



HAL
open science

A practical approach to constructing a knowledge graph for soil ecological research

Nicolas Le Guillarme, Wilfried Thuiller

► To cite this version:

Nicolas Le Guillarme, Wilfried Thuiller. A practical approach to constructing a knowledge graph for soil ecological research. *European Journal of Soil Biology*, 2023, 117, pp.103497. 10.1016/j.ejsobi.2023.103497 . hal-04182458

HAL Id: hal-04182458

<https://hal.science/hal-04182458>

Submitted on 18 Aug 2023

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

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



Distributed under a Creative Commons Attribution 4.0 International License

1

1 A Practical Approach to Constructing a Knowledge Graph for 2 Soil Ecological Research

3 **List of authors:** Nicolas Le Guillarme, Wilfried Thuiller

4 **Affiliations:**

5 Univ. Grenoble Alpes, Univ. Savoie Mont Blanc, CNRS, LECA, F-38000 Grenoble, France.

6 **Corresponding author:** Nicolas Le Guillarme; nicolas.leguillarme@univ-grenoble-alpes.fr

7 **Abstract**

8 With the rapid accumulation of biodiversity data, data integration has emerged as a hot topic
9 in soil ecology. Data integration has indeed the potential to advance our knowledge of global
10 patterns in soil biodiversity by facilitating large-scale meta-analytical studies of soil
11 ecosystems. However, ecologists are still poorly equipped when it comes to integrating
12 disparate datasets. In recent years, knowledge graphs have emerged as a powerful tool for
13 integrating large amounts of distributed heterogeneous data while making these data more
14 easily interpretable by humans and computers. This paper presents a practical approach to
15 constructing a biodiversity knowledge graph from heterogeneous and distributed
16 (semi-)structured data sources. To illustrate our approach, we integrate several datasets on
17 the trophic ecology of soil organisms into a trophic knowledge graph and show how both
18 explicit and implicit information can be retrieved from the graph to support multi-trophic
19 studies.

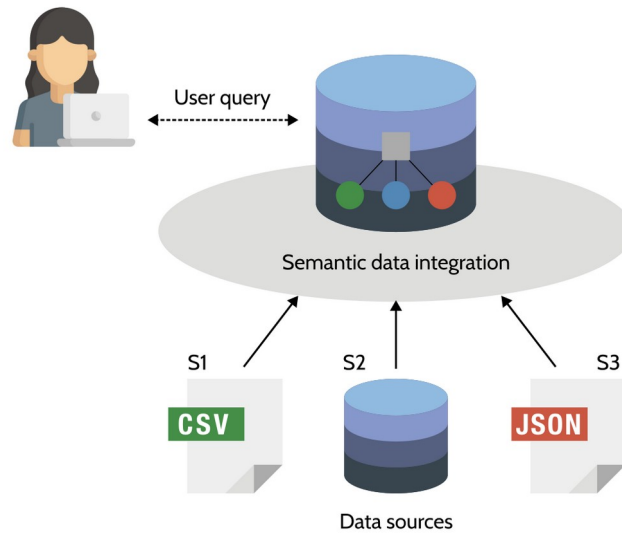
20

21 **Keywords:** data integration, knowledge graph, ontology, reasoning, soil ecology

22 Introduction

23 In recent years, a number of initiatives aiming at collecting new soil biodiversity data or
24 assembling existing datasets have emerged, resulting in a rapid accumulation of data in soil
25 ecology [1]. Because of the enormous phylogenetic, taxonomic and functional diversity of
26 soil organisms, datasets are often collected by individual scientists or small project teams
27 from different communities or disciplines to answer precise research questions. These
28 datasets are typically small, with a limited spatial/temporal/taxonomic coverage, and are
29 formatted according to the project needs, with little or no concern for data standardization
30 [2]. This causes datasets to be heterogeneous in semantics (differences in terminologies,
31 meaning or interpretation of data in different disciplines or research contexts), schema
32 (differences in data structures and formats) and syntax (differences in models or languages).
33 In addition, datasets are widely distributed: they reside on diverse locations, e.g. files or
34 databases on the local network or published on the web, and are accessible using different
35 interfaces, e.g., files downloads, database queries or web APIs.

36 Integrating these ‘long-tail data’ dispersed across different datasets could help address
37 research questions at larger scales [3]. Data integration is of growing interest in the
38 ecological domain, with much effort directed towards the creation of standard terminologies
39 for describing, sharing and facilitating the aggregation of biodiversity data, e.g. organismal
40 trait data [4, 5, 6, 7], into large open databases. Recent initiatives in trait-based ecology have
41 targeted specific taxonomic groups, e.g. ants [8], spiders [9], soil invertebrates [10, 11], fungi
42 [12, 13], plants [14]. Although these databases have made aggregated data more readily
43 accessible to scientists, they are not yet interoperable. The difficulty of integrating data
44 distributed across heterogeneous sources remains. As a result, integrative analyses of soil
45 communities that span several taxonomic groups and integrate multitrophic interactions are
46 scarce — see [15] for an example — although essential to improve our understanding of the
47 links between soil biodiversity and ecosystem functioning [16].

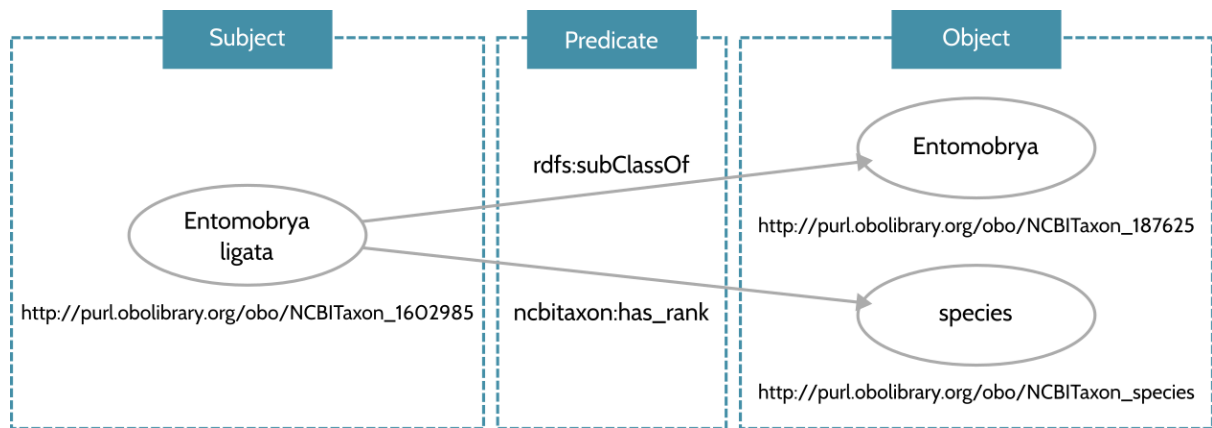


48

49 *Figure 1. Semantic data integration provides the user with a uniform access to a set of*
 50 *autonomous and possibly heterogeneous data sources in a particular application domain.*

51

52 Here, we address the problem of semantic data integration in the biodiversity science
 53 domain. Data integration is defined in [17] as the process of combining data residing at
 54 different sources, and providing the user with a unified view of these data. (Figure 1). As a
 55 result, the user has the ability to seamlessly manipulate data from multiple sources,
 56 regardless of the original format or location of the data. Semantic data integration aims at
 57 combining heterogeneous data in a way that preserves the original 'meaning' of the data in
 58 their particular semantic context. In practice, this often consists in establishing semantic
 59 correspondences (also called *mappings*) between the vocabularies of the different data
 60 sources and a common reference *ontology*. The result of this process is called a knowledge
 61 graph.



62

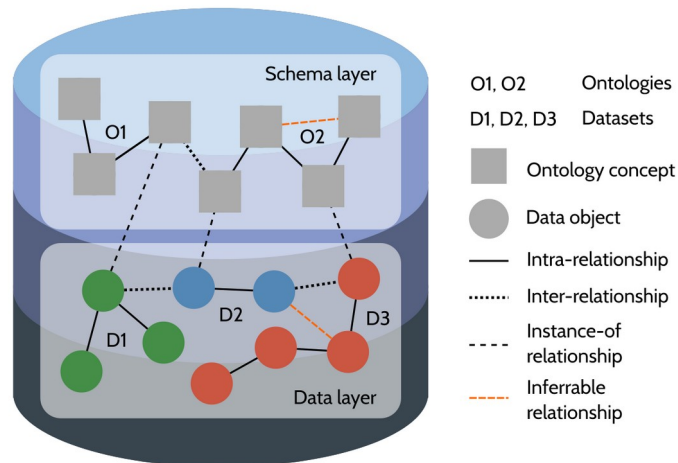
63 *Figure 2. The RDF data format represents factual statements about entities (here, the taxon*
 64 *Entomobrya ligata) as triples that consist of a subject, predicate and object. RDF triples form*
 65 *a labeled directed graph, which is why RDF databases are also called RDF graphs.*

