Electronic Communications of the EASST
Volume 85 Year 2025

deRSE25 - Selected Contributions of the 5th Conference for Research Software Engineering in Germany

Edited by: René Caspart, Florian Goth, Oliver Karras, Jan Linxweiler, Florian Thiery, Joachim Wuttke

From Code to Community: FAIR4RS Practices in Computational Archaeology

Florian Thiery, Fabian Fricke, Sophie C. Schmidt, Lutz Schubert, Fiona Schenk, Chrisowalandis Deligio, Karsten Tolle, Lasse Mempel-Länger

DOI: 10.14279/eceasst.v85.2706

Electronic Communications of the EASST (https://eceasst.org).
Published by **Berlin Universities Publishing**
(https://www.berlin-universities-publishing.de/)

From Code to Community: FAIR4RS Practices in Computational Archaeology

Florian Thiery[1,*], **Fabian Fricke**[2], **Sophie C. Schmidt**[3], **Lutz Schubert**[4], **Fiona Schenk**[5], **Chrisowalandis Deligio**[6], **Karsten Tolle**[6], **Lasse Mempel-Länger**[7]

[1] Research Squirrel Engineers Network & Leibniz-Zentrum für Archäologie (LEIZA)
[2] Deutsches Archäologisches Institut, Berlin
[3] Freie Universität Berlin
[4] IT Center University of Cologne (ITCC)
[5] Johannes Gutenberg-Universität Mainz
[6] Goethe-Universität Frankfurt am Main
[7] Leibniz-Zentrum für Archäologie, Mainz
[*] Correspondence: Florian Thiery, florian.thiery@littleminions.link

Abstract: This paper presents a curated set of community-developed software tools that enable FAIR and FAIR4RS-compliant practices in Computational Archaeology. Rather than offering a generic catalogue, the contribution follows a case-based structure: each tool cluster is introduced as a practical example of how FAIR principles and the REAL framework (Reproducibility, Executability, Attribution, Literalness) are implemented in real-world research workflows. The featured tools, developed within the NFDI4Objects consortium and the CAA-DE community, include semantic RDF transformation pipelines, lightweight scripting environments using R and Python, AI-powered analysis for image-based data, and infrastructures for community-based research software development. By explicitly linking each tool to FAIR4RS principles, the paper demonstrates how Research Software Engineers (RSEs) and domain scientists collaboratively advance sustainable, open, and reusable practices in the digital humanities. In doing so, it complements broader discussions on FAIR implementation in the humanities and shows how bottom-up, domain-specific RSE practices interface with emerging infrastructure initiatives such as nfdi.software and the NFDI Knowledge Graph. The goal is to provide a transferable blueprint for pragmatic FAIRification across disciplines.

Keywords: RSE, NFDI, Archaeology, FAIR, Open Science

1 Introduction

Research Software Engineering (RSE) has become a critical component of digital research workflows across disciplines. In the context of the humanities, and particularly in Computational Archaeology (also referred to as "Archäoinformatik" in the German-speaking community), the development and deployment of domain-specific research software plays a central role in handling, modelling, and FAIRifying complex archaeological data structures [Hug24, IEM+10]. As in many other fields, there is a growing demand to make not only research data but also the software tools themselves Findable, Accessible, Interoperable, and Reusable (FAIR) [WDA+16].

This applies equally to lightweight prototypes, community scripting environments, and software for infrastructure tasks. At the same time, implementing FAIR in the humanities requires approaches that reflect domain-specific practices, heterogeneous data types, and the strong reliance on community-driven workflows.

In this paper, FAIR and FAIR4RS [BCK+22] serve not just as a conceptual backdrop but as explicit analytical lenses. The contribution demonstrates, through concrete examples, how community-maintained scripting in Computational Archaeology operationalises key aspects of the FAIR principles, and how these same tools expose characteristic challenges in Reproducibility, Executability, Attribution, and Literalness as described in the REAL framework [ST25]. Introducing both frameworks at the outset provides a coherent basis for understanding the tool clusters presented later in this paper and situates the work within broader humanities discussions on sustainable, transparent, and reusable research software.

This paper addresses the challenges and opportunities of FAIRification in Computational Archaeology by presenting a curated set of lightweight, community-developed software tools used within the German National Research Data Infrastructure (NFDI), specifically in the NFDI4-Objects[1] consortium and in the context of the German chapter of the Computer Applications and Quantitative Methods in Archaeology association (CAA-DE). The focus lies on modular tools designed and maintained by the community-of-practice Research Squirrel Engineers Network, which actively contributes to Linked Open Data (LOD) infrastructures through tools such as the SPARQLing Unicorn Research Toolkit, the Jupyter Python Minions, and the NFDI4Objects Free and Open-Source Software (FOSS) Management Hub[2].

The presented software tools have been developed and used within the activities of NFDI4-Objects, CAA-DE, and the Special Interest Group on Scientific Scripting Languages in Archaeology (SIG SSLA[3]) of CAA International. These tools are embedded in applied semantic data pipelines involving Resource Description Framework (RDF) conversion, Linked Data querying, or geospatial visualisation in environments such as Quantum GIS (QGIS) or Jupyter Notebooks. All tools are made publicly available under open-source licences, documented in GitHub repositories, and in some cases formally archived with persistent identifiers and metadata following current recommendations for software citation. This reflects a wider trend in the digital humanities to treat software itself as a scholarly object, subject to citation, preservation, and critical evaluation.

Beyond presenting individual tools, the paper situates them within the broader conceptual and infrastructural context of FAIR-aligned research software in the humanities. It reflects on software reuse, semantic modelling, and sustainable community practices, linking each tool cluster explicitly to FAIR4RS considerations and the REAL framework. The approach taken here is pragmatic: instead of evaluating tools by abstract technical metrics, the paper contextualises them within ongoing research practices and open-science frameworks [SM20a, TM23b]. In doing so, it responds to current calls for greater transparency, explainability, and reproducibility

[1] cf. https://www.nfdi4objects.net

[2] This paper is based on the deRSE24 contributions IDs 16 (Research Software Engineering in NFDI4Objects: Community building, implementation of FAIRification Tools and scripting in Digital Archaeology, https://events.hifis.net/event/994/contributions/7979/) and 6 (Using RSE skills and Open Source Software to propose a heterogenious Management Hub in NFDI4Objects, https://events.hifis.net/event/994/contributions/7918/)

[3] cf. https://sslarch.github.io

in humanities-related software development. Several of the tools discussed are already in use or being integrated into Basic Services [RSS+24, HBF+24, HHG+24][4] of the NFDI, such as *nfdi.software*, *Jupyter4NFDI*, or the Knowledge Graph Infrastructure (*KGI4NFDI*), indicating growing convergence between community-driven practice and emerging infrastructure layers.

Section 2 outlines the criteria for tool selection and the limitations of scope. Section 3 presents the tools in four functional clusters: semantic FAIRification, lightweight scripting and visualisation, AI-based and data-driven methods, and community-based software management. Section 4 discusses these tools in light of FAIR4RS and the REAL framework, highlighting tensions between sustainability, domain specificity, and technical standardisation. Section 5 offers a summarising perspective and identifies paths forward for sustainable, community-driven software development in Computational Archaeology.

2 Methodology and Scope

This paper presents a curated overview of research software tools used within Computational Archaeology that contribute to the FAIRification of research data and workflows. The selection is informed by a pragmatic, practice-oriented perspective rooted in community-driven research software engineering. Rather than attempting to provide an exhaustive catalogue or comparative evaluation, the study focuses on demonstrators and prototypes developed and used by communities within the *NFDI4Objects* consortium, the German CAA chapter (CAA-DE), and related networks. This reflects how FAIRification in the humanities is typically advanced: not through top–down standardisation, but through iterative, bottom–up software reuse and domain knowledge exchange.

In addition to the FAIR4RS principles [BCK+22], the methodological framing draws on the REAL approach [ST25], which offers a complementary view on Reproducibility, Executability, Attribution, and Literalness in lightweight and exploratory software development. Stating these frameworks at the outset clarifies the analytical basis for the tool clusters introduced later in this paper and situates the methodology within broader discussions on transparency and methodological accountability in the digital humanities.

The tools included were identified based on the following criteria:

- they are actively used in applied research workflows within archaeology, computational heritage science, or the broader humanities;
- they are developed in the open and publicly accessible (e.g. via GitHub or Zenodo);
- they explicitly or implicitly support the FAIR principles for research data and/or follow FAIR4RS guidelines;
- they are embedded in or connected to semantic workflows (e.g., Linked Open Data, RDF, the SPARQL query language, ontology alignment);

[4] cf. nfdi.software: https://base4nfdi.de/projects/nfdi-software, Jupyter4NFDI: https://base4nfdi.de/projects/jupyter4nfdi, NGI4NFDI: https://base4nfdi.de/projects/kgi4nfdi

- and they have been referenced in presentations, workshops, or working groups within NFDI4Objects or deRSE and CAA-DE between 2021 and 2024.

Sources used to identify and contextualise these tools include project documentation, conference contributions (e.g., CAA, deRSE), GitHub repositories, tool publications with persistent identifiers, and community-maintained Wikis and knowledge bases. This paper does not aim to provide a systematic FAIR maturity assessment, to benchmark tools against formal software quality metrics, or to assess long-term sustainability. Instead, the methodology deliberately foregrounds the practical embedding of tools in domain-specific workflows and their function as reusable FAIRification services in a community-driven research environment. This aligns with current best practices for software citation and documentation, as recommended by Katz et al. [KH18, KNS+16] and related initiatives in the RSE community.

