of reasons. First, to contrast with the static Zenodo
 781 example provided earlier, where this Accessor/Projector combination are querying the
 782 UniProt database dynamically. In addition, because we wished to demonstrate the utility of
 783 the Projector's ability to transform the semantic framework of existing FAIR data in a
 784 discoverable way. For example, in UniProt, Gene Ontology terms do not have a richer
 785 semantic classification than "owl:Class". With respect to interoperability, this is problematic,
 786 as the lack of rich semantic typing prevents them from being used for automated discovery
 787 of resources that could potentially consume them, or use them for integrative, cross-domain
 788 queries. This FAIR Accessor/Projector advertises that it is possible to obtain EDAM-
 789 classified data, from UniProt, simply by resolving the Projector URL.

790

791

In UniProt	<pre> http://purl.uniprot.org/uniprot/C8UZX9 a http://purl.uniprot.org/core/Protein ; http://purl.uniprot.org/core/classifiedWith http://purl.obolibrary.org/obo/GO_0000462 . http://purl.obolibrary.org/obo/GO_0000462 a http://www.w3.org/2002/07/owl#Class </pre>
After Projection	<pre> http://identifiers.org/uniprot/C8UZX9 a http://edamontology.org/data_0896 ; http://purl.uniprot.org/core/classifiedWith http://purl.obolibrary.org/obo/GO_0000462 . http://purl.obolibrary.org/obo/GO_0000462 a http://edamontology.org/data_1176 </pre>

Figure 9: Data before and after FAIR Projection. Bolded segments show how the URI structure and the semantics of the data were modified, according to the mapping defined in the Triple Descriptor (data_0896 = "Protein report" and data_1176 = "GO Concept ID"). URI structure transformations may be useful for integrative queries against datasets that utilize the Identifiers.org URI scheme such as OpenLifeData (González et al., 2014). Semantic transformations allow integrative queries across datasets that utilize diverse and redundant ontologies for describing their data, and in this example, may also be used to add semantics where there were none before.

792

793 Discussion

794

795 Interoperability is hard. It was immediately evident that, of the four FAIR principles,
796 Interoperability was going to be the most challenging. Here we have designed a novel
797 infrastructure with the primary objective of interoperability for both metadata and data, but
798 with an eye to all four of the FAIR Principles. We wished to provide discoverable and
799 interoperable access to a wide range of underlying data sources - even those in
800 computationally opaque formats - as well as supporting a wide array of both academic and
801 commercial end-user applications above these data sources. In addition, we imposed
802 constraints on our selection of technologies; in particular, that the implementation should re-
803 use existing technologies as much as possible, and should support multiple and
804 unpredictable end-uses. Moreover, it was accepted from the outset that the trade-off
805 between simplicity and power was one that could not be avoided, since a key objective was
806 to maximize uptake over the broadest range of data repositories, spanning all domains - this
807 would be nearly impossible to achieve through, for example, attempting to impose a
808 'universal' API or novel query language. Thus, with the goal of maximizing global uptake and
809 adoption of this interoperability infrastructure, and democratizing the cost of implementation
810 over the entire stakeholder community - both users and providers - we opted for lightweight,
811 weakly integrative, REST solutions, that nevertheless lend themselves to significant degrees
812 of mechanization in both discovery and integration.

813

814 We now look more closely at how this interoperability infrastructure meets the expectations
815 within the FAIR Principles.

816

817 **FAIR facet(s) addressed by the Container Resource:**

818

- 819 ● **Findable** - The container has a distinct globally unique and resolvable
820 identifier, allowing it to be discovered and explicitly, unambiguously cited. This
821 is important because, in many cases, the dataset being described does not
822 natively possess an identifier, as in our example above where the dataset
823 represented the results of a query. In addition, the container's metadata
824 describes the research object, allowing humans and machines to evaluate the
825 potential utility of that object for their task.

826

- 827 ● **Accessible** - the Container URL resolves to a metadata record using
828 standard HTTP GET. In addition to describing the nature of the research
829 object, the metadata record should include information regarding licensing,
830 access restrictions, and/or the access protocol for the research object.
831 Importantly, the container metadata exists independently of the research
832 object it describes, where FAIR Accessibility requires metadata to be
833 persistently available even if the data itself is not.