66

67 A knowledge graph (KG) is a graph-structured knowledge base that stores factual
 68 information in the form of relationships between real-world entities (like people, places,
 69 ‘things’) [18]. Under the Resource Description Framework (RDF), the standard data model of
 70 the Semantic Web, a KG is a set of (subject, predicate, object) triples. A RDF triple is a
 71 factual statement about an entity (the subject), connected to another entity or a data value
 72 (the object) by a relationship (the predicate). A set of RDF triples forms a labeled directed
 73 graph, called a RDF graph (Figure 2). But not every RDF graph is a KG. The triples in a KG
 74 can be separated into two distinct, yet connected, layers (Figure 3). The schema layer is the
 75 conceptual model of the KG and is described by an ontology or a collection of ontologies. An
 76 ontology is a formal shared conceptualization of a domain of interest [19]. It defines a
 77 common agreed upon terminology in terms of concepts (also called *classes*, i.e. the types of
 78 things that exist in the domain) and the relationships holding among them. An ontology is
 79 specified using a logic-based ontology language — most often the Web Ontology Language
 80 (OWL), built upon RDF — that allows both humans and computers to understand the
 81 semantics (‘meaning’) of the data. The data layer holds the concrete, factual data. These
 82 data are *instances* of the general concepts (classes) defined in the ontology. For example, if
 83 we were to define a class ‘Article author’ in a hypothetical ontology to describe the concept

84 of 'people who write scientific papers', then 'Nicolas Le Guillarme' and 'Wilfried Thuiller'
 85 would be two instances of this class. In the context of semantic data integration, the data
 86 layer of a KG is populated with instance data from multiple sources. The ontology is used to
 87 link these disparate datasets at the schema-level, acting as a mediator for reconciling the
 88 structural and semantic heterogeneities between data sources.

89



90

91 *Figure 3. A knowledge graph is a graph database that embeds both the data and its*
 92 *semantics in two interconnected layers. The schema layer is an ontology or a collection of*
 93 *ontologies that integrate datasets at the schema-level and allow logical inference of implicit*
 94 *knowledge using specialized softwares called reasoners. The data layer is a collection of*
 95 *data from various sources.*

96

97 KGs have a number of advantages over other types of databases, such as relational ones.
 98 Their graph structure allows for efficient querying, intuitive visualization, and analysis using
 99 graph algorithms or relational machine learning [18]. Using an ontology as a schema layer,
 100 KGs embed a formal semantics with the data which can be used by computers to interpret
 101 and reason about the data, thus potentially allowing to infer new facts (e.g. the inferrable
 102 relationships in Figure 2). KGs make it easy to integrate new types of data by altering the
 103 ontology or adding a new ontology to the schema layer. When following the Linked Open

104 Data principles [20], domain-specific KGs can be easily interconnected into larger (possibly
105 cross-domain) KGs.

106 KGs have recently become prevalent as a framework for semantic data integration in many
107 different domains of science and industry [21]. It was R. Page, in his seminal 2016 paper,
108 who first suggested the use of KGs in the biodiversity field [22]. Since then, only a few
109 examples of biodiversity KGs have been published. Ozymandias [23] is a KG for the
110 Australian fauna that integrates data from several sources, including the Atlas of Living
111 Australia, the Australian Faunal Directory, the Biodiversity Heritage Library and the
112 Biodiversity Literature Repository. OpenBiodiv [24] integrates information extracted from the
113 biodiversity literature into a graph database using the OpenBiodiv-O ontology and an RDF
114 version of the Global Biodiversity Information Facility (GBIF) taxonomic backbone. TAXREF-
115 LD [25] is a KG representation of the French national taxonomical register for fauna, flora
116 and fungus that interlinks information about taxonomy, species interactions, development
117 stages, biogeography, conservation statuses, etc. In a recent talk at TDWG 2021, Michel et
118 al. [26] called for more biodiversity data producers to start publishing KGs. However, for
119 now, building a KG from multiple data sources is a complex and time-consuming task that
120 demands high Semantic Web expertise, and we are not aware of an existing tool specifically
121 designed to help ecologists transform their data sets into interoperable KGs — with the
122 notable exception of the iKNOW project [27], which is very similar in spirit to our work, but
123 whose current status is unknown to us.

124 In this paper, we present inteGraph, a framework and toolbox that facilitates the process of
125 building a KG from heterogeneous and distributed (semi-)structured data sources in the
126 biodiversity domain. With inteGraph, users can create automatic and reproducible semantic
127 data integration pipelines simply through the provision of configuration files. This declarative
128 approach requires no (or little) code from the user and minimizes the amount of manual
129 effort and Semantic Web expertise required to turn datasets into interoperable KGs.

130 To illustrate our approach, we will show how inteGraph can be used to integrate data on the
131 trophic ecology of soil organisms from multiple sources into a KG that can support

132 multitrophic studies. Multitrophic studies, spanning multiple trophic levels and/or taxonomic
133 groups, are essential to identify general patterns in community ecology [28], understand how
134 diversity is related to ecosystem stability and ecosystem functioning [29], and provide the
135 necessary guidance with biodiversity loss and environmental problems [30]. Multitrophic
136 approaches should acknowledge the complexity of ecosystems while remaining practical.
137 Large trait databases have the potential to address this trade-off between feasibility and
138 completeness. By supporting the assignment of species (or higher taxonomic ranks) to
139 trophic and/or functional groups, they reduce the dimensionality of ecological communities
140 without biasing studies toward a single trophic level or taxonomic group [31, 15]. Yet, some
141 challenges still remain. Although we have trait databases available for some groups of soil
142 organisms, our trait knowledge is limited for most of them. In addition, existing databases
143 tend to function as data silos whose lack of interoperability can discourage researchers to
144 include more trophic levels and/or taxonomic groups in their studies.

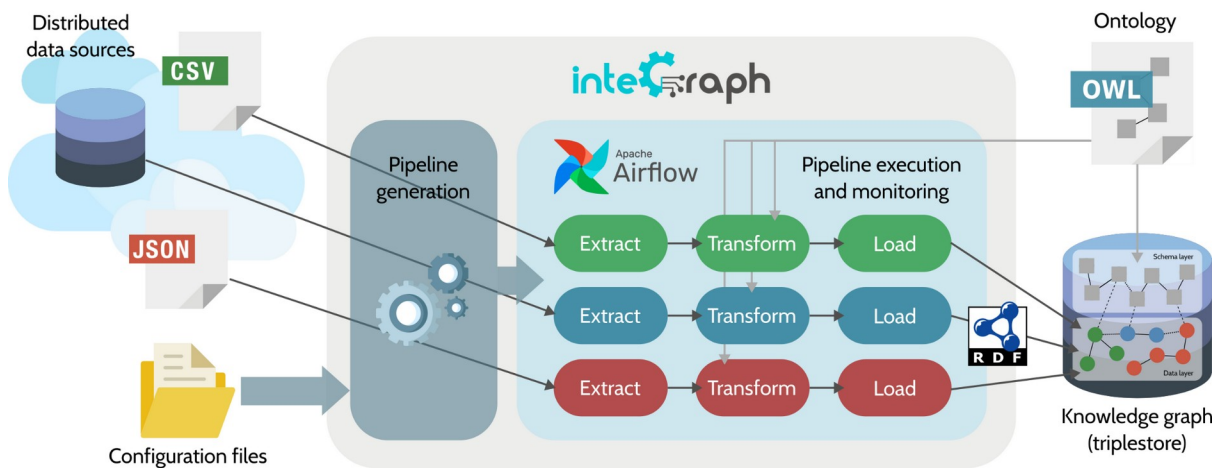
145 The ability to create a KG integrating trophic information from a number of trait databases
146 covering different soil taxonomic groups across several trophic levels could greatly facilitate
147 multitrophic studies in soil ecology research. Such a trophic KG would provide a unified
148 access to multigroup, multitrophic, and multisource information. The integration of several
149 trophic datasets (e.g. a first one containing information on carabid diets and a second one
150 focusing on the feeding habits of springtails) into a KG allows the use of a single query to
151 retrieve all organisms with a particular diet, regardless of the format, location or taxonomic
152 coverage of the original data source. KGs also offer the ability to reason about the integrated
153 data to derive additional knowledge. Reasoning about trophic interactions and dietary data
154 opens the way for automatic classification of soil organisms into trophic groups, which can
155 facilitate the reconstruction of consistent soil food webs from multisource data. This will be
156 illustrated with examples in the Results section.

157 Material and Methods

158 Overview of the approach

159 Figure 4 is a high-level representation of our approach to constructing a KG from
 160 heterogeneous and distributed (semi-)structured data sources. At the heart of our framework
 161 is inteGraph¹, an open-source toolkit for ontology-based data integration in the biodiversity
 162 domain that allows generating data integration pipelines dynamically from configuration files
 163 and scheduling and monitoring the execution of these pipelines.