The tools are structured into four functional clusters, corresponding to the main phases and challenges of FAIRification in archaeological research: (1) semantic transformation and modelling; (2) lightweight querying and visualisation; (3) AI-enhanced data analysis; (4) community-driven infrastructure and software visibility. These categories serve as the basis for Section 3, where each tool is introduced with contextual examples and references to ongoing projects. By clarifying the scope and rationale here, the subsequent sections can more transparently link individual tools to FAIR4RS considerations and, where relevant, aspects of the REAL framework.

3 FAIRification Tools in Computational Archaeology

In this section, we introduce a set of lightweight, domain-specific research software tools that support FAIRification workflows in Computational Archaeology. They are organised into four functional groups reflecting different aspects of semantic data management, visualisation, analysis, and infrastructure support. The tools presented here have been developed within or alongside initiatives such as NFDI4Objects, CAA-DE, or the Research Squirrel Engineers Network, and demonstrate how pragmatic software engineering can enable reusable and interoperable research practices. Each subsection highlights specific tools and illustrates their role in enabling FAIR and FAIR4RS-compliant data workflows. Rather than presenting a catalogue of research software and scripts, the focus lies on how concrete tools operationalise selected aspects of FAIR and, in several cases, expose tensions described by the REAL framework in balancing reproducibility and executability with exploratory, domain-specific practice.

3.1 Tools for Semantic FAIRification

Semantic FAIRification involves the transformation and modelling of research data using formal ontologies and interoperable representations. In Computational Archaeology, this typically means converting structured or semi-structured domain data into RDF, aligning it with reference ontologies such as CIDOC CRM, and publishing it as Linked Open Data (LOD). Several tools have been developed to support this process in a modular, low-threshold way. From a FAIR perspective, these tools primarily target Interoperability and Reusability by expressing archaeological knowledge in shared, machine-actionable vocabularies, while also contributing to Findability and Accessibility through persistent identifiers and queryable endpoints. In the humanities context, they respond directly to long-standing challenges of integrating heterogeneous sources, documenting uncertainty, and making interpretative steps explicit.

Academic Meta Tool. The Academic Meta Tool (AMT) [UTM19] provides a framework consisting of a meta-ontology and a web tool (including JavaScript reasoning) to model vagueness in RDF data and perform reasoning that can be used in several contexts [MT21, TM22, TM23a, UTM19]. The main entities are *amt:Concept* and *amt:Role*, as well as axioms (*amt:Axiom*) such as *amt:RoleChainAxiom*, *amt:InverseAxiom*, *amt:DisjointAxiom* and *amt:SelfDisjointAxiom*. The vagueness is represented by a degree of connection (*amt:weight*) within a range of [0;1]. A person–interest network is shown here as an example. To implement this network, a separate application ontology must be developed. This ontology consists of two concepts, five roles and twelve axioms: concepts *Person* (P) and *Interest* (I). To link these concepts, roles are implemented which contain both associative relationships and inverse relationships between the persons and interests: P *connectedWith* P, P *interestedIn* I, I *interestOf* P, I_2 *subInterestOf* I_1, P_1 *superInterestOf* P_2. To create new knowledge in the network, five role-chain axioms are introduced, which allow conclusions to be drawn by choosing a suitable multi-valued logic: Axiom 01: *connectedWith connectedWith connectedWith* (ProductLogic), Axiom 02: *subInterestOf subInterestOf subInterestOf* (GoedelLogic), Axiom 03: *superInterestOf superInterestOf superInterestOf* (GoedelLogic), Axiom 04: *interestedIn interestOf connectedWith* (LukasiewiczLogic), Axiom 05: *interestedIn subInterestOf interestedIn* (GoedelLogic). The following conclusions can be drawn from the ontology and the role-chain axioms: relationships between persons are inverse and transitive. This results in chains of relationships, cf. Axiom 01. Hierarchical relationships between interests are also inverse and transitive, cf. Axioms 02/03. If two people are interested in the same thing, they are connected to each other, cf. Axiom 04. If a person is interested in something, they are also interested in the “super-interest”, cf. Axiom 05 [UTM19]. In terms of FAIR, AMT supports Interoperability and Reusability by encoding vagueness and reasoning rules in explicit RDF structures, and contributes to REAL-style Literalness by making uncertainty and inference mechanisms transparent in the underlying data model. For humanities use cases, this explicit treatment of vagueness is particularly relevant where interpretative, graded relationships must be preserved rather than flattened into binary assertions.

Alligator. The Alligator – Allen Transformator is based on the Alligator Method [TM23b], introduced at EAA Barcelona. The Alligator’s aim is to create a reproducible RDF representation of the state of knowledge concerning the temporal sequences of time intervals based on correspondence analysis (CA) and other algorithms. The “Alligator Method” converts the results of a CA into a relative chronology following Allen’s interval algebra, enabling temporal reasoning and visualisation in graphs or timelines, as well as export in standards such as RDF. The Alligator Method can be divided into several steps. First, an AGT file is generated from the correspondence analysis output in which the quantitative relations of the Euclidean distances between finds and proposed time intervals are calculated. Following the “horseshoe paradigm”, the CA result may provide a measure of (possible chronological) overlap between intervals. Then, the Alligator Algorithm starts with 3D distance calculations using the coordinates of the first three CA dimensions. One step further, the Alligator Algorithm calculates undated wobbly floating periods by finding the next 3D CA neighbour. Following this, the new virtual time intervals are calculated, and the resulting virtual fuzzy years are generated based on Allen's interval algebra. Any resulting virtual fuzzy year is then transformed into relative intervals and stored in RDF format, where all the initial and calculated results, as well as the Allen intervals, are stored. This is the basis for visualisation, e.g. as a graph or a timeline [TM23b]. A fictive example of Roman emperors serves as a method test.[5] The SPARQL Unicorn Ontology Documentation Tool [HT24b] can be used for better visualisation. The input data for the correspondence analysis includes information on, e.g., time intervals, emperors, and reign years. On top of that, each emperor has a starting and end year (e.g. Traian[6]: 98–117 AD), and some intervals have unknown floating starting and/or end years (e.g. the unknown time period of the fictive second Domitian consulate). The horizontal CA dimension axis defines the overlap between the time intervals. One aim is to date the Domitian Consulate 2 and get the virtual fuzzy start and end years. After performing the Alligator Algorithm, the fictive Domitian Consulate 2[7] period is dated to 81–96 AD, with Domitian as the nearest neighbour for the start and end dates. The Domitian Consulate 2 period is also related as equal to Domitian[8] using Allen's interval algebra [All83, Fre92]. The Alligator – Allen Transformer [TM24a] is available as web tool[9] and API[10] based on a Java application [TM24b]. Here, Reproducibility and Executability in the sense of FAIR4RS and REAL are supported by a scripted pipeline from CA output to RDF, while the explicit temporal relations in Allen algebra improve Interoperability for downstream reasoning tools. For archaeological periodisation problems, this makes heuristic steps and inferred chronologies more inspectable and, in principle, reusable beyond a single case study.

SPARQLing Unicorn Research Toolkit. In the humanities, data modelling, Linked Open Data and their analyses play a central role. Moreover, one community-driven open-source Linked Data repository recently gained momentum: Wikidata. The SPARQL Unicorn idea was born at the "Computer Applications and Quantitative Methods in Archaeology" conference in 2019 in Krakow, Poland. As one of the important parts of scientific conferences, networking and knowledge exchange (here in the famous Stary Port) brought interesting things to daylight: we all want to publish open data and support volunteer community-driven data-collecting initiatives like Wikidata. We propose the SPARQL Unicorn as a user-friendly, easy-to-use research toolkit for researchers working with Wikidata. The unicorn's aim is to help researchers use the community-driven data from Wikidata (and other Linked Open Data) and make it accessible (also called LOUD) without expertise in semantics, Linked Open Data or SPARQL using the SPARQL Unicorn principles: (1) describe your data in well-documented semantic structured open formats; (2) model, generate and publish your data as five-star Linked Open Data; (3) publish your data in Wikidata and interlink them to other resources in the Linked Open Data cloud; (4) use existing tools to query Wikidata dynamically and to do real-time data analysis and support developers or develop new tools to give people without any deeper knowledge in Linked

[5] cf. https://github.com/RGZM/alligator-mt-data/tree/master/v1
[6] cf. https://leiza-rse.github.io/alligator-mt-data/m0wQKa/index.html
[7] cf. https://leiza-rse.github.io/alligator-mt-data/VAvO57/index.html
[8] cf. https://leiza-rse.github.io/alligator-mt-data/BwyQ6x/index.html
[9] cf. https://tools.leiza.de/alligator/
[10] cf. https://api.leiza.de/alligator/

Open Data (SPARQL Unicorn tools) the possibility to also do dynamical real-time analysis; (5) use Wikidata and the SPARQL Unicorn tools in your own research and promote the SPARQL Unicorn principles so that other interested researchers in the community may start with principle 1 [TSHT20]. The SPARQLing Unicorn Research Toolkit comprises several research tools, especially the SPARQLing Unicorn QGIS Plugin [TH24b] and the SPARQL Unicorn Ontology Documentation [HT24b]. These two tools have been used in several interdisciplinary contexts [STT22, HT21, TH20b]. The SPARQLing Unicorn QGIS Plugin allows the execution of Linked Data queries in (Geo)SPARQL to selected triplestores and geo-enabled SPARQL endpoints. It thus prepares the results of the queries in QGIS for the geocommunity.