834

- 835 ● **Interoperable** - The metadata is provided in RDF - a globally-applicable
syntax for data and knowledge sharing. In addition, the metadata uses
shared, widely-adopted public ontologies and vocabularies to facilitate
interoperability at the metadata level.

- 836
- **Reusable** - the metadata includes citation information related to the authorship of the container and/or its contents, and license information related to the reuse of the data, by whom, and for what purpose.
- 837
- 838
- 839

840 **Other features of the Container Resource**

- **Privacy protection** - The container metadata provides access to a rich description of the content of a resource, without exposing any data within that resource. While a provider may choose to include MetaRecord URLs within this container, they are not required to do so if, for example, the data is highly sensitive, or no longer easily accessible; however, the contact information provided within the container allows potential users of that data to inquire as to the possibility of gaining access in some other way. As such, this container facilitates a high degree of FAIRness, while still providing a high degree of privacy protection.
- 841
- 842
- 843
- 844
- 845
- 846
- 847
- 848
- 849

850 **FAIR Facet(s) Addressed by the MetaRecord:**

- **Findable** - The MetaRecord URL is a globally-unique and resolvable identifier for a data entity, regardless of whether or not it natively possesses an identifier. The metadata it resolves to allows both humans and machines to interrogate the nature of a data element before deciding to access it.
 - **Accessible** - the metadata provided by accessing the MetaRecord URL describes the accessibility protocol and license information for that record, and describes all available formats.
 - **Interoperable** - as with the Container metadata, the use of shared ontologies and RDF ensures that the metadata is interoperable.
 - **Reusable** - the MetaRecord metadata should carry record-level citation information to ensure proper attribution if the data is used. We further propose, but do not demonstrate, that authorship of the MetaRecord itself could be carried in a second named-graph, in a manner similar to that proposed by the NanoPublication specification.
- 851
- 852
- 853
- 854
- 855
- 856
- 857
- 858
- 859
- 860
- 861
- 862
- 863
- 864
- 865
- 866

867 **Other features of the MetaRecord**

- **Privacy protection** - the MetaRecord provides for rich descriptive information about a specific member of a collection, where the granularity of that description is entirely under the control of the data owner. As such, the MetaRecord can provide a high degree of FAIRness at the level of an individual record, without necessarily exposing any identifiable information. In addition, the provider may choose to stop at this level of FAIRness, and not include further URLs giving access to the data itself.
 - **Symmetry of traversal** - Since we predict that clients will, in the future, query over indexes of FAIR metadata searching for dataset or records of interest, it is not possible to predict the position at which a client or their agent will enter your FAIR Accessor. While the container metadata provides links to individual MetaRecords, the MetaRecord similarly provides a reference back “upwards”
- 868
- 869
- 870
- 871
- 872
- 873
- 874
- 875
- 876
- 877
- 878
- 879

880 to its container. Thus a client can access repository-level metadata (e.g.
881 curation policy, ownership, linking policy) for any given data element it
882 discovers. This became particularly relevant as a result of the European Court
883 of Justice decision (Court of Justice, 2016) that puts the burden of proof on
884 those who create hyperlinks to ensure the document they link to is not, itself,
885 in violation of copyright.

886 ● **High granularity of access control** - individual elements of a collection may
887 have distinct access constraints or licenses. For example, individual patients
888 within a study may have provided different consent. MetaRecords allow each
889 element within a collection to possess, and publish, its own access policy,
890 access protocol, license, and/or usage-constraints, thus providing fine-
891 grained control of the access/use of individual elements within a repository.