164



165

166 *Figure 4. A high-level representation of the proposed declarative approach for constructing a*
 167 *knowledge graph from distributed (semi-)structured data sources.*

168 Data sources

169 Data sources can be (and often are) distributed on several machines, on a local network
 170 and/or on the web. Data must be accessible in a (semi-)structured form, for instance as
 171 tabular (e.g. tables in relational databases or in CSV files) or hierarchical data (e.g. data in
 172 XML or JSON format). At present, inteGraph does not include information extraction
 173 components that would allow the integration of unstructured textual data from the literature.

9 1 Available at <https://github.com/nleguillarme/inteGraph>

174 In our running example, we will use inteGraph to build a trophic KG from the following three
175 data sources:

176 ● The Fun^{Fun} database [13] collates fungal functional trait data, including information
177 about trophic guilds, from a variety of sources, for thousands of species across the
178 fungal tree of life. Data are provided in a tabular format (CSV file) and can be
179 downloaded from Zenodo (<https://zenodo.org/record/1216257>).

180 ● BETSI [11] is an open database gathering data on morphological traits and
181 ecological preferences for 7 taxonomic groups of soil invertebrates (Aranae,
182 Carabidae, Chilopoda, Collembola, Diplopoda, Isopoda and Diplotesticulata) from
183 about 2000 literature references. BETSI is accessible on demand via a web portal
184 (<https://portail.betsi.cnrs.fr>) that provides the user with an interface to write queries
185 and download subsets of the database in a tabular format (CSV file). In the following,
186 we will integrate a dataset containing Carabidae diet data.

187 ● The Global Biotic Interactions (GloBI) provides open access to species interaction
188 data (e.g. predator-prey, pollinator-plant, pathogen-host, parasite-host) aggregated
189 from existing open datasets [32]. As of April 2023, GloBI contains over 17M
190 interaction records obtained from 342 datasets, covering 823,033 taxa. GloBI
191 provides several ways to access its data, including a web portal
192 (<https://www.globalbioticinteractions.org/>), a downloadable snapshot of the entire
193 database in a tabular format (CSV file), and a web API. In our example, we will use
194 the web API to download data about the trophic interactions of Collembola.

195 Information from these three data sources will populate the data layer of our trophic KG once
196 it has been transformed into a common representation.

197 Target ontology

198 InteGraph adopts a top-down approach to KG creation. In this type of approach, the schema
199 layer of the KG is populated with a predefined ontology. The semantic data integration
200 process then consists of populating the data layer of the KG with data extracted from the

201 different data sources, and creating semantic links between the schema of the sources and
202 the (global) schema of the KG using mapping rules.

203 To reconcile schematic and semantic heterogeneities between our trophic data sources, the
204 schema layer of our example KG will be populated with two ontologies: the NCBITaxon
205 ontology and the Soil Food Web Ontology (SFWO) [7]. NCBITaxon is a formal translation of
206 the NCBI Taxonomy database into an ontology, in which each taxon is treated as a class
207 whose instances would be individual organisms, e.g. 'Nicolas Le Guillarme' instance_of
208 NCBITaxon_9606 (*Homo sapiens*). To our knowledge, the NCBI Taxonomy database is the
209 only taxonomic nomenclature available as an OWL ontology. SFWO is an ontology for
210 representing knowledge on the trophic ecology of soil organisms across taxonomic groups
211 and trophic levels. SFWO captures the semantics of trophic concepts such as trophic
212 interactions, feeding processes, diets or trophic groups. SFWO also includes machine-
213 interpretable definitions for most of these concepts, that allow for inference of implicit
214 knowledge using automated reasoning, e.g. deducing a consumer's diet(s) from the trophic
215 interaction(s) in which it participates.

216 Triplestore

217 A triplestore is a database management system, i.e., a software used for storing and
218 querying a database, specifically designed to support the storage and the efficient querying
219 of RDF triples. A triplestore is needed to store both the schema and data layers of a KG.
220 Information stored in the triplestore can be retrieved using SPARQL queries. A multitude of
221 triplestore implementations are available (see [33] for a survey), which offer different
222 capabilities and performance in terms of data storage and indexing, query processing,
223 reasoning, etc.

224 InteGraph assumes the existence of a running triplestore instance. It is not tied to a specific
225 implementation and provides connectors to several triplestore solutions. The user is
226 expected to provide connection information as part of the pipeline configurations. As a top-

227 down approach, inteGraph assumes that the target ontology has been loaded in the
228 triplestore before the data integration process starts.

229 To store our example knowledge graph, we will use GraphDB Free², a RDF triplestore
230 solution that can manage billions of explicit statements on a desktop hardware, while
231 providing optimized query evaluation and OWL reasoning.

232 Configuration files

233 InteGraph implements a declarative approach to building KGs, which means that it provides
234 control over the creation and execution of semantic data integration pipelines using
235 configuration files. InteGraph requires the user to provide two types of configuration files: a
236 single graph configuration file (Figure 5a), and a set of source configuration files, one for
237 each data source (Figure 5b).

238 The graph configuration file contains global information, including the name of the KG (that
239 acts as a prefix to create an identifier for each graph generated from a data source), the
240 name of the directory containing the source configuration files, the triplestore connection
241 information, and the declaration of the target ontologies.

242 The source configuration file is where the user specifies the information needed by
243 inteGraph to instantiate the Extract and Transform components of the data integration
244 pipeline for a given source. This includes:

- 245 ● the internal identifier of the data source;
- 246 ● data access information, which determines the type of data source – file-like or HTTP
247 – and the appropriate data extraction component to be added to the pipeline;
- 248 ● information about the format of the input data, e.g. tab or comma-separated values;
- 249 ● the path to an (optional) data cleansing script;
- 250 ● for each entity (e.g. taxon, trait) in the input data, the name of the columns containing
251 information about the entity (label and/or identifier), and a sequence of semantic

- 252 annotation components whose role is to map the entity to its equivalent in the target
 253 ontology;
- 254 ● the path to the spreadsheet containing the schema mapping rules.

```
[core]
base_iri=http://leca.osug.fr/example

[sources]
dir=sources

[load]
id=graphdb
conn_type=http
host=0.0.0.0
port=7200
user=integraph
password=iNtEgR@pH
repository=example

[ontologies]
sfwo=https://purl.org/sfwo/sfwo.owl
```

```
[core]
source_id=funfun

[extract]
[extract.file]
file_path=https://github.com/traitecoevo/fungaltraits/
releases/latest/download/funtothefun.csv

[transform]
format=csv
delimiter=";"
chunksize=1000

[transform.cleanse]
script="clean.py"

[transform.annotate]
[transform.annotate.taxon]
label=speciesMatched
id=ifungorum_number
source=ifungorum
target=["ncbi"]

[transform.annotate.guild]
label=guild_fg
target=["sfwo", "mapping.yml"]

[transform.triplify]
mapping=mapping.xlsx
```

(a) Graph configuration

(b) Source configuration for the Fun^{Fun} database

- 255 *Figure 5. InteGraph implements a declarative approach to KG construction, giving the user*
 256 *control over the creation of data integration pipelines through simple configuration files.*
- 257

Extracted data

(a)

obj_id	speciesMatched	ifungorum_number	trait_name	value	
D1:	Anderson_2008_2	Aspergillus niger	284309	guild_fg	Plant Pathogen-Wood Saprotroph
	Anderson_2008_5	Botrytis allii	237131	guild_fg	Plant Pathogen

taxon_name	trait_name	attribute_trait	
D2:	Abax (Abax) ovalis (Duftschmid, 1812)	Diet	Zoophage
	Abax (Abax) ovalis (Duftschmid, 1812)	Diet	Necrophagous
	Abax (Abax) parallelepipedus (Piller & Mitterpacher, 1783)	Diet	Phytophage

source_taxon_name	source_taxon_external_id	interaction_type	target_taxon_name	target_taxon_external_id	
D3:	Entomobrya ligata	EOL:1022839	eats	rotting wood	ENVO:00002040
	Entomobrya ligata	EOL:1022839	eats	Fungi	GBIF:5
	Tetradontophora bielansensis	GBIF:4538324	eats	Calocera viscosa	INAT_TAXON:63405

Cleansed data

(b)

obj_id	speciesMatched	ifungorum_number	guild_fg	
D1:	Anderson_2008_2	Aspergillus niger	284309	Plant Pathogen
	Anderson_2008_2	Aspergillus niger	284309	Wood Saprotroph
	Anderson_2008_5	Botrytis allii	237131	Plant Pathogen