This can be community-driven SPARQL endpoints such as Wikidata or the NFDI4Objects Knowledge Graph (here archaeological sites of Irish Ogham stones [ST22], Figure 1). Furthermore, the SPARQL Unicorn Ontology Documentation Tool [TH24a, TSB^{+}24] is an extension of the SPARQLing Unicorn QGIS Plugin with the idea of converting an RDF dump to an enriched HTML deployment. The result should be ready to host on platforms such as GitHub and GitLab Pages. The documentation process should be usable as a continuous integration component as well, and the deployment should be done in such a way that it is useful for a maximum of research communities. Here, the main idea is to create static web pages that mimic already established Linked Open Data browsers, generate a Protégé-inspired class tree for navigation, create statistics of the dataset and publish them using VOID files and detect a variety of common vocabularies to create customised page widgets and pages: 3D model viewer, Leaflet view for geometries, dictionary view for OntoLex-Lemon dictionaries. A use-case example is the Campanian Ignimbrite findspots [TS23c, TS23a]. The RDF/Turtle data[11] can be transformed using GitHub Actions[12] to create GitHub Pages using a template [Thi24b]. This results in a dataset[13], accessed via GitHub Pages (e.g. all the findspots[14], Figure 2). One example is the findspot Urluia[15] in Romania [TS23d, TS23c, TS23b]. In FAIR terms, the toolkit strengthens Findability and Accessibility by exposing semantically rich datasets through standardised SPARQL endpoints and human-readable HTML documentation, while FAIR4RS aspects of Executability and Attribution are supported through openly available code, configuration templates, and citability of generated artefacts. For other humanities domains, the same pattern–semantic alignment, SPARQL-accessible endpoints and human-readable documentation–can be adopted with different ontologies and corpora.

From a cluster perspective, the semantic FAIRification tools presented here exemplify how relatively small, community-driven software projects can substantially improve Interoperability and Reusability of archaeological datasets. At the same time, they illustrate a REAL-style tension between lightweight, evolving prototypes and expectations of long-term sustainability and strict reproducibility, which is taken up again in the discussion. Crucially, they show how FAIR and REAL can be interpreted in ways that respect the interpretative, uncertainty-laden character of humanities data rather than forcing it into rigid pipelines.

[11] cf. https://github.com/Research-Squirrel-Engineers/campanian-ignimbrite-geo/blob/main/rdf/ci_full.ttl
[12] cf. https://github.com/Research-Squirrel-Engineers/campanian-ignimbrite-geo/blob/main/.github/workflows/main.yml
[13] cf. https://research-squirrel-engineers.github.io/campanian-ignimbrite-geo/CI_FSL/index.html
[14] cf. https://research-squirrel-engineers.github.io/campanian-ignimbrite-geo/Site_collection/index.html
[15] cf. https://research-squirrel-engineers.github.io/campanian-ignimbrite-geo/cisite_52/index.html

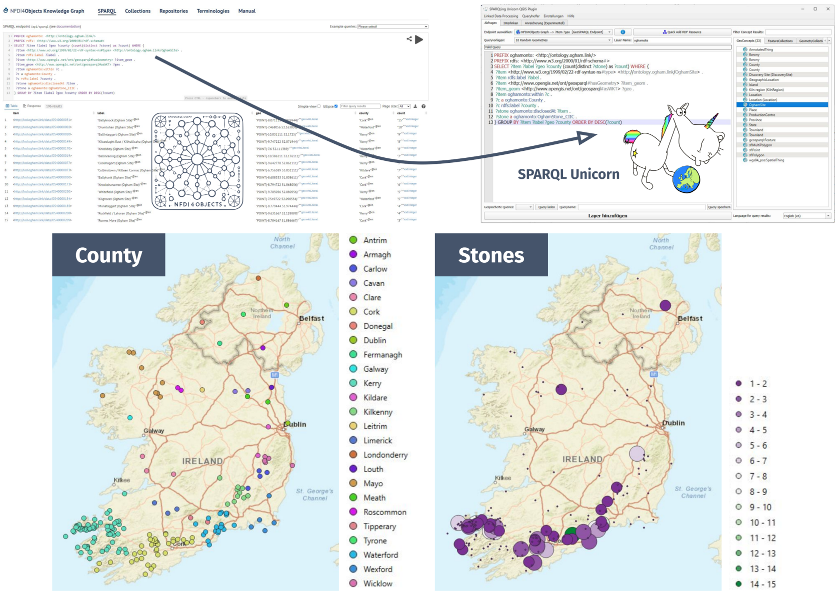

Figure 1: Schema for visualising a query of the NFDI4Objects Knowledge Graph for Irish Ogham stones, their county and quantity, via https://t1p.de/v0lg7, using the SPARQLing Unicorn QGIS Plugin. Florian Thiery, CC BY 4.0.

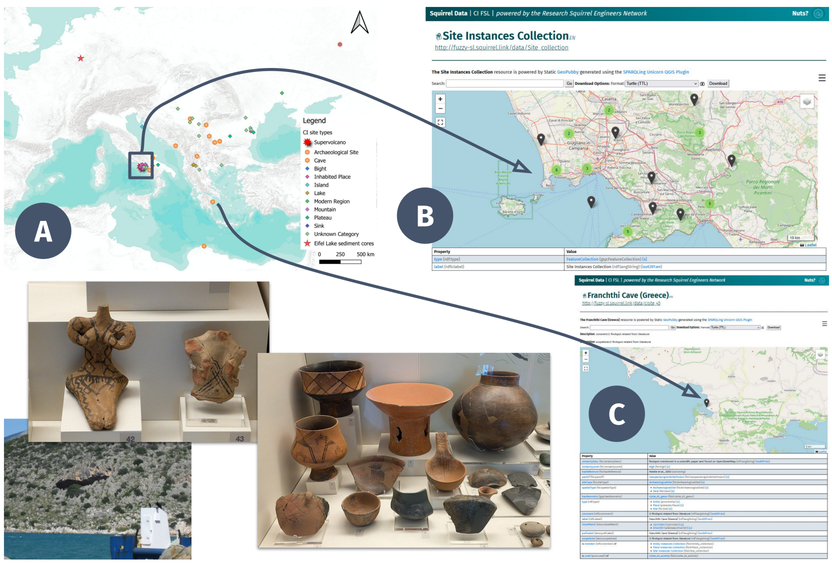

Figure 2: Campanian Ignimbrite (CI) sites created using the SPARQLing Unicorn Research Toolkit. A: CI sites categorised by type; B: Excerpt of the findspots near Naples; C/D: Franchthi Cave as cisite_45 with archaeological findings and cave entrance. A/B/C: CC BY 4.0, Florian Thiery & Fiona Schenk; C images: CC BY-SA 4.0, Zde (artefacts); Xalaros, CC BY (cave).

3.2 Lightweight Scripting and Visualisation

While semantic modelling and data transformation are essential for FAIRification, enabling domain researchers to interact with research data in everyday environments is equally critical. Lightweight scripting and visualisation tools serve as bridges between abstract data models and practical research workflows. They make semantic data usable in familiar applications, reduce technical barriers, and support interactive querying, mapping, and documentation. The scripting languages Python and R are widely used in Research Software Engineering applications in Computational Archaeology. The list of open-source archaeological software and resources *open-archaeo*[16] reflects this phenomenon as well, analysed in a survey of collaborative software engineering in archaeological research [Bat21]. The following sections showcase use cases and applications for R and Python in statistical and data-transforming domains. In FAIR4RS terms, these examples foreground Executability and Reproducibility by sharing scripts and workflows, while also touching on Attribution and Literalness where code and data are archived together. They further illustrate typical humanities scenarios in which legacy data, mixed-quality measurements and evolving research questions have to be handled pragmatically.

Introduction to R in Archaeology. There are a number of different ways the scripting language R is being used in archaeology, most of them connected to statistical analyses. The Rchaeologists group is a loose group of people who exchange news and discuss papers employing this scripting language. Though rarely part of any regular training for archaeologists (for Germany, see Schmidt 2024 [Sch24]), workshops at conferences, summer and winter schools as well as the development of courses in Digital and Computational Archaeology have increasingly educated archaeologists in using R. Linked to the open-science movement is the trend of publishing R code together with a publication. This practice seems to be on the rise in the field [SM20b]. The reproducible research tools package, *rrtools*, by Ben Marwick has been a guide for many to write papers in R Markdown and publish data and R Markdown files in the form of an R package. Ben Marwick also hosts a CRAN Task View for archaeologists as well as an overview of papers that publish R code [Mar22]. The published code ranges in variety and complexity: from the creation of relatively simple plots [HS19, SH19] to the creation of packages[17] [MS20], some of which are published on CRAN (e.g. *era*[18] by Joe Roe). In some cases old software is being transcribed into R, such as the Tools for Quantitative Archaeology that were developed by Keith Kintigh, and are in part rewritten in R by Matt Peeples[19] and the SIG SSLA [SSL24]. The following examples will showcase applications for R in archaeology. These developments directly contribute to FAIR4RS by promoting shared, citable code, and by integrating data, analysis scripts and narrative text in a single, reproducible research object. At the same time, they illustrate how reproducibility in practice often depends on social factors such as training, documentation and community norms.

[16] cf. https://open-archaeo.info/
[17] such as *percopackage* [SM22]
[18] cf. https://cran.r-project.org/web/packages/era/index.html
[19] cf. https://mattpeeples.net/

R in Archaeometallurgy. The implementation of statistical methods to archaeometallurgical data significantly predates the beginning of the development of scripting languages like R or Python. A notable example of this from the 1960s and 1970s is the project "Studies on the beginnings of metallurgy" (or SAM for short) by Siegfried Junghans, Edward Sangmeister and Manfred Schröder [JSS68]. The aim was to create clusters within which the trace element contents could be arranged as a normal distribution. A total of 22,000 analyses were published as part of the SAM project between 1960 and 1974. The researchers recognised that a statistical evaluation over such a large area requires an equally extensive basis in order to be statistically meaningful. In 1962, Klein and Sangmeister categorised these analyses into 29 groups based on 12,000 measurements [Kra03]. The correct way to assess such clusters remains an up-to-date question in archaeometallurgic research to this day [Pol18]. The earlier mentioned scripting languages offer a huge variety of tools, which can be implemented by archaeologists to improve the results of cluster analysis. Quite often it is necessary to use data of varying quality within one study, which were generated with different methods over a longer period of time. For such cases combined clustering methods can be used effectively. The R package *FactoMineR*[20] offers a wide variety of such methods. The function `HCPC` combines three methods. Firstly, a principal component analysis is performed to reduce the number of variables of the dataset. After that, a hierarchical cluster analysis with a given number of clusters is carried out. The central points of the clusters of these analyses are then used as the centroids of a k-means cluster analysis, which provides the final results. If this algorithm is combined with a range of methods for the assumption of the number of clusters (for example the gap, silhouette and elbow diagrams beside the traditional dendrogram) the statistical assumption of clusters can be significantly improved. Publishing R scripts and workflows for such analyses makes the decisions underlying complex clustering more transparent and, at least in principle, reproducible and reusable by other archaeometallurgical projects. For other materials-focused humanities domains, the same pattern can be adopted for large, historically grown measurement corpora.