892
893

894 **FAIR Facet(s) Addressed by the Triple Descriptors and FAIR Projectors:**

- 895 ● **Findable** - Triple Descriptors, in isolation or when aggregated into FAIR
896 Profiles, provide one or more semantic interpretations of data elements. By
897 indexing these descriptors, it would become possible to search over datasets
898 for those that contain data-types of interest. Moreover, FAIR Projectors, as a
899 result of the TPF URI structure, create a unique URL for every data-point
900 within a record. This has striking consequences with respect to scholarly
901 communication. For example, it becomes possible to unambiguously refer-to,
902 and therefore “discuss” and/or annotate, individual spreadsheet cells from any
903 data repository.
- 904 ● **Accessible** - Using the TPF design patterns, all data retrieval is
905 accomplished in exactly the same way - via HTTP GET. The response
906 includes machine-readable instructions that guide further exploration of the
907 data without the need to define an API. FAIR Projectors also give the data
908 owner high granularity access control; rather than publishing their entire
909 dataset, they can select to publish only certain components of that dataset,
910 and/or can put different access controls on different data elements, for
911 example, down to the level of an individual spreadsheet cell.
- 912 ● **Interoperable** - FAIR Projectors provide a standardized way to export any
913 type of underlying data in a machine-readable structure, using widely used,
914 public shared vocabularies. Data linkages that were initially implicit in the
915 datastore, identifiers for example, become explicit when converted into URIs,
916 resulting in qualified linkages between formerly opaque data deposits.
917 Similarly, data that resides within computationally opaque structures or
918 formats can also be exposed, and published in a FAIR manner if there is an
919 algorithm capable of extracting it and exposing it via the TPF interface.
- 920 ● **Reusable** - All data points now possess unique identifiers, which allows them
921 to be explicitly connected to their citation and license information (i.e. the
922 MetaRecord). In this way, every data point, even when encountered in
923 isolation, provides a path to trace-back to its reusability metadata.

924

925

Other features of FAIR Projection

926

927

928

929

930

931

932

933

934

935

936

937

938

939

940

941

942

943

944

945

946

947

948

949

950

951

952

Incentives and Barriers to Implementation

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

- **Native formats are preserved** - As in many research domains, bioinformatics has created a large number of data/file formats. Many of these, especially those that hold “big data”, are specially formatted flat-files that focus on size-efficient representation of data, at the expense of general machine-accessibility. The analytical tooling that exists in this domain is capable of consuming these various formats. While the FAIR Data community has never advocated for wholesale Interoperable representations of these kinds of data - which would be inefficient, wasteful, and lacking in utility - the FAIR Projector provides a middle-ground. Projection allows software to query the core content of a file in a repository prior to downloading it; for example, to determine if it contains data about an entity or identifier of interest. FAIR Projectors, therefore, enable efficient discovery of data of interest, without requiring wasteful transformation of all data content into a FAIR format.
- **Semantic conversion of existing Triplestores** - It is customary to re-cast the semantic types of entities within triplestores using customized SPARQL BIND or CONSTRUCT clauses. FAIR Projectors provide a standardized, SPARQL-free, and discoverable way to accomplish the same task. This further harmonizes data, and simplifies interoperability.
- **Standardized interface to (some) Web APIs** - Many Web APIs in the biomedical domain have a single input parameter, generally representing an identifier for some biochemical entity. FAIR Projectors can easily replace these myriad Web APIs with a common TPF interface, thus dramatically enhancing discoverability, machine-readability, and interoperability between these currently widely disparate services.

Looking forward, there is every indication that FAIRness will soon be a requirement of funding agencies and/or journals. As such, infrastructures such as the one described in this exemplar will almost certainly become a natural part of scholarly data publishing in the future. Though the FAIR infrastructure proposed here may appear difficult to achieve, we argue that a large portion of these behaviours - for example, the first two layers of the Accessor - can be accomplished using simple fill-in-the-blank templates. Such templating tools are, in fact, already being created by several of the co-authors, and will be tested on the biomedical data publishing community in the near future to ensure they are clear and usable by this key target-audience.