Annotated data

(c)

obj_id	speciesMatched	ifungorum_number	guild_fg	taxon_iri	guild_iri	
D1:	Anderson_2008_2	Aspergillus niger	284309	Plant Pathogen	NCBITaxon:5061	SFWO:0000159
	Anderson_2008_2	Aspergillus niger	284309	Wood Saprotroph	NCBITaxon:5061	SFWO:0000070
	Anderson_2008_5	Botrytis allii	237131	Plant Pathogen	NCBITaxon:279185	SFWO:0000159

taxon_name	trait_name	attribute_trait	taxon_iri	diet_iri	
D2:	Abax (Abax) ovalis (Duftschmid, 1812)	Diet	Zoophage	NCBITaxon:106379	ECOCORE:00000088
	Abax (Abax) ovalis (Duftschmid, 1812)	Diet	Necrophagous	NCBITaxon:106379	ECOCORE:00000090
	Abax (Abax) parallelepipedus (Piller & Mitterpacher, 1783)	Diet	Phytophage	NCBITaxon:102642	ECOCORE:00000019

source_taxon_name	interaction_type	target_taxon_name	source_taxon_iri	interaction_iri	target_taxon_iri	
D3:	Entomobrya ligata	eats	rotting wood	NCBITaxon:1602985	RO:0002470	SFWO:0000149
	Entomobrya ligata	eats	Fungi	NCBITaxon:1602985	RO:0002470	NCBITaxon:4751
	Tetradontophora bielansensis	eats	Calocera viscosa	NCBITaxon:48717	RO:0002470	NCBITaxon:63146

 semantic annotations

RDF data (N-quads)

(d)

subject	predicate	object	graph	
D1:	_:consumer_0	rdf:type	obo:CARO_0001010	https://leca.osug.fr/funfun
	_:consumer_0	rdf:type	sfo:SFWO_0000159	https://leca.osug.fr/funfun
	_:consumer_0	rdf:type	obo:NCBITaxon_5061	https://leca.osug.fr/funfun

subject	predicate	object	graph	
D2:	_:consumer_0	rdf:type	obo:CARO_0001010	https://leca.osug.fr/betsi_carabidae
	_:consumer_0	rdf:type	obo:ECOCORE_00000088	https://leca.osug.fr/betsi_carabidae
	_:consumer_0	rdf:type	obo:NCBITaxon_106379	https://leca.osug.fr/betsi_carabidae

subject	predicate	object	graph	
D3:	_:interaction_0	rdf:type	obo:ECOCORE_00000088	https://leca.osug.fr/globi_collembola
	_:interaction_0	obo:RO_0000057	_:consumer_0	https://leca.osug.fr/globi_collembola
	_:interaction_0	obo:RO_0000057	_:resource_0	https://leca.osug.fr/globi_collembola
	_:consumer_0	rdf:type	obo:CARO_0001010	https://leca.osug.fr/globi_collembola

The predicate *rdf:type* connects the data layer to the schema layer by indicating that the subject is an instance of a class in the target ontology.

259 *Figure 6. An illustration of the application of data transformation to our example datasets*
260 *(D1: Fun^{Fun}, D2: Carabidae diet data from BETSI, D3: Collembola trophic interaction data*
261 *from GloBI). Figure 6d shows the set of RDF triples (in N-quads format) generated from the*
262 *first row of each data table.*

263 Anatomy of inteGraph pipelines

264 InteGraph pipelines are structured according to the Extract-Transform-Load (ETL) paradigm.
265 An ETL pipeline collects data from an input source (extract), cleans and maps the data from
266 a source schema — the schema of the original data source — to a target schema
267 (transform), and saves the transformed data into a triplestore (load). In a typical ETL
268 process, a copy of the extracted data is stored in a data staging area and all transformations
269 are applied to the staged data. In inteGraph, an ETL pipeline is dynamically created at
270 runtime for each data source from the configuration files provided by the user. This ETL
271 pipeline extracts and stages the raw data from the data source, transforms the staged data
272 into a RDF graph, and loads the RDF graph into the data layer of the triplestore.

273 Data extraction

274 This first step of the ETL data integration process involves collecting data from the data
275 source. InteGraph implements a number of components to connect to different types of data
276 sources. At the moment, inteGraph supports the following types of data sources:

- 277 ● File-like data sources: inteGraph can download files from remote or local file-like
278 sources by specifying the local path or the URL of the source in the configuration.
279 Archive files, including compressed archives, are supported, and unpacked before
280 staging.
- 281 ● HTTP data sources: inteGraph can extract data from remote databases exposed
282 through a web-based API by sending HTTP GET requests to the API endpoint. In
283 that case, the user is expected to provide the URL of the endpoint and the query
284 string. Paginated results are supported using the limit and offset parameters.

285 These two components alone are sufficient to access most ecological datasets. We plan to
286 add more connectors in the future, including connectors to SQL databases, RDF databases,
287 etc. The extracted data are staged on the local file system.

288 Figure 6a shows the data extracted from our three example data sources. The three
289 datasets use different data structures and terminologies to organize and describe taxonomic
290 and trophic information.

- 291 ● The Fun^{Fun} database uses the Index Fungorum taxonomic nomenclature. Each line of
292 the data table contains a single trait information for a given taxon. The name of the
293 trait is given in the trait_name column and its value(s) is given in the value column.
294 The terminology used to describe the guild of each taxon is inherited from the
295 FunGuild database.
- 296 ● BETSI does not encode taxonomic information using identifiers from a reference
297 taxonomy. Taxa are designated only by their scientific name. Similar to Fun^{Fun}, each
298 line of the data table contains a single trait information for a given taxon. The diet
299 terminology is taken from the T-SITA thesaurus [4].
- 300 ● Each line in GloBI's data table contains information about a single interaction. The
301 interaction is directed, so each line contains information about the source and target
302 taxa (names and identifiers in an external reference taxonomy, e.g. ITIS, NCBI,
303 GBIF...) and the interaction name. GloBi maintains a mapping between different
304 taxonomic nomenclature internally, but each taxon in the data table is linked to a
305 single identifier. The target of the trophic interaction can also be a non-taxonomic
306 entity, e.g. rotten wood.

307 Data transformation

308 The second step of the ETL data integration process involves transforming the staged data
309 into a RDF graph, i.e. a set of RDF triples. In inteGraph, data transformation consists of two
310 successive operations: data cleansing and schema mapping.

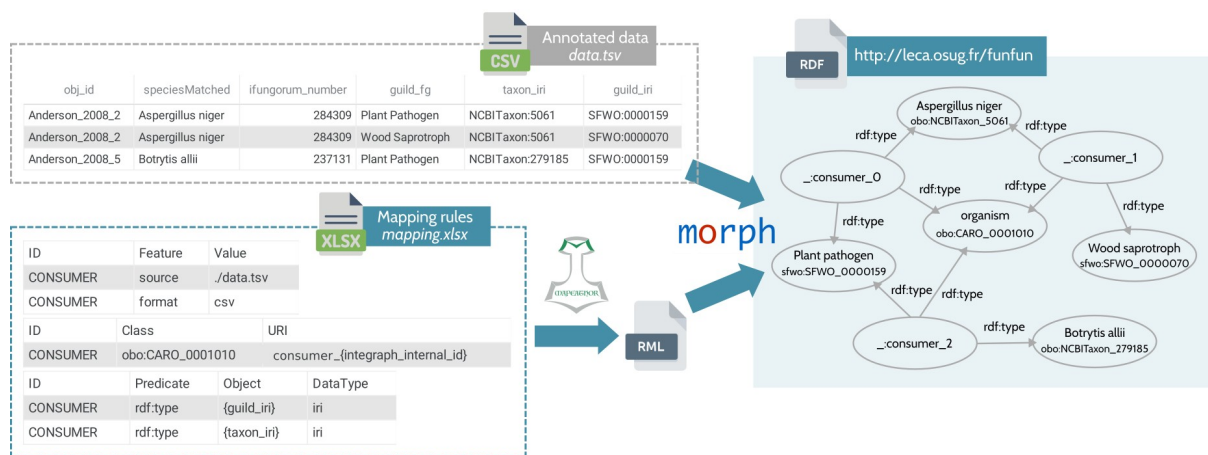
311 Under the term data cleansing, we include all the dataset-specific data processing
312 operations that aim at formatting the extracted data so that they are ready for further
313 processing by the schema mapping component. This includes operations such as removing
314 or filling missing values, removing duplicates, dropping irrelevant data, splitting strings (i.e.
315 splitting a string representing a set of values, e.g. 'bacterivore-detritivore', into a set of
316 strings, each string representing a single value, e.g. 'bacterivore', 'detritivore'), joining two or
317 more data tables, etc. Figure 6b shows an example of applying cleansing operations to the
318 Fun^{Fun} dataset so that each line of the data table contains a single guild value. As possible
319 cleansing operations are very diverse and highly dependent on the structure of the input
320 data, they cannot be specified in the source configuration file. Instead, the user should
321 provide a Python or R script that implements the data cleansing operations as a separate
322 file. This script should respect some input/output constraints so that it can be ingested by
323 inteGraph at runtime and incorporated into the ETL pipeline.