Analysis with R on Irish Ogham data Ogham stones are monoliths bearing inscriptions in the early medieval Gaelic *primitive Irish* Ogham script [Mac45, Mac97], erected mainly on the island of Ireland and in the western part of Great Britain between the 4th and 9th centuries AD. The information on the Ogham stones and sites collected in the Linked Open Ogham Project[21] enables many analytical approaches [STT22, HT24b, SM20b, Thi22]. This section will concentrate on spatial analysis and exploring word counts performed before 2020 [ST22]. To import Wikidata's Ogham data into R, the package *WikidataQueryServiceR* [Pop20] is used. It is an API client for the Wikidata Query Service[22] that transforms the data retrieved from Wikidata through an input SPARQL query into a data frame. Furthermore, several R packages[23] were utilised to investigate the spatial relations between the Ogham sites and word frequencies. It is evident from the density and distribution analysis carried out using the R project *oghamaps* [Sch21a, Sch21b] that the majority of Ogham sites are located on the Dingle Peninsula (western hotspot in Figure 3,

[20] cf. http://factominer.free.fr/
[21] cf. https://ogham.link
[22] cf. https://query.wikidata.org/
[23] cf. R Core; for data cleaning and visualisation: *tidyr*, *reshape*, *ggplot2*, *viridis*; for spatial analysis: *ggspatial*, *rgdal*, *sp*

top-left). The percolation analysis, an explorative clustering algorithm [Mad20], which lends itself well to the identification of clusters and the establishment of densities and distances between archaeological sites [MS20], has been applied to identify clusters. Euclidean distance is used to create clusters from points within a certain radius of each other. This radius is then increased iteratively, and maps and summaries of the development of certain parameters concerning this radius are generated [SM20a]. In our example, the change in the mean cluster size is shown with respect to the clustering radius (Figure 3, bottom). A word analysis has been implemented based on the Ogham Extractor Tool[24] [TH20a, HT24a] and the table `words.csv`[25] to show how often words co-occur (Figure 3, top-right). The formula words *MAQI* (son) and *MUCOI* (tribe) are the most common. This, combined with the preponderance of names in the inscriptions, underscores the importance of familial affiliation in the communities that used the Ogham alphabet. Here, FAIR is advanced not only through open data and open code, but also by publishing reusable workflows that combine LOD access, spatial analysis and epigraphic statistics in a transparent way. The example also underlines how semantically enriched epigraphic datasets can serve as shared testbeds for methods that are of interest beyond a single project.

Stratigraphical reconstruction using Python. One of the benefits of Python is the number of off-the-shelf visualisation packages that allow for full user control. Combined with data analysis and mathematical packages, this allowed us to visualise and analyse the stratigraphy of excavations. By aligning the side sections in 3D space, it was easy to identify misalignments. What is more, using the spatial information of each layer, i.e. their x-, y-, z-coordinates in our 3D space, we were able to interpolate the 3D shape (i.e. surface) of the respective layer. The simplest approach is to use the standard Non-Uniform Rational B-Splines (NURBS) package for surface fitting.[26] Notably, material types influence the shape of the layer, so that curvatures derived from surface fitting can lead to unrealistic shapes. By using finds and their layer assignments as additional fitting data, we can improve the 3D reconstruction further. In multiple cases, the finds are associated with layers, but not with absolute depth (z coordinate), which means that the respective points need to be assigned with a depth range or uncertainty for surface fitting. We achieved the most realistic results by using one standard deviation of the z coordinate with the NumPy standard deviation function.[27] In our current work we add material properties to further adapt surface fitting behaviour. The approach allowed us to identify finds that were potentially assigned to the wrong stratigraphic layer [SPJ17]. Such examples show how lightweight Python tooling can be used to create executable and inspectable reconstructions of complex stratigraphies, which supports FAIR4RS-style reproducibility of analytical steps beyond static figures. More generally, they point to a pattern in which spatial and stratigraphic reasoning in the humanities can be encoded in reusable, parameterised scripts rather than remaining implicit in drawings or narratives.

[24] cf. https://linkedopenogham.github.io/o3d-epidoc-extractor/
[25] cf. https://github.com/ogi-ogham/oghamextractor/blob/master/words/words.csv
[26] cf. https://nurbs-python.readthedocs.io/en/5.x/fitting.html
[27] cf. https://numpy.org/doc/stable/reference/generated/numpy.std.html

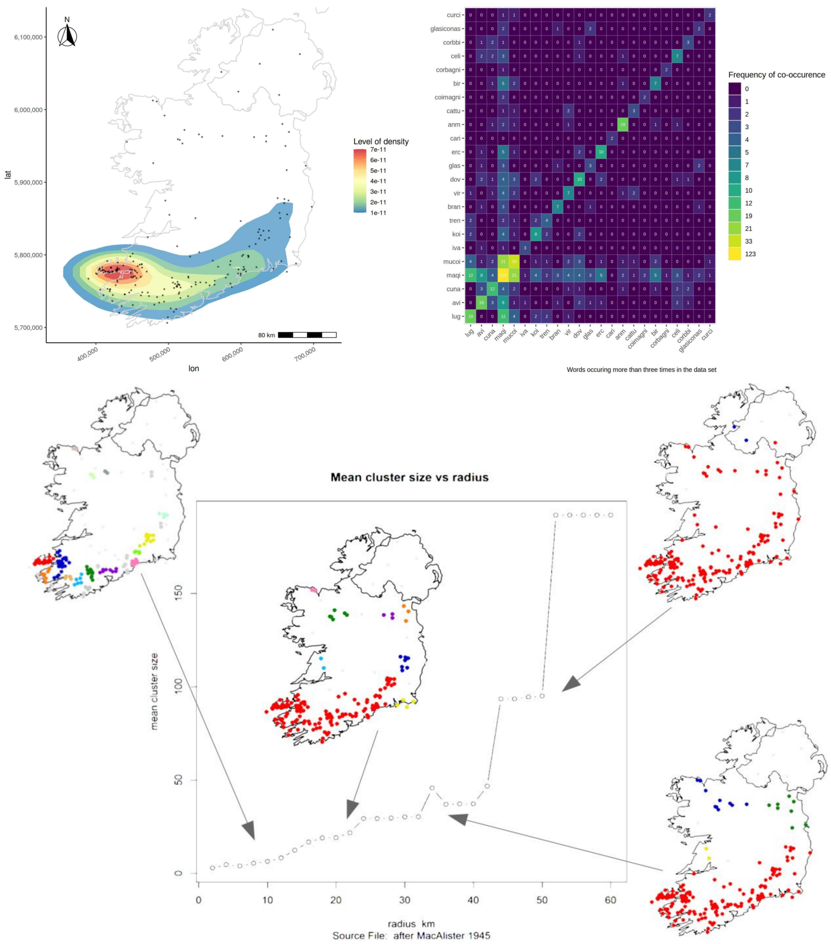

Figure 3: Top-left: CIIC Ogham Ireland density map created with R (*oghamaps*). Top-right: CIIC Ogham Ireland co-occurrence statistics created with R (*oghamaps*) for words that occur more than three times. Bottom: percolation analysis. Graph showing the mean cluster size at differing radii. The maps represent particularly significant points in the distribution (*oghamaps*). Graphics: Sophie C. Schmidt, CC BY 4.0.

Python to transform CSV to RDF Python scripts (little Python minions) can transform CSV data into RDF (Linked Open Data) using an underlying ontology. This digital method and simple approach are scalable to other disciplines and reproducible and can be used in, e.g., archaeology and geosciences. The resulting RDF data can be used in the SPARQL Unicorn to represent CSV origins as LOD. Simple scripts[28] used in the Brandenburg 5000 BC project [TS23e] (BB52K) transform CSV data[29] into RDF[30] using an ontology as data model.[31] The SPARQL

[28] cf. https://github.com/Research-Squirrel-Engineers/bb-5kbc/blob/main/py/fundplatz.py
[29] cf. https://github.com/Research-Squirrel-Engineers/bb-5kbc/blob/main/csv/fundplatz.csv
[30] cf. https://github.com/Research-Squirrel-Engineers/bb-5kbc/blob/main/rdf/site.ttl
[31] cf. https://github.com/Research-Squirrel-Engineers/bb-5kbc/blob/main/ontology/bb5bc.png

Unicorn Ontology Documentation Tool and GitHub Actions[32] create HTML pages[33] to click through. Similar scripts[34] have also been implemented to transform Irish Ogham stones [Mac45, Mac97, Thi22] data into Linked Open Ogham[35] [Thi24a]. With the help of the SPARQLing Unicorn QGIS Plugin, SPARQL queries can be applied to the dataset, such as all Ogham findspots in Ireland from the CIIC collection [Mac45] in the Barony of Corca Dhuibhne, the Dingle Peninsula. The same method can be applied to geosciences, such as the use case of Campanian Ignimbrite findspots [TS23c]. Python scripts[36] have been performed to create LOD from the CI findspots[37] [TS23b, SHB+24]. Here, the SPARQLing Unicorn QGIS Plugin can also create query-driven maps for findspots in caves. In FAIR terms, these "little minions" play the role of reusable, executable building blocks that encode provenance, semantic alignment and transformation steps in code rather than in opaque manual procedures. They also exemplify how humanities-oriented FAIRification often starts from simple, script-based pipelines rather than fully fledged infrastructures.