Projection, however, is clearly a complex undertaking, and one that is unlikely to be accomplished by non-informaticians on their own. Transformation from unstructured or semi-structured formats into interoperable formats cannot be fully automated, and we do not claim to have fully solved the interoperability bottleneck. We do, however, claim to have created an infrastructure that improves on the *status quo* in two ways: First, we propose to

968 replace the wasteful, one-off, "reuseless" data transformation activities currently undertaken
969 on a daily basis throughout the biomedical community (and beyond), with a common,
970 reusable, and machine-readable approach, by suggesting that all data transformations
971 should be described in RML and transformed data exposed using TPF. Second, the solution
972 we propose may, in many cases, partially automate the data transformation process itself.
973 RML can be used, in combination with generic software such as RML Processor
974 (<http://github.com/RMLio>) to actuate a data transformation over many common file formats
975 such as CSV or XML. As such, by focusing on building RML models, *in lieu* of reuseless
976 data transformation scripts, data publishers achieve both the desired data transformation, as
977 well as a machine-readable interface that provides that transformed data to all other users.
978 This may be incentivized even more by creating repositories of RML models that can be
979 reused by those needing to do data transformations. Though the infrastructure for capturing
980 these user-driven transformation events and formalizing them into FAIR Projectors does not
981 yet exist, it does not appear on its surface to be a complex problem. Thus, we expect that
982 such infrastructure should appear soon after FAIRness becomes a scholarly publishing
983 requirement, and early prototypes of these infrastructures are being built by our co-authors.

984

985 Several communities of data providers are already planning to use this, or related FAIR
986 implementations, to assist their communities to find, access, and reuse their valuable data
987 holdings. For example, the Biobanking and Rare disease communities will be given end-
988 user tools that utilize/generate such FAIR infrastructures to: guide discovery by researchers;
989 help both biobankers and researchers to re-code their data to standard ontologies building
990 on the SORTA system (Pang et al., 2015); assist to extend the MOLGENIS/BiobankConnect
991 system (Pang et al., 2016); add FAIR interfaces to the BBMRI (Biobanking and BioMolecular
992 resources Research Infrastructure) and RD-connect national and European biobank data
993 and sample catalogues. There are also a core group of FAIR infrastructure authors who are
994 creating large-scale indexing and discovery systems that will facilitate the automated
995 identification and retrieval of relevant information, from any repository, in response to end-
996 user queries, portending a day when currently unused - "lost" - data deposits once again
997 provide return-on-investment through their discovery and reuse.

998

999 Conclusions

1000

1001 There is a growing movement of governing bodies and funding organizations towards a
1002 requirement for open data publishing, following the FAIR Principles. It is, therefore, useful to
1003 have an exemplar "reference implementation" that demonstrates the kinds of behaviours that
1004 are expected from FAIR resources.

1005

1006 Of the four FAIR Principles, Interoperability is arguably the most difficult FAIR facet to
1007 achieve, and has been the topic of decades of informatics research. Several new standards
1008 and frameworks have appeared in recent months that addressed various aspects of the
1009 Interoperability problem. Here, we apply these in a novel combination, and show that the
1010 result is capable of providing interoperability between formerly incompatible data formats

1011 published anywhere on the Web. In addition, we note that the other three aspects of FAIR -
1012 Findability, Accessibility, and Reusability - are easily addressed by the resulting
1013 infrastructure. The outcome, therefore, provides machine-discoverable access to richly
1014 described data resources in any format, in any repository, with the possibility of
1015 interoperability of the contained data down to the level of an individual “cell”. No new
1016 standards or APIs were required; rather, we rely on REST behaviour, with all entities being
1017 resources with a resolvable identifier that allow hypermedia-driven “drill-down” from the level
1018 of a repository descriptor, all the way to an individual data point in the record.

1019
1020 Such an interoperability layer may be created and published by anyone, for any data source,
1021 without necessitating an interaction with the data owner. Moreover, the majority of the
1022 interoperability layer we describe may be achieved through dynamically generated files from
1023 software, or even (for the Accessor portion) through static, manually-edited files deposited in
1024 any public repository. As such, knowledge of how to build or deploy Web infrastructure is not
1025 required to achieve a large portion of these FAIR behaviours.