324 After data cleansing is complete and cleansed data are staged, the pipeline moves on to
325 schema mapping which involves converting data from the schema of the original data source
326 to the schema of the knowledge graph, i.e. the target ontology. Schema mapping in
327 inteGraph consists of two successive tasks: semantic annotation and RDF graph generation.
328 Semantic annotation is the process of linking the input data with the concepts in the target
329 ontology that best capture the semantics of the data (Figure 6c). InteGraph provides several
330 components for semantic annotation of biodiversity data. The first component maps
331 taxonomic entities (identified by their name and/or identifier in a source taxonomic
332 nomenclature) to a target taxonomy — in our running example, the NCBITaxon ontology.
333 Taxonomic mapping uses GNparser [34] to parse scientific names and nomen [35] to match
334 taxon names and identifiers to their equivalent in the target taxonomy. The second
335 annotation component allows any entity (e.g. trait name, trait value, interaction type) to be
336 linked to concepts in a target ontology using exact string matching. For instance, to link the
337 term 'Plant pathogen' found in the Fun^{Fun} database to the corresponding class in the Soil
338 Food Web Ontology, the component will retrieve all the classes whose label (or the label of

339 one of its synonyms) matches exactly the lookup term. If a single eligible class is found, the
340 term in the input data is annotated with this concept identifier, here SFWO:0000159 which is
341 the identifier of the class 'plant pathogen' in SFWO. A third annotation component allows the
342 user to provide a YAML file containing a dictionary with term:concept pairs. Semantic
343 annotation components can be chained together to handle mismatched terms (see the
344 source configuration file example in Figure 5b). Figure 6c shows the result of applying
345 semantic annotation on our example datasets.

346 Once the relevant data have been linked to the corresponding concepts in the target
347 ontology, the final step of schema mapping is the conversion of the annotated dataset into
348 an RDF graph (Figure 6d). InteGraph uses RDF Mapping Language (RML) [36] rules to
349 transform tabular data into RDF triples. RML is a declarative language for expressing rules
350 that map data in heterogeneous structures to the RDF data model. These rules describe the
351 desired graph structure, that is how the data and schema layers of the graph should be
352 connected to each other. The schema mapping rules should be provided by the user as part
353 of the data source configuration. However, writing RML mapping documents is beyond the
354 reach of most non-expert users. To face this issue, inteGraph enables to specify mapping
355 rules in spreadsheets that are automatically translated into RML documents using
356 Mapeathor [37] (Figure 7). This provides a more user-friendly manner to declare mapping
357 rules in a language-independent way. Finally, inteGraph applies Morph-KGC [38], a modern
358 RML processing engine with a focus on speed and scalability, to execute the RML mapping
359 rules and generate the RDF graph. Morph-KGC uses the RML rules to determine how the
360 annotated data should be transformed into RDF triples. The RML rules are applied to each
361 row in the annotated data to generate the RDF representation of the information in the row.
362 In case of missing data (e.g. a taxonomic entity that could not be mapped to the target
363 taxonomic), the RDF triples that use the missing data are not materialized. RML rules
364 processing results in a RDF graph which is the sum of the sets of RDF triples generated for
365 each row. Figure 4d shows an extract of the RDF graphs obtained by applying RML rules to
366 our example datasets. RDF graphs are staged in N-quads format, a serialization format for

367 RDF data that associates each triple with an optional context value at the fourth position.
 368 This context value takes the form of a graph label, indicating which RDF graph the triple
 369 belongs to. This graph label is used to keep track of the data provenance (original data
 370 source) after the different RDF graphs are merged into a single KG in the triplestore.
 371



372
 373 *Figure 7. InteGraph allows the user to specify schema mapping rules in a spreadsheet,*
 374 *which are automatically converted into RML rules using Mapeathor. The RML rules are*
 375 *executed using Morph-KGC to generate the RDF graph.*

376 Data loading

377 The third and final step of the ETL data integration process is to save the RDF graph
 378 generated during the data transformation stage in an external RDF database, i.e. a
 379 triplestore. The triplestore must be set up beforehand, either on the same machine running
 380 inteGraph or on a dedicated server. Different triplestore implementations may use different
 381 techniques to ingest RDF data. InteGraph provides connectors to the following triplestore
 382 solutions: RDFox, GraphDB, and Virtuoso. InteGraph also supports loading RDF data to a
 383 triplestore using SPARQL Update operations. In our example, the trophic KG is stored on an
 384 instance of GraphDB Free. The GraphDB connector provided by inteGraph simply loads
 385 RDF data to the triplestore using an HTTP POST request.

386 At the moment, inteGraph supports full load only. This means that the transformed data are
 387 loaded in full at each run of the ETL pipeline. Therefore, the KG is reconstructed from

388 scratch every time the data integration pipelines are executed. A useful alternative would be
389 incremental data load, i.e. updating the KG at regular intervals by loading only the data that
390 has changed (new or updated data) since the last execution. This requires additional tools to
391 compare the data from the data source with the existing data present in the KG. Incremental
392 data load has a number of advantages over full load, including faster processing and
393 preservation of data history.

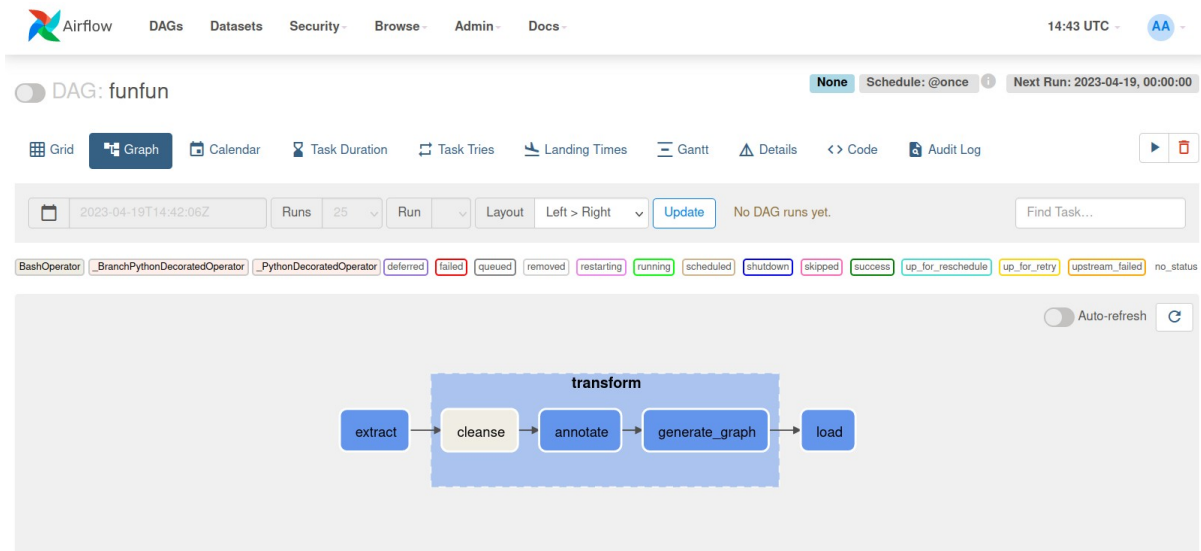
394 Pipeline creation, scheduling and monitoring

395 InteGraph uses Apache Airflow³ to schedule and monitor the execution of the data
396 integration pipelines. Airflow provides a flexible programmatic (i.e. code-based) approach to
397 easily build scheduled data processing pipelines as directed acyclic graphs (DAGs) of tasks.
398 DAGs are a natural representation for ETL pipelines as each step in the ETL process is
399 executed after the previous one has been completed (there is no circular dependency
400 between ETL tasks). Tasks in Airflow should be atomic – they either succeed and produce
401 some proper result or fail in a manner that does not affect the state of the system – and
402 idempotent, i.e. rerunning a task without changing the inputs should not change the overall
403 output.

404 At runtime, inteGraph parses the graph and source configuration files and creates one
405 Airflow pipeline per data source, decomposing the full ETL pipeline into a DAG of atomic and
406 idempotent tasks. A schedule interval can be assigned to each pipeline, which determines
407 when and how often the pipeline is run. Alternatively, the user can manually trigger the
408 execution of a pipeline in Airflow's graphical user interface. This interface also allows to
409 visualize the pipelines generated by inteGraph and monitor their execution (Figure 8). Airflow
410 can handle failures in ETL operations by retrying them a couple of times. If the error persists,
411 the user can easily explore the logs of the failing task, identify the cause of the failure, and
412 rerun the failing task (together with any subsequent tasks that depend on that task). Airflow
413 also has the ability to run multiple tasks in parallel. Therefore, pipelines can be executed

22 3 <https://airflow.apache.org/>

414 efficiently, taking advantage of any parallelism inherent in the tasks dependency structure.
 415 For example, inteGraph can split the input data into chunks that are transformed in parallel
 416 and merged before loading, reducing pipeline execution time. In addition, ETL pipelines can
 417 run in parallel as each data source is independent of the others.
 418



419
 420 *Figure 8. A high-level view of the ETL pipeline for the Fun^{Fun} database in the Airflow user*
 421 *interface.*

422 Results

423 Knowledge retrieval

424 At the end of the semantic data integration process, the target ontology and the transformed
 425 data are both saved in a single triplestore. The triplestore is responsible for storing the KG
 426 and executing SPARQL queries to retrieve information from it. SPARQL is a query language
 427 for retrieving and manipulating data stored in RDF format. SPARQL is based on matching
 428 graph patterns against the RDF graph. The basic graph pattern is the triple pattern, which is
 429 like a RDF triple where any part of the triple can be replaced by a variable. A graph pattern is
 430 a combination of such triple patterns. When executing a SPARQL query against a KG in a
 431 triplestore, the triplestore searches for the set(s) of triples that exactly match the graph

432 patterns defined in the query, regardless of the provenance (i.e. the original source) of the
 433 triples, unless explicitly requested. This means that the set of RDF triples returned in
 434 response to a SPARQL query may contain facts originating from different data sources. The
 435 KG provides the user with a unified view of the original data sources through querying, and
 436 enables combining multisource information as part of a query response. Figure 9 shows a
 437 SPARQL query searching the KG for phytophagous species. The same query returns both
 438 springtail and carabid species, whose dietary information originates from the GloBi and
 439 BETSI databases respectively. This simple example shows how a single query against the
 440 KG can retrieve information from multiple sources simultaneously, thus greatly facilitating
 441 integrative studies across taxonomic groups and/or trophic levels.