Taken together, these scripting and visualisation tools show how FAIR4RS-aligned practice in Computational Archaeology often emerges from small, reusable code fragments rather than from monolithic software systems. By encoding methodological decisions explicitly in scripts, notebooks and configuration files, they make analytical choices more inspectable and transferable across projects. At the same time, their dependence on evolving libraries, services and community know-how underlines the importance of shared documentation, training and social infrastructures, themes that reappear in the AI examples below and the broader discussion in Section 4.

3.3 Domain-specific Applications of AI and NLP

Besides explicit semantic modelling and low-threshold tooling, several research projects in Computational Archaeology have explored the use of AI and Natural Language Processing (NLP) to enhance data interpretation, classification, and structuring. These approaches operate at the interface between data-driven methods and knowledge modelling and often address challenges of ambiguity, volume, or heterogeneity in archaeological data. While still largely experimental, they demonstrate the potential of AI-based methods to support FAIRification processes by generating reusable annotations, structured metadata, or enriched semantic contexts. At the same time, they foreground REAL-style concerns about robustness and generalisability, as model performance can be highly dependent on corpus characteristics and expert supervision. In contrast to the more mature scripting and semantic tools described above, these AI workflows are best read as emerging, exploratory components within humanities FAIRification pipelines.

AI and Coins in the ClaReNet Project. Within the ClaReNet project, our team explored using computer vision (CV) techniques to analyse and work with ancient Celtic coins. Our efforts

[32] cf. https://github.com/Research-Squirrel-Engineers/bb-5kbc/blob/main/.github/workflows/main.yml
[33] cf. https://research-squirrel-engineers.github.io/bb-5kbc/
[34] cf. https://github.com/LinkedOpenOgham/ogham-lod/tree/main/rdf/py
[35] cf. https://github.com/LinkedOpenOgham/ogham-lod/tree/main/rdf
[36] cf. https://github.com/Research-Squirrel-Engineers/campanian-ignimbrite-geo/blob/main/py/CI_full.py
[37] cf. https://github.com/Research-Squirrel-Engineers/campanian-ignimbrite-geo/blob/main/rdf/ci_full.ttl

focused on different CV tasks: object detection, supervised image classification, unsupervised image classification and image matching. Rather than seeking new architectural innovations, our goal was to investigate the adaptability of existing methods to Celtic coin datasets and understand where these methods can support the experts in their work. One of our main datasets was the hoard of Le Câtillon II. We worked with Jersey Heritage, who allowed us to work with the hoard, which contained around 70,000 coins. The extraction, preparation and identification of these coins took around 10 years of exhaustive manual labour. The choice of CV tasks was closely related to a numismatist's workflow when analysing a coin dataset, reflecting the various stages of sorting used in numismatic analysis. Typically, this involves an initial sorting by denomination, followed by grouping based on similarity, ultimately leading to the identification of classes or types. Finally, differentiation by the dies is used to strike the coins. The aim of our study was to simulate these stages and to explore ways in which computational methods could support the work of numismatists. The main reason for using object detection was to crop the image, as only the coin itself was relevant for the following steps. By detecting the scale bar on the images, we could calculate and approximate the size of the coin. This allowed us to distinguish between quarter staters (small coins around 13 mm) and staters (around 22 mm) and add the approximate size to the data as it was not recorded (Figure 4). We then simulated sorting the hoard without extra information by applying unsupervised image classification and manually evaluating the resulting clusters. We identified strong clusters (high similarity) and clusters with corroded or broken coins, which we excluded from further processing. Such a sorting showed great potential to speed up cases with such high numbers of coins. To add one more evaluation step, we wanted to focus on the staters as we had labels for them, indicating to which of the six stater classes they belong (forming the ground truth). To extract a dataset based on our result, we combined the manually evaluated clusters with the size calculated in the step before and the information given (staters have a size of around 22 mm). By concentrating on the 22 mm $\pm$ 2 mm area, we were left with 26,000 coins in good condition, and by checking with the ground truth (GT) we could confirm that we were only extracting staters. With this cleaner and more focused dataset, we applied the unsupervised method again (with $k = 25$). The evaluation of the clusters showed that 18 out of 25 clusters had coins belonging to a class with a percentage of at least 79% up to 99% (the next below was 55%). This again confirms the potential of using such a method. We then took those 18 strong clusters and used the coins of one class as GT for training a supervised model. This means from the cluster with 79% we used those 79% (762 of 965 coins) and so forth, summing up to 13,855 coins. Using this supervised model, we re-evaluated the excluded coins in batches, first the misclassified coins in the strong cluster. Cases where the prediction and the GT did not correspond were discussed with the numismatic expert. This led to an improvement in the GT in cases where the model prediction, in fact, was correct (115 cases out of 328). Die identification for a die study is a different challenge than classification. An additional problem is that existing die studies are rare and difficult to validate. We had a die study given for one of the stater classes based on 1,355 coins with more than 40 assigned dies. We tried the same methods (unsupervised and supervised) as in the previous tasks. However, we got the best results by using image matching methods like *Oriented FAST and Rotated BRIEF* (ORB).[38] This approach requires no training material. It is based on a pairwise comparison by detecting

[38] cf. https://docs.opencv.org/3.4/d1/d89/tutorial_py_orb.html

key points in the image and trying to match them. As the coins are compared in pairs and a value is calculated between each pair, this could be stored as a vector for each coin. We then generated a hierarchical clustering on those vectors, allowing us to visualise coins with a low distance (higher similarity) close to each other in a tree structure. We performed this by using the Orange Data Mining tool, which provides the computation and interactive visualisation (Figure 5, left). The interactive tree allows the user to more easily inspect the results created by the algorithm. The interactivity was welcomed when presenting and discussing the work with numismatists, as it gave the expert a clearer understanding of how the sorting is done. We have further adapted it and added our own functionality so that the user can assign die names (or classes, etc.) and add comments, like working remarks, including uncertainties. We expect that the combination of the result of the algorithm, even if not perfect, with the interactive visualisation can enhance the manual work for a die study. For the staters of class six, this approach worked well, which could be validated with the given GT of the main numismatic expert for this hoard, Philip de Jersey. Philip is currently working with our approach to speed up the die study for class one (over 8,000 coins). However, it also needs to be emphasised that we tried this approach with Celtic coins called Bushel series, with much less promising results. We therefore think that it is important to keep a domain expert in the loop, and adjustments in the preprocessing need to be fine-tuned for each case separately. In terms of FAIR, such workflows can generate reusable annotations and derived features for large image corpora, but their robustness and transferability—key concerns in REAL-style reflections—must be carefully evaluated case by case. They are thus best understood as semi-automated aids to expert-driven interpretation, not as fully autonomous pipelines.

group	quantity (59442)
not identified	1778 (2.91%)
damaged coins	97 (0.16%)
*small coins**	3340 (5.64%)
Statere?	54,227 (91.29%)

*quater statere and 'petit billons'

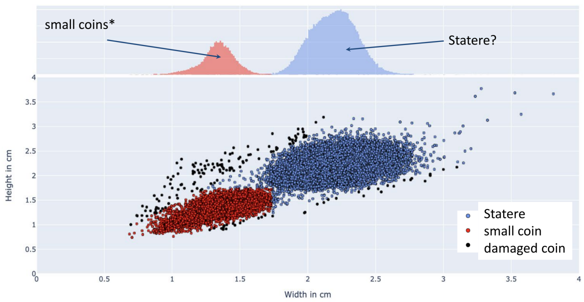

Figure 4: Scatter plot of the approximated diameter. Graphic: C. Deligio, Big Data Lab, CC BY 4.0.

NLP and AI in Conservation Science. To develop controlled vocabularies such as thesauri or ontologies, researchers must draw on their accumulated experience and knowledge to model appropriate systems. They may also examine and reuse existing models developed by colleagues or access lexicons. Research software engineering in natural language processing (NLP) can further assist this process by providing processed textual data, employing simple algorithms or more sophisticated machine-learning techniques. LEIZA employs natural language to document its projects and processes in conservation science. This includes domain-specific databases, publications, and other documents such as worksheets. The documents in question contain raw information that can be linked and enriched to establish semantic relationships. This ultimately

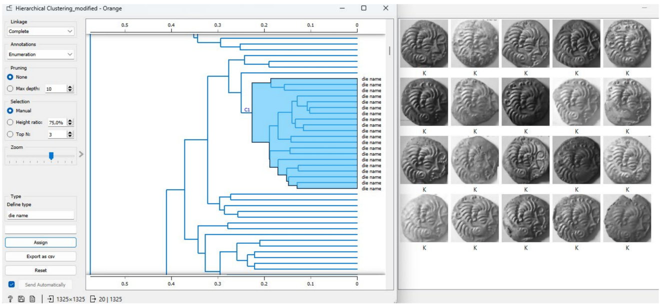

Figure 5: Left: Part of the dendrogram created with the hierarchical clustering widget in Orange Data Mining based on the results of ORB. Right: images of the currently selected coins in the left widget. Graphic: C. Deligio, Big Data Lab, CC BY 4.0, photos: Jersey Heritage.

facilitates the modelling process and provides a quantitative basis. A simple yet effective example is creating a word list from the documents sorted by frequency. This process allows the identification of key terms that recur throughout the corpus. Standard text mining techniques must be applied to achieve this, which involve cleaning and normalising the data. This process involves segmenting sentences and words into individual tokens, which can be achieved by applying rule-based methods or more sophisticated language models. Subsequently, terms are normalised to either lower or upper case and then lemmatised to their base form or stemmed to their root, thereby standardising their representation. Finally, the resulting tokens are refined to exclude stop words such as "and" or "he", which are ubiquitous in all texts and, therefore, have no informative value as key terms within a corpus. The tokens derived from this workflow, when applied to a first subset of the conservation worksheets in the LEIZA media database *easydb*, demonstrate its practicality. The term "object" emerges as the most frequent term in the list, with terms such as "glass", "fragment", or "material" also representing a genuine conservation vocabulary. Figure 6 (left) presents the results visually in the form of a word cloud, with more frequent terms displayed in larger font sizes. Another straightforward approach to identifying connections between the resulting words is to examine the words that frequently occur alongside a specific word in sentences (co-occurrences). This approach can be used to postulate a semantic connection between these words. For the vocabulary of the LEIZA conservation worksheets examined, Figure 6 (middle and right) illustrate the co-occurrences of the terms "shoe" and "glass". The outcomes also demonstrate the efficacy of the methodology employed: while "sole" and "leather" are terms that occur with greater frequency in the context of "shoe", "shard" and "part" are more typical companions of "glass". Incorporating *n*-grams, which are sets of consecutively occurring words, can also be considered for these simple methods.