1026
1027 The trade-off between power and simplicity was considered acceptable, as a means to
1028 hopefully encourage wide adoption. The modularity of the solution was also important
1029 because, in a manner akin to crowdsourcing, we anticipate that the implementation will
1030 spread through the community on a needs-driven basis, with the most critical resource
1031 components being targeted early - the result of individual researchers requiring interoperable
1032 access to datasets/subsets of interest to them. The interoperability design patterns
1033 presented here provide a structured way for these individuals to contribute and share their
1034 individual effort - effort they would have invested anyway - in a collaborative manner, piece-
1035 by-piece building a much larger interoperable and FAIR data infrastructure to benefit the
1036 global community.

1037

1038 Acknowledgements

1039
1040 In January 2014 the Lorentz Center hosted the ‘Jointly Designing a Data FAIRport’
1041 workshop. This workshop was organized by Barend Mons in collaboration with and co-
1042 sponsored by the Lorentz center, The Dutch Techcentre for Life Sciences/ ELIXIR-NL and
1043 the Netherlands eScience Center. The workshop led to the formalization of FAIR principles
1044 and subsequently to the formation of a FAIR Skunkworks team and a FAIR Data engineering
1045 team. We thank Barend Mons for critical discussions leading up to this article. We would
1046 also like to thank the UniProt RDF and SPARQL team at the Swiss-Prot group of the SIB
1047 Swiss Institute of Bioinformatics for their advice and assistance. We would like to
1048 acknowledge the advice and feedback from the leaders and participants of BioHackathon
1049 2016, hosted by the Integrated Database Project (Ministry of Education, Culture, Sports
1050 Science and Technology, Japan), the National Bioscience Database Center (NBDC -
1051 Japan), and the Database Center for Life Sciences (DBCLS - Japan).
1052

1053

1054 **References**

- 1055 Bechhofer S., Buchan I., De Roure D., Missier P., Ainsworth J., Bhagat J., Couch P.,
1056 Cruickshank D., Delderfield M., Dunlop I., Gamble M., Michaelides D., Owen S.,
1057 Newman D., Sufi S., Goble C. 2013. Why linked data is not enough for scientists. *Future*
1058 *generations computer systems: FGCS* 29:599–611.
- 1059 Berners-Lee T. 2006. Linked Data. *Available at*
1060 *<https://www.w3.org/DesignIssues/LinkedData.html>* (accessed September 27, 2016).
- 1061 Ciccarese P., Soiland-Reyes S., Belhajjame K., Gray AJ., Goble C., Clark T. 2013. PAV
1062 ontology: provenance, authoring and versioning. *Journal of Biomedical Semantics* 4:37.
1063 DOI: 10.1186/2041-1480-4-37.
- 1064 Cook CE., Bergman MT., Finn RD., Cochrane G., Birney E., Apweiler R. 2016. The
1065 European Bioinformatics Institute in 2016: Data growth and integration. *Nucleic acids*
1066 *research* 44:D20–6.
- 1067 Court of Justice of the European Union, PRESS RELEASE No 92/16, September 8, 2016.
1068 *Available at [http://curia.europa.eu/jcms/upload/docs/application/pdf/2016-](http://curia.europa.eu/jcms/upload/docs/application/pdf/2016-09/cp160092en.pdf)*
1069 *09/cp160092en.pdf* (Accessed December 21, 2016).
- 1070 Covitz PA., Hartel F., Schaefer C., De Coronado S., Fragoso G., Sahni H., Gustafson S.,
1071 Buetow KH. 2003. caCORE: a common infrastructure for cancer informatics.
1072 *Bioinformatics* 19:2404–2412.
- 1073 Crosswell LC., Thornton JM. 2012. ELIXIR: a distributed infrastructure for European
1074 biological data. *Trends in biotechnology* 30:241–242.
- 1075 Das S., Sundara S., Cyganiak R. 27 September, 2012. *R2RML: RDB to RDF Mapping*
1076 *Language*. W3C Recommendation.
- 1077 Fallside DC, Walmsley P (eds). 8 October, 2004. XML Schema part 0 - Primer Second