442

```



PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX obo: <http://purl.obolibrary.org/obo/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX ncbitaxon: <http://purl.obolibrary.org/obo/ncbitaxon#>
SELECT DISTINCT ?species_name
WHERE {
  ?individual      rdf:type
  ?individual      rdf:type
  ?species_taxon  ncbitaxon:has_rank
  ?species_taxon  rdfs:label
}
LIMIT 10
    
```

The query starts by declaring prefixes, which allow the IRIs to be abbreviated in the rest of the query.

The information returned by the query.

The identifier (IRI) of the concept *herbivore (syn. phytophage)* in SFWO.

443

species_name	
Sminthurinus aureus	
Sminthurus viridis	
Sphaeridia pumilis	
Sminthurinus niger	
Dicyrtoma fusca	
Bourletiella hortensis	
Orchesella villosa	
Pseudoophonus griseus	
Harpalus griseus	
Amara eurynota	

444

445 *Figure 9. Example of a SPARQL query returning the species names of phytophagous taxa. ?*
 446 *x denotes a variable called x. The LIMIT keyword is used to limit the number of results to the*

447 *first 10 entries. The query returns information about both phytophagous springtails (from the*
448 *GloBi database) and phytophagous carabid beetles (from the BETSI database).*

449 Making implicit knowledge explicit

450 The semantic data integration process builds a KG by linking heterogeneous datasets at the
451 schema level using an ontology. During the process, the data layer of the KG is populated
452 with the factual information stated in the different datasets and transformed into knowledge
453 through semantic annotation and transformation into a RDF graph. Based on these explicit
454 facts, additional knowledge that is not explicitly present in the data can be derived using
455 reasoning. This ability is a direct consequence of OWL (the standard language for specifying
456 ontologies) being based on a subset of first-order logic. Therefore, automated reasoners can
457 be employed to evaluate the logical implications of the knowledge encoded in the ontology
458 on the explicitly stated data.

459 There are two principle strategies for logical inference: forward chaining and backward
460 chaining [38]. Forward chaining, also known as materialization, derives all the facts that can
461 be logically deduced from the existing facts and a set of logical rules, and stores these
462 inferred facts in the triplestore for later querying. Precomputing all inferred facts enables
463 efficient query answering, but it can also be very expensive both in time (the materialization
464 process needs to consider all possible inferences) and memory (the process can derive a
465 large number of facts). In addition, materialization must be redone each time the data is
466 updated.

467 Backward chaining (query rewriting) starts from a query and applies the logical rules only as
468 far as they are needed to answer the query. With backward chaining, reasoning is done at
469 runtime and no time- and space-consuming precomputation is needed. Furthermore, no
470 recomputation has to be done when the data is updated. However, a major drawback of
471 backward chaining is that reasoning must be done for each new query, which can be
472 computationally expensive and slow.

473 In our running example, the target ontologies (the NCBITaxon ontology and the Soil Food
 474 Web Ontology) are loaded in a triplestore supporting reasoning based on forward chaining.
 475 The Soil Food Web Ontology provides a set of logical rules that map consumer-resource
 476 interactions to diets (e.g. an animal feeding on detritus is a detritivore), as well as rules for
 477 classifying soil-associated consumers into hybrid taxonomic-trophic groups (e.g.
 478 detritivorous springtails are members of the group *Collembola.detrivores*). These rules
 479 make it possible to automate the process of assigning taxa to trophic groups using logical
 480 inference, thus reducing the burden of manual trophic group assignment.

481 Figure 10 illustrates how information about trophic group membership is made explicit in our
 482 trophic KG using materialization. After transformed data are loaded in the triplestore at the
 483 end of the data integration pipeline, inference rules are applied repeatedly to the asserted
 484 (explicit) statements until no further inferred (implicit) statements are produced. Given (1) the
 485 explicit information about *Entomobrya ligata* feeding on rotting wood (see first line of data
 486 table D3 in Figure 6a), (2) the hierarchy of taxonomic concepts provided by the NCBITaxon
 487 ontology, and (3) the logical rules provided by the Soil Food Web Ontology, the triplestore
 488 reasoner is able to materialize the following logical implications:

- 489 ● *E. ligata* is a species of springtails (*Collembola*) ;
- 490 ● *E. ligata* is a detritivore, as it feeds on rotten wood, which is a type of detritus ;
- 491 ● *E. ligata* belongs to the group of detritivorous springtails (*Collembola.detrivores*) as
 492 a logical consequence of the two previous assertions.

493

```

PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX obo: <http://purl.obolibrary.org/obo/>
PREFIX rdfs: <http://www.w3.org/2000/01/rdf-schema#>
PREFIX sfwo: <http://purl.org/sfwo/>
SELECT DISTINCT ?tg_iri ?tg_name
WHERE {
  ?collembola rdf:type obo:NCBITaxon_1602985.
  ?collembola obo:RO_0002350 ?tg_iri.
  ?tg_iri rdfs:subClassOf sfwo:SFWO_0000127.
  ?tg_iri rdfs:label ?tg_name.
}
  
```

The identifier of *E. ligata*
in the NCBITaxon ontology.

The identifier of the relationship
member of in SFWO.

The identifier of the concept
trophic group in SFWO.

494

tg_iri	tg_name
http://purl.org/sfwo/SFWO_0000302	Collembola.all
http://purl.org/sfwo/SFWO_0000304	Collembola.detritivores

495

496 *Figure 10. Example of a SPARQL query returning the trophic groups to which Entomobrya*
 497 *ligata belongs. A trophic group is defined in the Soil Food Web Ontology as ‘a collection of*
 498 *organisms that feed on the same food sources and have the same consumers’ [7, 31].*
 499 *SFWO provides a logical formalization of the hierarchical classification of soil consumers*
 500 *proposed in [39].*

501 Performance

502 InteGraph relies on a number of external tools with a strong focus on scalability (e.g.
 503 GNparser for scientific name parsing, Morph-KGC for RML rules execution). This, combined
 504 with Airflow’s ability to run independent tasks in parallel, makes inteGraph itself quite
 505 efficient at handling large datasets in a reasonable time. In our experiments, we were able to
 506 convert tabular data with over 440K rows into an RDF graph in about 6 minutes on a laptop
 507 with twelve 2.60GHz Intel Core i7 CPUs and 16GB of RAM. Currently, the main bottlenecks
 508 are taxonomic mapping, which in some cases may require many calls to web APIs, and
 509 logical inference, the performance of which depends on the types of reasoning and the
 510 optimisations implemented by the triplestore (inteGraph provides no reasoning component,
 511 the inference is left entirely to the triplestore).

512 With the ability to chain semantic annotation components, including user-provided dictionary-
 513 like mapping files, inteGraph is able to convert most of the input data into RDF, with however
 514 some entries being dropped because they cannot be linked to concepts in the target
 515 ontology. Most of the time, this happens because the taxonomic entities could not be
 516 mapped to the target taxonomy. This can be due to the source and target taxonomies being
 517 incompatible, the taxon name being ambiguous or deprecated, etc. For example, in the
 518 FunFun database, 41 of the 508 unique taxa could not be mapped to the NCBITaxon
 519 ontology, resulting in 15% of the input data being dropped during the data integration

520 process. In the Carabidae dataset extracted from BETSI, this proportion is only 0.3% (with
521 18 of the 5491 unique taxa for which inteGraph could not find a correspondence in the
522 NCBITaxon ontology).

523 Discussion

524 Multitrophic studies require harmonizing and integrating datasets across a large variety of
525 taxonomic groups and trophic levels. Despite considerable efforts to make more biodiversity
526 data freely available in a (semi-)structured format, the multiple dimensions of data
527 heterogeneity (semantic, structural, syntactic) constitute a major obstacle to the
528 interoperability of data sources [3]. Here, we introduced a practical approach to data
529 integration that aims at making heterogeneous and distributed biodiversity data sources
530 interoperable as part of a single KG. KGs provide a unified representation of disparate data
531 sources and allow for retrieving data across these sources using a single query. By using
532 ontologies as global schemas, they add semantics to the integrated data, making it easier for
533 humans and computers to interpret the data and for reasoners to infer additional facts. As
534 seen from the example discussed in the Results section, the ability to reason about the data
535 in our trophic KG opens avenues for automatic classification of soil organisms, which can
536 facilitate the reconstruction of consistent soil food webs from multisource data. In addition,
537 KGs provide support for a number of applications [41], including both in-KG, e.g. link
538 prediction, error detection, and out-of-KG applications, e.g. relation extraction from text,
539 recommender systems, etc.