For further investigation of the LEIZA conservation worksheets, it is initially proposed to process the entire corpus to establish a more robust data foundation for subsequent analysis. This encompasses enhanced text-recognition workflows for older worksheets only available as scanned images. Subsequently, more sophisticated algorithms, such as Latent Dirichlet Allocation [BNJ03], can be employed to more accurately identify related "topics" within a corpus using probabilistic methods. Additionally, machine learning can be leveraged for this purpose. The vector-spatial representation of words within a corpus as embeddings offers considerable potential for addressing semantic questions. By employing various unsupervised machine-learning techniques [MCCD13, PSM14], these can be represented as vectors within a space in a manner that places related terms closer to one another. Even "arithmetic operations" such as *king* − *man* + *woman* = *queen* are possible to a certain extent [AH19]. BERTopic combines, for example, the embeddings of transformer language models, clustering, and the TF–IDF measure for mining "topics" [Gro22]. An alternative way to identify the characteristic vocabulary of a corpus is to compare it with reference corpora. Various keyness measures can be used to identify words that are overrepresented compared to another corpus [BS10]. It is, therefore, possible to narrow down the corpus of LEIZA worksheets describing the conservation of archaeological objects more specifically to genuine conservation terms by comparing them with an archaeological reference corpus. In comparison to the modelling of thesauri, ontologies present a more significant challenge, as they are not hierarchies but rather graphs comprising semantic triples. A preliminary approach to acquiring data for their construction is the extraction of subject–verb–object (SVO) triplets from the corpus [SDFC15]. This can be achieved through the application of part-of-speech tagging to all sentences, which are then converted into parse trees.

Furthermore, there are potential applications for generative large language models like the popular ChatGPT in the field of text analysis. The input of impure text recognition, including instances of disjointed or faulty words, can be corrected to produce an output that is more accurate. However, in this case, the output texts are left to the statistical functioning of the models. It is essential to conduct validation to ascertain whether they are sufficiently accurate for the selected domain. In many cases, no free licences are defined for digital descriptions of conservation processes, such as those of borrowed objects or objects of unknown origin, including those from illicit excavations. Language models in which data reaches the provider via an API cannot be used for these items. Furthermore, the data protection of employees whose data is mentioned in the corpus cannot be guaranteed. A solution to this problem is currently being developed through the creation of numerous free, increasingly powerful language models that can be used locally on private computers, thereby preventing the leakage of sensitive data onto the internet. If a sufficient quantity of clean text files is already available, models can also be fine-tuned to the desired area of application. In terms of FAIR and REAL, this work illustrates both the promise and the limitations of AI-driven enrichment: it can create new, potentially reusable semantic structures and vocabularies, but also raises questions of transparency, attribution and control over sensitive data that must be addressed at infrastructure and policy levels. This again highlights that FAIR4RS and REAL cannot be reduced to technical checklists but must engage with legal, ethical and organisational constraints in humanities settings.

From a cluster perspective, these AI- and NLP-based approaches illustrate both the promise and the limits of automated FAIRification in the humanities. They can generate rich, reusable annotations, vocabularies and derived features at scales that would be difficult to achieve manually, yet they also depend on careful expert curation, corpus-sensitive parameter choices and appropriate governance of data and models. Read through the lens of FAIR4RS and REAL, they underscore that transparency, attribution and literalness are particularly challenging where opaque models intervene between raw data and published results. This tension motivates the more cautious, infrastructure-centred reflections developed in Section 4.

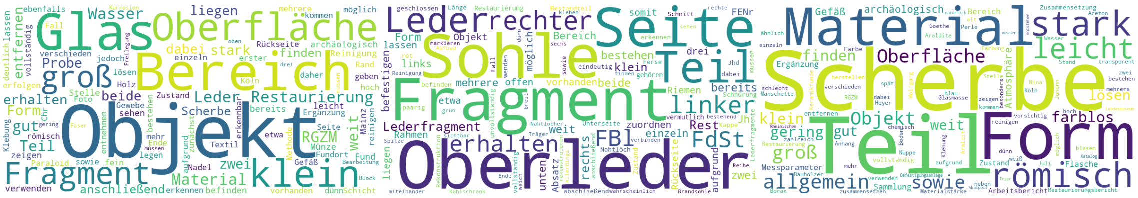

Figure 6: Word clouds of the most common terms (left) and of the most frequent co-occurrences of the terms "shoe" (middle) and "glass" (right) in the conservation worksheets of the LEIZA database. Graphic: Lasse Mempel-Länger, CC BY 4.0.

3.4 FOSS Management and Community Infrastructure

The Hub. A large number of involved institutions impose additional challenges for the day-to-day workflows as their IT infrastructure is usually restricted to being used by their own employees only. RSE skills and the combination of free and open-source software (FOSS) help create a management hub (Figure 7) that is uniformly usable for everybody, wherever the person may be employed. The central tool of our management hub is a Nextcloud instance. This instance provides a set of basic services (file sharing, task management, notes, shared calendars) and other FOSS services can be easily integrated. These services currently include Rocket Chat for communication, OpenProject for project launch planning, Zammad as a helpdesk ticketing system, Collabora for online document editing, and LimeSurvey combined with ShinyR for surveys. On top of that, reports could be imported/exported into a Wikibase using the included frameworks to create a management knowledge graph. Rocket Chat is not FOSS but is provided by the directorate of the NFDI e.V. and, therefore, is a central communication tool for the whole NFDI community. Lately, we have been preparing to add Pretalx and Pretix to the hub. Both pieces of software are conference-management tools. Pretalx handles the call-for-papers and the reviewing processes, and Pretix offers straightforward tools for the registration and ticketing of the conference attendees. Taken together, these components underpin FAIR4RS-aligned practices for software and service management, for instance by enabling shared, documented workflows, access-controlled yet transparent collaboration spaces, and the potential to record provenance and responsibilities in a structured way. For Computational Archaeology and related humanities consortia, such hubs also function as pedagogical spaces in which FAIR concepts are practised on everyday organisational data.

The Philosophy. As mentioned above, the core task of the NFDI is the construction of an up-to-date, sustainable, sovereign and powerful infrastructure for the needs of scientific research. The tools we combine into this infrastructure provide solutions for most tasks in the everyday working life of our consortium, and they do that based on the same design principles as those infrastructures which contain the research data itself. This way, we foster expertise and understanding of these principles within our community and practice using those technologies as developers and users. And maybe most importantly, we give ourselves a sustainable and sovereign, up-to-date infrastructure, which cannot be interrupted by changes in licensing models and financing. From a FAIR and REAL perspective, this management hub illustrates how organisational and social aspects of software use—not only code artefacts—contribute to long-term Reusability, accountability and resilience in research software ecosystems. In combination with the tool clusters described above, it forms the socio-technical backbone against which the discussion in Section 4 reflects the role of grassroots RSE for FAIRification in the humanities.

Viewed as a cluster, the management hub and its surrounding community structures make visible that FAIR4RS in Computational Archaeology is sustained as much by organisational routines and shared platforms as by individual tools. They offer concrete places where responsibilities, workflows and provenance can be negotiated, recorded and iteratively improved, complementing more formal basic services such as *nfdi.software* or *Jupyter4NFDI*. By foregrounding sovereignty, openness and collaboration, they help to stabilise otherwise fragile, prototype-driven software practices and provide an essential bridge between grassroots RSE initiatives and long-term infrastructural commitments.

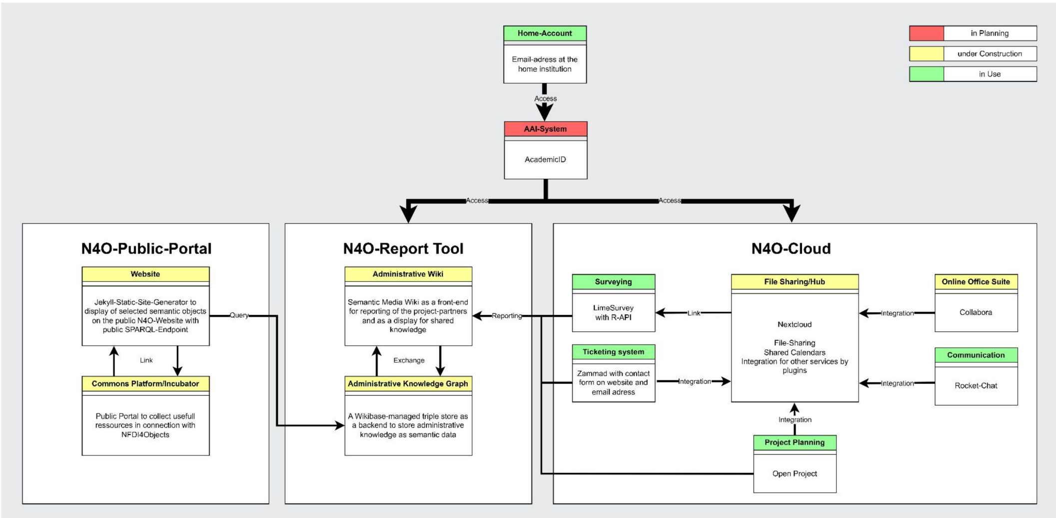

Figure 7: Scheme of the complete N4O management hub as of March 2024. Graphic: Fabian Fricke, CC BY 4.0.