- 1078 Edition. Available at <https://www.w3.org/TR/xmlschema-0/> (accessed December 20,
1079 2016)
- 1080 De Giovanni R., Copp C., Döring M., Hobern D. 2010. *TAPIR - TDWG Access Protocol for*
1081 *Information Retrieval*. TSWG Standards .
- 1082 Dimou A., Vander Sande M., Colpaert P., Verborgh R., Mannens E., van de Walle R. 2014.
1083 RML: A Generic Language for Integrated RDF Mappings of Heterogeneous Data. In:
1084 *Proceedings of the 7th Workshop on Linked Data on the Web*. Vol. 1184, CEUR
1085 Workshop Proceedings.
- 1086 Dumontier M., Gray AJG., Marshall MS., Alexiev V., Ansell P., Bader G., Baran J., Bolleman
1087 JT., Callahan A., Cruz-Toledo J., Gaudet P., Gombocz EA., Gonzalez-Beltran AN.,
1088 Groth P., Haendel M., Ito M., Jupp S., Juty N., Katayama T., Kobayashi N.,
1089 Krishnaswami K., Laibe C., Le Novère N., Lin S., Malone J., Miller M., Mungall CJ.,
1090 Rietveld L., Wimalaratne SM., Yamaguchi A. 2016. The health care and life sciences
1091 community profile for dataset descriptions. *PeerJ* 4:e2331.
- 1092 Fielding RT., Taylor RN. 2002. Principled design of the modern Web architecture. *ACM*
1093 *Transactions on Internet Technology* 2:115–150.
- 1094 Gessler DDG., Schiltz GS., May GD., Avraham S., Town CD., Grant D., Nelson RT. 2009.
1095 SSWAP: A Simple Semantic Web Architecture and Protocol for semantic web services.
1096 *BMC Bioinformatics* 10:309.
- 1097 González AR., Callahan A., Cruz-Toledo J., Garcia A., Egaña Aranguren M., Dumontier M.,
1098 Wilkinson MD. 2014. Automatically exposing OpenLifeData via SADI semantic Web
1099 Services. *Journal of biomedical semantics* 5:46.
- 1100 Gray AJG., Baran J., Marshall MS., Dumontier M. 2015. Dataset descriptions: HCLS
1101 community profile. Interest group note, W3C (May 2015) [http://www.w3.org/TR/hcls-](http://www.w3.org/TR/hcls-dataset)
1102 [dataset](http://www.w3.org/TR/hcls-dataset).
- 1103 Heery R., Patel M. 2000. Application profiles: mixing and matching metadata schemas.

- 1104 Ariadne.
- 1105 Ison J., Kalas M., Jonassen I., Bolser D., Uludag M., McWilliam H., Malone J., Lopez R.,
1106 Pettifer S., Rice P. 2013. EDAM: an ontology of bioinformatics operations, types of data
1107 and identifiers, topics and formats. *Bioinformatics* 29:1325–1332.
- 1108 Kuhn T., Tobias K., Christine C., Michael K., Núria Q-R., Ruben V., George G., Ngomo A-
1109 CN., Raffaele V., Michel D. 2016. Decentralized provenance-aware publishing with
1110 nanopublications. *PeerJ Computer Science* 2:e78.
- 1111 Lanthaler M., Gütl C. Hydra: A Vocabulary for Hypermedia-Driven Web APIs. In:
1112 *Proceedings of the 6th Workshop on Linked Data on the Web (LDOW2013)*.
- 1113 Maali F., Erickson J., Archer P. 2014. *Data Catalog Vocabulary (DCAT)*. W3C
1114 Recommendation .
- 1115 Martin D., Paolucci M., McIlraith S., Burstein M., McDermott D., McGuinness D., Parsia B.,
1116 Payne T., Sabou M., Solanki M., Srinivasan N., Sycara K. 2005. Bringing Semantics to
1117 Web Services: The OWL-S Approach. In: 26–42. DOI: 10.1007/978-3-540-30581-1_4.
- 1118 Martin D., Paolucci M., Wagner M. 2007. Towards Semantic Annotations of Web Services:
1119 OWL-S from the SAWSDL Perspective. In: *OWL-S Experiences and Future
1120 Developments, Proceedings*.
- 1121 Miles A., Bechhofer S. 18 August, 2009. *SKOS Simple Knowledge Organization System
1122 Reference*. W3C Recommendation .
- 1123 van Ommen G-JB., Törnwall O., Bréchet C., Dagher G., Galli J., Hveem K., Landegren U.,
1124 Luchinat C., Metspalu A., Nilsson C., Solesvik OV., Perola M., Litton J-E., Zatloukal K.
1125 2015. BBMRI-ERIC as a resource for pharmaceutical and life science industries: the
1126 development of biobank-based Expert Centres. *European journal of human genetics:
1127 EJHG* 23:893–900.
- 1128 Pang C., Sollie A., Sijtsma A., Hendriksen D., Charbon B., de Haan M., de Boer T., Kelpin
1129 F., Jetten J., van der Velde JK., Smidt N., Sijmons R., Hillege H., Swertz MA. 2015.