540 Despite their many advantages, KG construction is currently out-of-reach for most
541 biodiversity data providers and consumers as they require in-depth expertise in Semantic
542 Web technologies. InteGraph is an attempt to make semantic data integration and KG
543 construction more accessible to the biodiversity science community. Requiring little or no
544 code and minimal knowledge of the Semantic Web, inteGraph facilitates the processes of
545 converting a biodiversity dataset into a KG and of integrating multiple datasets into a single

546 KG. Given a set of distributed data sources and a target ontology, inteGraph allows the user
547 to control the creation and execution of reproducible ontology-based data integration
548 pipelines through a set of simple configuration files. This declarative approach relieves the
549 user of the implementation burden. Instead, the user can focus on the desired structure of
550 the target KG and on the schema mapping rules needed to transform the input data into
551 RDF graphs. InteGraph relies on high-performance third-party tools (gnparser, nomer,
552 Morph-KGC, Airflow), which guarantees a certain ability to scale to large datasets. The
553 viability of our approach has been tested by creating a KG of soil trophic ecology from
554 multiple open trait databases, using the Soil Food Web Ontology and the NCBITaxon
555 ontology as the KG schema. Currently, inteGraph is at the proof-of-concept stage. It still
556 needs some development to make it more robust, scalable and user-friendly. We also plan
557 to add more advanced features in the future, especially regarding data provenance tracking
558 and continuous KG updating.

559 Although it represents a significant advance in the field of ontology-based biodiversity data
560 integration, inteGraph suffers limitations related to current practices in biodiversity data
561 management. First, inteGraph requires the data sources to provide data in a
562 (semi-)structured format through a programmatic interface, e.g. a URL to download the data
563 file or a web API that handles HTTP requests. Still lots of data about soil biodiversity are not
564 accessible this way, e.g. data from the BETSI database must be downloaded manually. We
565 are confident that this situation will become less frequent in the future. Second, as a top-
566 down approach to KG creation, inteGraph requires a predefined ontology to act as the
567 mediating schema to link heterogeneous data sources. Creating an ontology to model
568 knowledge in a domain of interest is a complex process that requires a significant investment
569 of time and effort. Ontology engineering asks for a group of experts to produce a consensual
570 conceptualization of the domain. For instance, in the domain of soil trophic ecology, this
571 means trying to harmonize the use of diet terms that may have different meanings from one
572 taxonomic group to another. However, we believe that the result is worth the effort, as a

573 properly designed ontology can benefit the whole community by facilitating knowledge
574 sharing, dataset standardization and, ultimately, data integration.

575 Finally, although it aims to make semantic data integration more accessible to a non-expert
576 audience, inteGraph still requires a minimum of knowledge of Semantic Web technologies
577 (RDF data, ontologies, SPARQL queries...). Just as the environmental community has
578 begun to embrace new artificial intelligence tools from recent developments in deep machine
579 learning, we encourage the community to take an interest in Semantic Web tools for better
580 biodiversity knowledge management.

581 Continuing our efforts to develop more and more biodiversity ontologies [42, 25, 43, 7] would
582 allow us to envision increasing semantification of the ecology domain in the near future.

583 Combined with tools such as inteGraph, which facilitate the conversion of biodiversity
584 datasets into graph knowledge bases, these semantic resources could support the creation
585 by different communities of numerous domain-specific KGs, which could eventually be
586 interconnected to form a single biodiversity KG covering the entire tree of life and the full
587 diversity of global ecosystems.

588

Box 1. Glossary

Class: in an ontology, a description of a concept in the domain of interest.

A class is a set of individuals that share common characteristics, and the class definition gives the properties that these individuals must fulfill to be members of the class.

For instance, 'bacterivore' is the class of all individual organisms that feed on bacteria.

Extract-Transform-Load: a three-phase data integration process that combines data from multiple sources into a single central repository.

Instance (individual): a real-world realization of a concept defined in an ontology. In ontological terms, an individual is an instance of a class in the ontology.

Knowledge graph: a knowledge base that uses a graph-structured data model to integrate data.

N-quads: a line-based, plain text format for storing and transmitting RDF data. A N-quads statement is a RDF triple extended with an optional context value that takes the form of a graph label, indicating which graph the triple belongs to.

Ontology: the formal and consensual description of a domain of interest as a set of interrelated concepts.

Reasoner: a computer program that uses an ontology to infer logical consequences from a set of asserted facts.

Resource Description Framework (RDF): the standard data model of the Semantic Web. RDF represents any piece of information as subject-predicate-object triples.

RDF Mapping Language: a language for expressing mapping rules from heterogeneous data structures to the RDF data model.

Semantic data integration: the process of combining data from different sources into a single, unified view using ontologies.

Semantic Web: a set of standard technologies - including the Resource Description Framework (RDF), Web Ontology Language (OWL), and SPARQL - that help make computers better able to interpret data and information published on the web.

SPARQL: the standard query language for retrieving and manipulating data stored in RDF format.

Triplestore: a database engine optimized for the storage and retrieval of RDF data.

Web API: an interface consisting of one or more endpoints publicly exposed on the web, that allow a user to programmatically access some specific features or the data of an application, e.g. a database.

Web Ontology Language (OWL): a family of knowledge representation languages and the World Wide Web Consortium's (W3C) standard for authoring ontologies, built on RDF and characterized by formal semantics based on description logics (decidable fragments of first-order logic).

589 Acknowledgements

590 We acknowledge support from the European Union's Horizon Europe under grant
591 agreement N°101060429 (NaturaConnect), the French Agence Nationale de la Recherche
592 through the EcoNet (ANR-18-CE02-0010), GlobNet (ANR-16-CE02-0009) and FishPredict
593 projects and the MIAI@Grenoble Alpes (ANR-19-P3IA-0003) institute.

594 References

- 595 [1] White, H. J., León-Sánchez, L., Burton, V. J., Cameron, E. K., Caruso, T., Cunha, L., ... &
596 Caplat, P. (2020). Methods and approaches to advance soil macroecology. *Global Ecology*
597 *and Biogeography*, 29(10), 1674-1690.
- 598 [2] Poisot, T., Bruneau, A., Gonzalez, A., Gravel, D., & Peres-Neto, P. (2019). Ecological
599 data should not be so hard to find and reuse. *Trends in ecology & evolution*, 34(6), 494-496.
- 600 [3] Vanderbilt, K., & Gries, C. (2021). Integrating long-tail data: How far are we?. *Ecological*
601 *Informatics*, 64(C).
- 602 [4] Pey, B., Laporte, M. A., Nahmani, J., Auclerc, A., Capowiez, Y., Caro, G., ... & Hedde, M.
603 (2014). A thesaurus for soil invertebrate trait-based approaches. *PLoS One*, 9(10), e108985.
- 604 [5] Garnier, E., Stahl, U., Laporte, M. A., Kattge, J., Mougnot, I., Kühn, I., ... & Klotz, S.
605 (2017). Towards a thesaurus of plant characteristics: an ecological contribution. *Journal of*
606 *Ecology*, 105(2), 298-309.
- 607 [6] Schneider, F. D., Fichtmueller, D., Gossner, M. M., Güntsch, A., Jochum, M., König-Ries,
608 B., ... & Simons, N. K. (2019). Towards an ecological trait-data standard. *Methods in Ecology*
609 *and Evolution*, 10(12), 2006-2019.
- 610 [7] Le Guillarme, N., Hedde, M., Potapov, A., Berg, M. P., Briones, M. J. I., Hohberg, K., ... &
611 Thuiller, W. (2023). The Soil Food Web Ontology: aligning trophic groups, processes, and
612 resources to harmonise and automatise soil food web reconstructions. bioRxiv doi:
613 10.1101/2023.02.03.526812