4 Discussion

The tools and practices presented in this paper illustrate how community-driven research software in Computational Archaeology can support FAIRification workflows in modular, transparent, and reproducible ways. Taken together, the four clusters discussed in Section 3—semantic FAIRification, lightweight scripting and visualisation, AI-enhanced analysis, and FOSS-based management infrastructure—show that FAIR-aware practice in this domain is less a matter of a single, fully compliant tool than of interlocking services, scripts, and organisational arrangements. The deliberately detailed, almost "catalogue-like" presentation in Section 3 is therefore not intended as an exhaustive survey but as an empirical foundation for analysing how FAIR and FAIR4RS principles are implemented, negotiated, and challenged in real archaeological workflows. In this sense, the section functions as a set of grounded case studies rather than a typological overview.

From a FAIR4RS perspective [BCK+22], most tools address *Findability* and *Accessibility* by providing open-access repositories, documented workflows, and versioned code, often linked to persistent identifiers via Zenodo or GitHub. *Interoperability* is supported by using RDF, standard ontologies (e.g. CIDOC CRM), and semantic query languages such as SPARQL, particularly within the semantic FAIRification cluster. *Reusability*, however, emerges as the most variable dimension: it often depends on tacit community knowledge, informal documentation, or the continued availability of individual contributors—factors characteristic of humanities-oriented, bottom-up software development. The examples for R and Python demonstrate how reproducible workflows can be created in principle, while simultaneously revealing how sensitive such workflows remain to changing library versions, endpoint stability, or ambiguities not fully captured in comments or metadata.

Applying the REAL criteria [ST25] highlights further structural tensions:

- **Reproducibility** is supported through shared data pipelines and version-controlled notebooks, yet frequently limited by unstable endpoints, non-containerised environments, or dependency drift in evolving ecosystems.
- **Executability** varies widely, especially for tools that rely on external services (e.g. Wikidata, SPARQL endpoints, Solid Pods) or bespoke local setups ("little minions"), where environment configuration remains an implicit prerequisite.
- **Attribution** is partially institutionalised through DOIs, repository metadata and community-maintained documentation, but still distributed across heterogeneous platforms and therefore inconsistently captured.
- **Literalness**–the transparency of how a tool translates analytical assumptions into algorithmic representations–is most challenged where complex ontological logic or automated visualisation hides intermediate transformations.

These tensions should not be interpreted as shortcomings of individual tools but as typical characteristics of grassroots RSE in the humanities. Archaeological data are interpretative, uncertain, and heterogeneous by nature; software that interacts with such data must remain flexible and iterative. FAIR4RS and REAL therefore function less as prescriptive checklists than as orienting frameworks that help articulate where reproducibility and executability can realistically be strengthened—and where domain-specific needs necessitate controlled flexibility.

A further challenge concerns the relationship between community–developed tools and formal infrastructures. While services such as *nfdi.software*, *Jupyter4NFDI*, *TS4NFDI* or KGI4NFDI offer stable, standardised environments, many of the tools discussed here operate intentionally below that threshold. Their value lies in experimentation, conceptual openness, and rapid adaptation—qualities that are difficult to preserve when subjected to rigid lifecycle, security, or sustainability requirements. At the same time, greater infrastructural integration can enhance executability, persistence, and formal recognition. The tension between flexibility and standardisation therefore needs to be addressed through lightweight onboarding pathways, registries that accommodate prototypes, or "incubator" spaces for evolving tools.

Community structures—such as the *Research Squirrel Engineers Network*, the *SIG SSLA*, and the *Community Cluster Research Software Engineering* of NFDI4Objects—-play a crucial role in resolving or mitigating these tensions. They sustain tool knowledge, provide informal training, and enable peer-feedback mechanisms that function as social quality assurance. Importantly, they also mediate attribution, responsibility, and shared ownership in ways that traditional software citation practices do not yet fully capture, especially in collaborative humanities contexts.

Finally, the discussion highlights the limits of generalising FAIRification strategies across disciplines. The tools presented here implicitly encode archaeological assumptions—about spatial uncertainty, periodisation, linguistic variation, or material properties—and therefore cannot be transferred directly to other humanities or scientific domains without contextual adaptation. What emerges instead is a picture of FAIRification as a continuum: not every tool achieves full FAIR or FAIR4RS compliance, but collectively they advance the research landscape by enabling meaningful reuse, fostering interdisciplinary dialogue, and lowering the barrier for engaging with semantic and open data infrastructures. FAIR and REAL thus serve less as endpoints than as iterative, practice-oriented frameworks that help align community-developed tooling in Computational Archaeology with broader infrastructural and methodological standards.

5 Conclusion and Outlook

This paper has presented a practice-based overview of FAIRification tools developed and used in Computational Archaeology, with a focus on community-driven research software engineering. Rather than providing a comprehensive catalogue, it has used a set of concrete tool clusters as case studies to demonstrate how small-scale, modular, and open tools can contribute to the broader goals of FAIR and FAIR4RS, even when they are not part of formal infrastructure portfolios or standardised workflows. The analysed tools span multiple phases of the archaeological research data lifecycle—semantic modelling and data transformation, querying and visualisation, AI-supported interpretation, and community curation—and were grouped into four functional clusters. Taken together, they demonstrate that meaningful FAIRification can be achieved through lightweight, domain-embedded software practices, provided that semantic standards, transparency, and reusability are actively pursued.

At the same time, the discussion has shown that FAIRification is not a binary state but a continuum. Tools often excel in some FAIR4RS dimensions while remaining fragile in others, especially when it comes to sustainability, long-term reusability, or literal interpretability of results. This insight aligns with the REAL framework, which calls for a more differentiated view of software reproducibility, robustness, and generalisability. In this sense, FAIR and REAL have been used here not as evaluative checklists but as complementary vocabularies to describe characteristic tensions in grassroots RSE practice: between experimental flexibility and infrastructural stability, between executable prototypes and long-term stewardship, and between highly tailored domain tools and expectations of cross-disciplinary reuse. The case studies suggest that, for the humanities, "good enough" FAIRification often emerges incrementally, through iteratively improved workflows, documentation, and community conventions rather than through a single decisive intervention.

A key takeaway of this study is the importance of community infrastructure. Projects such as the FOSS N4O Management Hub, the SPARQLing Unicorn Toolkit, or the Jupyter Python Minions would not exist without the collaborative ethos of networks like the Research Squirrel Engineers, the SIG SSLA, or the community clusters of NFDI4Objects. These initiatives provide not only technical resources but also social frameworks for mutual learning, peer validation, and open innovation. They should be understood as critical enablers of sustainable research software practices, and as essential counterparts to formal infrastructures like *nfdi.software*, *Jupyter4NFDI*, or *KGI4NFDI*. Recognising and supporting such community structures—including their informal, experimental character—is therefore central to any realistic strategy for FAIR4RS implementation in the humanities.

Looking ahead, several steps are necessary to strengthen FAIRification in Computational Archaeology and related fields:

- improved metadata and software registration, e.g. through integration into *nfdi.software* and other services and registries such as *open-archaeo*;
- containerisation and versioning of demonstrators to ensure long-term executability and mitigate dependency drift;
- training and onboarding materials to lower the barrier for research data management, semantic modelling, and software citation practices;
- community-based quality assessment mechanisms to complement institutional reviews and encourage shared responsibility for software maintenance;
- conceptual alignment with FAIR Digital Objects (FDOs) and semantic core ontologies, in order to better connect LOD-centric and infrastructure-centric approaches and to support federated knowledge graph ecosystems.

The field of Computational Archaeology thus serves as a productive real-world laboratory for FAIR4RS implementation: shaped by historical data complexity, digital heterogeneity, and cross-domain collaboration. While the tools and workflows discussed here are firmly grounded in archaeological and heritage use cases, many of the underlying patterns—lightweight FAIRification scripts, semantic enrichment pipelines, community-run management hubs—are transferable, provided they are adapted to local data cultures and governance frameworks. By fostering bottom-up tool development, embracing semantic and infrastructural openness, and reflecting critically on the role of software in research, Computational Archaeology can help define good practices for other domains at the intersection of the humanities, computer science, and open data ecosystems.

Acknowledgements: This paper is part of the DFG-funded NFDI initiative, specifically the Research Data Infrastructure for the Material Remains of Human History (NFDI4Objects), DFG Project number 501836407. Some works were funded by Wikimedia Deutschland, Fellow-Programm Freies Wissen 2020/21: “Irische Ogham Steine im Wikimedia Universum”.