- 1130 SORTA: a system for ontology-based re-coding and technical annotation of biomedical
1131 phenotype data. *Database: the journal of biological databases and curation* 2015. DOI:
1132 10.1093/database/bav089.
- 1133 Pang C., van Enckevort D., de Haan M., Kelpin F., Jetten J., Hendriksen D., de Boer T.,
1134 Charbon B., Winder E., van der Velde KJ., Doiron D., Fortier I., Hillege H., Swertz MA.
1135 2016. MOLGENIS/connect: a system for semi-automatic integration of heterogeneous
1136 phenotype data with applications in biobanks. *Bioinformatics* 32:2176–2183.
- 1137 Roche DG., Kruuk LEB., Lanfear R., Binning SA. 2015. Public Data Archiving in Ecology and
1138 Evolution: How Well Are We Doing? *PLoS biology* 13:e1002295.
- 1139 Rodriguez Iglesias, Alejandro, Marconi, Marco, Sesma, Ane, & Wilkinson, Mark. (2016).
1140 RDF Representation of RNA Metabolism Evolution data - version 3 (diagrammed in
1141 <https://zenodo.org/deposit/47641/>) [Data set]. Zenodo.
1142 <http://doi.org/10.5281/zenodo.161719>. DOI: 10.5281/zenodo.161719 (accessed 12
1143 December, 2016)
- 1144 SIB Swiss Institute of Bioinformatics Members. 2016. The SIB Swiss Institute of
1145 Bioinformatics' resources: focus on curated databases. *Nucleic acids research* 44:D27–
1146 37.
- 1147 Speicher S., Arwe J., Malhotra A. 2015. *Linked data platform 1.0*. W3C Recommendation.
- 1148 Starr J., Castro E., Crosas M., Dumontier M., Downs RR., Duerr R., Haak LL., Haendel M.,
1149 Herman I., Hodson S., Hourclé J., Kratz JE., Lin J., Nielsen LH., Nurnberger A., Proell
1150 S., Rauber A., Sacchi S., Smith A., Taylor M., Clark T. 2015. Achieving human and
1151 machine accessibility of cited data in scholarly publications. *PeerJ. Computer science* 1.
1152 DOI: 10.7717/peerj-cs.1.
- 1153 Stein LD., Knoppers BM., Campbell P., Getz G., Korb J. 2015. Data analysis: Create a
1154 cloud commons. *Nature* 523:149–151.
- 1155 Stevens RD., Robinson AJ., Goble CA. 2003. myGrid: personalised bioinformatics on the