- 614 [8] Parr, C. L., Dunn, R. R., Sanders, N. J., Weiser, M. D., Photakis, M., Bishop, T. R., ... &
615 Gibb, H. (2017). GlobalAnts: a new database on the geography of ant traits (Hymenoptera:
616 Formicidae). *Insect Conservation and Diversity*, 10(1), 5-20.
- 617 [9] Pekár, S., Wolff, J. O., Černecká, L., Birkhofer, K., Mammola, S., Lowe, E. C., ... &
618 Cardoso, P. (2021). The World Spider Trait database: a centralized global open repository
619 for curated data on spider traits. *Database*, 2021.
- 620 [10] Potapov, A., Sandmann, D., & Scheu, S. (2019). Ecotaxonomy: Linking traits, taxa,
621 individuals and samples in a flexible virtual research environment for ecological studies.
622 *Biodiversity Information Science and Standards*, 3, e37166.
- 623 [11] Joimel, S., Nahmani, J., Hedde, M., Auclerc, A., Léa, B., Bonfanti, J., ... & Benjamin, P.
624 (2021, April). A large database on functional traits for soil ecologists: BETSI. In *Global*
625 *Symposium on Soil Biodiversity* (pp. 523-528).
- 626 [12] Soudzilovskaia, N. A., Vaessen, S., Barcelo, M., He, J., Rahimlou, S., Abarenkov, K., ...
627 & Tedersoo, L. (2020). FungalRoot: global online database of plant mycorrhizal associations.
628 *New Phytologist*, 227(3), 955-966.
- 629 [13] Zanne, A. E., Abarenkov, K., Afkhami, M. E., Aguilar-Trigueros, C. A., Bates, S.,
630 Bhatnagar, J. M., ... & Treseder, K. K. (2020). Fungal functional ecology: bringing a trait-
631 based approach to plant-associated fungi. *Biological Reviews*, 95(2), 409-433.
- 632 [14] Kattge, J., Bönisch, G., Díaz, S., Lavorel, S., Prentice, I. C., Leadley, P., ... & Cuntz, M.
633 (2020). TRY plant trait database—enhanced coverage and open access. *Global change*
634 *biology*, 26(1), 119-188.
- 635 [15] Calderón-Sanou, I., Zinger, L., Hedde, M., Martínez-Almoyna, C., Saillard, A., Renaud,
636 J., ... & Thuiller, W. (2022). Energy and physiological tolerance explain multi-trophic soil
637 diversity in temperate mountains. *Diversity and Distributions*, 28(12), 2549-2564.
- 638 [16] Eisenhauer, N., Bender, S. F., Calderón-Sanou, I., de Vries, F. T., Lembrechts, J. J.,
639 Thuiller, W., ... & Potapov, A. (2022). Frontiers in soil ecology—Insights from the World
640 Biodiversity Forum 2022. *Journal of Sustainable Agriculture and Environment*, 1(4), 245-261.

- 641 [17] Lenzerini, M. (2002). Data integration: A theoretical perspective. In *Proceedings of the*
642 *twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*
643 (pp. 233-246).
- 644 [18] Nickel, M., Murphy, K., Tresp, V., & Gabrilovich, E. (2015). A review of relational
645 machine learning for knowledge graphs. *Proceedings of the IEEE*, 104(1), 11-33.
- 646 [19] Madin, J. S., Bowers, S., Schildhauer, M. P., & Jones, M. B. (2008). Advancing
647 ecological research with ontologies. *Trends in ecology & evolution*, 23(3), 159-168.
- 648 [20] Mountantonakis, M., & Tzitzikas, Y. (2019). Large-scale semantic integration of linked
649 data: A survey. *ACM Computing Surveys (CSUR)*, 52(5), 1-40.
- 650 [21] Ryen, V., Soylu, A., & Roman, D. (2022). Building Semantic Knowledge Graphs from
651 (Semi-) Structured Data: A Review. *Future Internet*, 14(5), 129.
- 652 [22] Page, R. D. M. (2016). Towards a biodiversity knowledge graph. *Research Ideas and*
653 *Outcomes*, 2.
- 654 [23] Page, R. D. M. (2019). Ozymandias: a biodiversity knowledge graph. *PeerJ*, 7, e6739.
- 655 [24] Penev, L., Dimitrova, M., Senderov, V., Zhelezov, G., Georgiev, T., Stoev, P., & Simov,
656 K. (2019). OpenBiodiv: a knowledge graph for literature-extracted linked open data in
657 biodiversity science. *Publications*, 7(2), 38.
- 658 [25] Michel, F., Faron, C., Tercerie, S., Gargominy, O. (2017-2022) TAXREF-LD: Knowledge
659 Graph of the French taxonomic registry. <https://doi.org/10.5281/zenodo.5848916>
- 660 [26] Michel, F., Ettorre, A., Faron, C., Kaplan, J., & Gargominy, O. (2021). Biodiversity
661 Knowledge Graphs: Time to move up a gear!. *Biodiversity Information Science and*
662 *Standards*, 5, e73699.
- 663 [27] Babalou, S., Kleinstuber, E., El Haoui, B., Zander, F., Costa, D. S., Kattge, J., &
664 König-Ries, B. (2022). iKNOW-A Knowledge Graph Management Platform for the
665 Biodiversity Domain. *International Semantic Web Conference (ISWC) 2022: Posters,*
666 *Demos, and Industry Tracks*.
- 667 [28] Gaüzere, P., O'Connor, L., Botella, C., Poggiato, G., Münkemüller, T., Pollock, L.J.
668 Brose, U., Maiorano, L., Harfoot, M.H. and Thuiller, W. (2022) The diversity of interactions

- 669 complements functional and phylogenetic facets of biodiversity. *Current Biology*, 32(9),
670 2093-2100
- 671 [29] Thompson, R. M., Brose, U., Dunne, J. A., Hall Jr, R. O., Hladyz, S., Kitching, R. L., ... &
672 Tylianakis, J. M. (2012). Food webs: reconciling the structure and function of biodiversity.
673 *Trends in ecology & evolution*, 27(12), 689-697.
- 674 [30] Seibold, S., Cadotte, M. W., MacIvor, J. S., Thorn, S., & Müller, J. (2018). The necessity
675 of multitrophic approaches in community ecology. *Trends in ecology & evolution*, 33(10),
676 754-764.
- 677 [31] Hedde, M., Blight, O., Briones, M. J., Bonfanti, J., Brauman, A., Brondani, M., ... &
678 Capowiez, Y. (2022). A common framework for developing robust soil fauna classifications.
679 *Geoderma*, 426, 116073.
- 680 [32] Poelen, J. H., Simons, J. D., & Mungall, C. J. (2014). Global biotic interactions: An open
681 infrastructure to share and analyze species-interaction datasets. *Ecological Informatics*, 24,
682 148-159.
- 683 [33] Ali, W., Saleem, M., Yao, B., Hogan, A., & Ngomo, A. C. N. (2020). Storage, Indexing,
684 Query Processing, and Benchmarking in Centralized and Distributed RDF Engines: A
685 Survey. *arXiv preprint arXiv:2009.10331*.
- 686 [34] Mozzherin, D. Y., Myltsev, A. A., & Patterson, D. J. (2017). "gnparser": a powerful parser
687 for scientific names based on Parsing Expression Grammar. *BMC bioinformatics*, 18(1), 1-
688 14.
- 689 [35] Salim, J. A., & Poelen, J.. (2022). globalbioticinteractions/nomer: 0.4.8 (0.4.8). Zenodo.
690 <https://doi.org/10.5281/zenodo.7458675>.
- 691 [36] Dimou, A., Vander Sande, M., Colpaert, P., Verborgh, R., Mannens, E., & Van de Walle,
692 R. (2014). RML: a generic language for integrated RDF mappings of heterogeneous data.
693 *Proceedings of the Workshop on Linked Data on the Web co-located with the 23rd*
694 *International World Wide Web Conference (WWW 2014), Seoul, Korea, April 8, 2014*.

- 695 [37] Iglesias-Molina, A., Pozo-Gilo, L., Dona, D., Ruckhaus, E., Chaves-Fraga, D., & Corcho,
696 O. (2020, January). Mapeathor: Simplifying the specification of declarative rules for
697 knowledge graph construction. In *ISWC (Demos/Industry)*.
- 698 [38] Arenas-Guerrero, J., Chaves-Fraga, D., Toledo, J., Pérez, M. S., & Corcho, O. (2022).
699 Morph-KGC: Scalable knowledge graph materialization with mapping partitions. *Semantic*
700 *Web*.
- 701 [39] Antoniou, G., Batsakis, S., Mutharaju, R., Pan, J. Z., Qi, G., Tachmazidis, I., ... & Zhou,
702 Z. (2018). A survey of large-scale reasoning on the web of data. *The Knowledge*
703 *Engineering Review*, 33.
- 704 [40] Potapov, A. M., Beaulieu, F., Birkhofer, K., Bluhm, S. L., Degtyarev, M. I., Devetter,
705 M., ... & Scheu, S. (2022). Feeding habits and multifunctional classification of soil-associated
706 consumers from protists to vertebrates. *Biological Reviews*, 97(3), 1057-1117.
- 707 [41] Wang, Q., Mao, Z., Wang, B., & Guo, L. (2017). Knowledge graph embedding: A survey
708 of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*,
709 29(12), 2724-2743.
- 710 [42] Walls, R. L., Deck, J., Guralnick, R., Baskauf, S., Beaman, R., Blum, S., ... & Wooley, J.
711 (2014). Semantics in support of biodiversity knowledge discovery: an introduction to the
712 biological collections ontology and related ontologies. *PloS one*, 9(3), e89606.
- 713 [43] Abdelmageed, N., Algergawy, A., Samuel, S., & König-Ries, B. (2021). BiodivOnto:
714 towards a core ontology for biodiversity. In *The Semantic Web: ESWC 2021 Satellite Events:*
715 *Virtual Event, June 6–10, 2021, Revised Selected Papers 18* (pp. 3-8). Springer International
716 Publishing.