- 1156 information grid. *Bioinformatics (Oxford, England)* 19 Suppl 1:i302--4.
- 1157
- 1158 Thompson R., Johnston L., Taruscio D., Monaco L., Bérout C., Gut IG., Hansson MG., 't
1159 Hoen P-BA., Patrinos GP., Dawkins H., Ensini M., Zatloukal K., Koubi D., Heslop E.,
1160 Paschall JE., Posada M., Robinson PN., Bushby K., Lochmüller H. 2014. RD-Connect:
1161 an integrated platform connecting databases, registries, biobanks and clinical
1162 bioinformatics for rare disease research. *Journal of general internal medicine* 29 Suppl
1163 3:S780–7.
- 1164 Tim Berners-lee T., Chen Y., Chilton L., Connolly D., Dhanaraj R., Hollenbach J., Lerer A.,
1165 Sheets D. 2006. Tabulator: Exploring and analyzing linked data on the semantic web.
1166 In: *Proceedings of the 3rd International Semantic Web User Interaction Workshop*.
- 1167 Verborgh R., Ruben V., Sande MV., Olaf H., Van Herwegen J., De Vocht L., De Meester B.,
1168 Gerald H., Pieter C. 2016. Triple Pattern Fragments: A low-cost knowledge graph
1169 interface for the Web. *Web Semantics: Science, Services and Agents on the World*
1170 *Wide Web* 37-38:184–206.
- 1171 Verborgh R., Dumontier M. 2016. A Web API ecosystem through feature-based reuse.
- 1172 W3C. 2014. Data Catalog Vocabulary (DCAT). Available at
1173 <http://www.w3.org/TR/vocab-dcat> (accessed December 10, 2016)
- 1174 Wilkinson MD., Dumontier M., Aalbersberg IJJ., Appleton G., Axton M., Baak A., Blomberg
1175 N., Boiten J-W., da Silva Santos LB., Bourne PE., Bouwman J., Brookes AJ., Clark T.,
1176 Crosas M., Dillo I., Dumon O., Edmunds S., Evelo CT., Finkers R., Gonzalez-Beltran A.,
1177 Gray AJG., Groth P., Goble C., Grethe JS., Heringa J., 't Hoen PAC., Hooft R., Kuhn T.,
1178 Kok R., Kok J., Lusher SJ., Martone ME., Mons A., Packer AL., Persson B., Rocca-
1179 Serra P., Roos M., van Schaik R., Sansone S-A., Schultes E., Sengstag T., Slater T.,
1180 Strawn G., Swertz MA., Thompson M., van der Lei J., van Mulligen E., Velterop J.,

- 1181 Waagmeester A., Wittenburg P., Wolstencroft K., Zhao J., Mons B. 2016. The FAIR
1182 Guiding Principles for scientific data management and stewardship. *Scientific data*
1183 3:160018.
- 1184 Wilkinson MD., Senger M., Kawas E., Bruskiwich R., Gouzy J., Noirot C., Bardou P., Ng A.,
1185 Haase D., Saiz E de A., Wang D., Gibbons F., Gordon PMK., Sensen CW., Carrasco
1186 JMR., Fernández JM., Shen L., Links M., Ng M., Opushneva N., Neerincx PBT.,
1187 Leunissen JAM., Ernst R., Twigger S., Usadel B., Good B., Wong Y., Stein L., Crosby
1188 W., Karlsson J., Royo R., Párraga I., Ramírez S., Gelpi JL., Trelles O., Pisano DG.,
1189 Jimenez N., Kerhornou A., Rosset R., Zamacola L., Tarraga J., Huerta-Cepas J.,
1190 Carazo JM., Dopazo J., Guigo R., Navarro A., Orozco M., Valencia A., Claros MG.,
1191 Pérez AJ., Aldana J., Rojano MM., Fernandez-Santa Cruz R., Navas I., Schiltz G.,
1192 Farmer A., Gessler D., Schoof H., Groscurth A. 2008. Interoperability with Moby 1.0--it's
1193 better than sharing your toothbrush! *Briefings in bioinformatics* 9:220–31. DOI:
1194 10.1093/bib/bbn003.
- 1195 Wilkinson MD., Vandervalk B., McCarthy L. 2011. The Semantic Automated Discovery and
1196 Integration (SADI) Web service Design-Pattern, API and Reference Implementation.
1197 *Journal of biomedical semantics* 2:8. DOI: 10.1186/2041-1480-2-8.
- 1198 Withers D., Kawas E., McCarthy L., Vandervalk B., Wilkinson M. 2010. Semantically-Guided
1199 Workflow Construction in Taverna: The SADI and BioMoby Plug-Ins. In: *Leveraging*
1200 *Applications of Formal Methods, Verification, and Validation 4th International*
1201 *Symposium on Leveraging Applications, Lecture Notes in Computer Science*. 301–312.
1202
1203

1204