Bibliography

[AH19] C. Allen, T. Hospedales. Analogies Explained: Towards Understanding Word Embeddings. In *Proceedings of the 36th International Conference on Machine Learning*. Pp. 223–231. PMLR, May 2019. ISSN: 2640-3498.
https://proceedings.mlr.press/v97/allen19a.html

[All83] J. F. Allen. Maintaining knowledge about temporal intervals. *Communications of the ACM* 26(11):832–843, Nov. 1983.
doi:10.1145/182.358434

[Bat21] Z. Batist. Open archaeology: a survey of collaborative software engineering in archaeological research. zackbatist/caa2021-openarchaeo, June 2021.
https://github.com/zackbatist/caa2021-openarchaeo

[BCK+22] M. Barker, N. P. Chue Hong, D. S. Katz, A.-L. Lamprecht, C. Martinez-Ortiz, F. Psomopoulos, J. Harrow, L. J. Castro, M. Gruenpeter, P. A. Martinez, T. Honeyman. Introducing the FAIR Principles for research software. *Scientific Data* 9(1):622, Oct. 2022.
doi:10.1038/s41597-022-01710-x

[BNJ03] D. M. Blei, A. Y. Ng, M. I. Jordan. Latent Dirichlet allocation. *J. Mach. Learn. Res.* 3:993–1022, 2003. Publisher: JMLR.org.

[BS10] M. Bondi, M. Scott (eds.). *Keyness in Texts*. Studies in Corpus Linguistics 41. John Benjamins Publishing Company, Amsterdam, Nov. 2010.
doi:10.1075/scl.41

[Fre92] C. Freksa. Temporal Reasoning Based on Semi-Intervals. *Artif. Intell.* 54(1):199–227, 1992. QID: Q125700932.
doi:10.1016/0004-3702(92)90090-K

[Gro22] M. Grootendorst. BERTopic: Neural topic modeling with a class-based TF-IDF procedure. 2022. Version Number: 1.
doi:10.48550/ARXIV.2203.05794

[HBF+24] B. Hagemeier, A. Bleier, B. Flemisch, M. Lieber, K. Reuter, G. Dogaru. Jupyter4NFDI. *Zenodo* Jupyter4NFDI, July 2024.
doi:10.5281/zenodo.12699382

[HHG+24] M. Hammitzsch, B. Hetenyi, B. Grüning, A. Marcic, M. Löbe, L.-J. Castro, F. Engel, K. Höfner, M. Zacharias, O. Werth, V. Seibert, A. Rausch. nfdi.software - Initialisation Phase. *nfdi.software*, Dec. 2024. Publisher: Zenodo.
doi:10.5281/zenodo.14507302

[HS19] S. Hageneuer, S. C. Schmidt. Monumentality by numbers. In Buccellati et al. (eds.), *Size Matters - Understanding Monumentality Across Ancient Civilizations*. Pp. 291–308. transcript Verlag, Dec. 2019.
doi:10.1515/9783839445389-017

[HT21] T. Homburg, F. Thiery. Little Minions and SPARQL Unicorns as tools for archaeology. In Bogdani et al. (eds.), *ArcheoFOSS XIV 2020: Open Software, Hardware, Processes, Data and Formats in Archaeological Research*. Pp. 90–100. Archaeopress Publishing Ltd, Oxford, 2021.

[HT24a] T. Homburg, F. Thiery. o3d-epidoc-extractor. LinkedOpenOgham/o3d-epidoc-extractor/, Sept. 2024.
https://github.com/LinkedOpenOgham/o3d-epidoc-extractor/

[HT24b] T. Homburg, F. Thiery. SPARQL Unicorn Ontology Documentation. *Squirrel Papers* 6(2):#2, Mar. 2024. v0.17.
doi:10.5281/zenodo.10780476

[Hug24] J. Huggett. Changing Theory and Practice? CAA and Archaeology's Digital Turn. *Journal of Computer Applications in Archaeology* 7(1), Sept. 2024.
doi:10.5334/jcaa.144

[IEM+10] L. Isaksen, G. Earl, K. Martinez, S. Keay, N. Gibbins. Linking archaeological data. In Frischer et al. (eds.), *Making History Interactive. Computer Applications and Quantitative Methods in Archaeology (CAA): Proceedings of the 37 th International Conference*. Volume 2010, pp. 2–8. Archaeopress, Williamsburg, Virginia, USA, 2010. Num Pages: 8.
https://eprints.soton.ac.uk/204529/

[JSS68] S. Junghans, E. Sangmeister, M. Schröder. *Kupfer und Bronze in der Frühen Metallzeit Europas*. Gebr. Mann Verlag, Berlin, 1968.

[KH18] D. S. Katz, N. P. C. Hong. Software Citation in Theory and Practice. *arXiv* Computer Science(Computers and Society), 2018. Publisher: arXiv Version Number: 1.
doi:10.48550/ARXIV.1807.08149

[KNS+16] D. S. Katz, K. E. Niemeyer, A. M. Smith, W. L. Anderson, C. Boettiger, K. Hinsen, R. Hooft, M. Hucka, A. Lee, F. Löffler, T. Pollard, F. Rios. Software vs. data in the context of citation. Dec. 2016.
doi:10.7287/peerj.preprints.2630v1

[Kra03] R. Krause. *Studien zur kupfer- und frühbronzezeitlichen Metallurgie zwischen Karpatenbecken und Ostsee*. PhD thesis, Rahden/Westf., 2003.

[Mac45] R. A. S. Macalister. *Corpus inscriptionum insularum Celticarum*. Volume Vol. I. Stationery Office, Dublin, 1945. qid: Q70256237.

[Mac97] D. MacManus. *A Guide to Ogam*. Maynooth monographs 4. An Sagart, Maynooth, 1997. OCLC: 248434889 qid: Q70310399.

[Mad20] M. S. Maddison. Percolation analysis. In Gillings et al. (eds.), *Archaeological Spatial Analysis: A Methodological Guide*. Pp. 77–92. Routledge, 2020.
doi:10.4324/9781351243858

[Mar22] B. Marvick. CRAN Task View: Archaeological Science. benmarwick/ctv-archaeology/, Nov. 2022.
https://github.com/benmarwick/ctv-archaeology/

[MCCD13] T. Mikolov, K. Chen, G. Corrado, J. Dean. Efficient Estimation of Word Representations in Vector Space. 2013. Version Number: 3.
doi:10.48550/ARXIV.1301.3781

[MS20] M. S. Maddison, S. C. Schmidt. Percolation Analysis – Archaeological Applications at Widely Different Spatial Scales. *Journal of Computer Applications in Archaeology* 3(1):269–287, Sept. 2020.
doi:10.5334/jcaa.54

[MT21] A. Mees, F. Thiery. Un visualiseur et éditeur numérique sur Internet pour une ontologie des typologies de terre sigillée. *Actes des congrès de la Société Française d'Étude de la Céramique Antique en Gaule* Actes du Congrès d'Auch(2021):585–589, 2021.

[Pol18] M. Pollard. *Beyond provenance: New approaches to interpreting the chemistry of archaeological copper alloys*. University Press, Leuven, 2018.

[Pop20] M. Popov. WikidataQueryServiceR: API Client Library for 'Wikidata Query Service'. 2020.
https://CRAN.R-project.org/package=WikidataQueryServiceR

[PSM14] J. Pennington, R. Socher, C. Manning. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Pp. 1532–1543. Association for Computational Linguistics, Doha, Qatar, 2014.
doi:10.3115/v1/D14-1162

[RSS+24] L. Rossenova, R. Shigapov, M. Schubotz, F. Limani, B. Zapilko, K. U. Förstner. KGI4NFDI - Knowledge Graph Infrastructure for the German National Research Data Infrastructure. *Zenodo* Base4NFDI, July 2024.
doi:10.5281/zenodo.13118749

[Sch21a] S. C. Schmidt. oghamaps. *GitHub* ogi-ogham, Jan. 2021.
doi:10.5281/zenodo.3969519

[Sch21b] S. C. Schmidt. oghamaps. ogi-ogham/oghamaps, Jan. 2021.
https://github.com/ogi-ogham/oghamaps

[Sch24] S. C. Schmidt. Archäoinformatik als Teil eines zukunftsfähigen Lehrplans der Archäologie. *Archäologische Informationen* 46, 2024.
https://dguf.de/fileadmin/AI/archinf-ev_schmidt.pdf

[SDFC15] S. Sudhahar, G. De Fazio, R. Franzosi, N. Cristianini. Network analysis of narrative content in large corpora. *Natural Language Engineering* 21(1):81–112, Jan. 2015.
doi:10.1017/S1351324913000247

[SH19] S. C. Schmidt, S. Hageneuer. Monumentality by numbers. *OSF*, 2019. Publisher: OSF.
doi:10.17605/OSF.IO/3AJ7V

[SHB+24] F. Schenk, U. Hambach, S. Britzius, D. Veres, F. Sirocko. A Cryptotephra Layer in Sediments of an Infilled Maar Lake from the Eifel (Germany): First Evidence of Campanian Ignimbrite Ash Airfall in Central Europe. *Quaternary* 7(2):17, Mar. 2024.
doi:10.3390/quat7020017

[SM20a] S. Schmidt, M. Maddison. Percolation Analysis. Apr. 2020.
doi:10.17605/OSF.IO/7EXTC

[SM20b] S. C. Schmidt, B. Marwick. Tool-Driven Revolutions in Archaeological Science. 3(1):18–32, Jan. 2020.
doi:10.5334/jcaa.29

[SM22] S. Schmidt, M. Maddison. Percolation Analysis. *OSF*, Aug. 2022. Publisher: Open Science Framework.
doi:10.17605/OSF.IO/7EXTC

[SPJ17] L. Schubert, A. Predoi, K. Jeffery. Analysing the archaeological context: Reconstructing stratigraphic layers. Volume 1863. 2017.
doi:10.1063/1.4992196

[SSL24] SSLA - Scientific Scripting Languages in Archaeology. Tools for Quantitative Archaeology – in R. sslarch/tfqar, Oct. 2024.
https://github.com/sslarch/tfqar

[ST22] S. C. Schmidt, F. Thiery. SPARQLing Ogham Stones: New Options for Analyzing Analog Editions by Digitization in Wikidata. *CEUR Workshop Proceedings* 3110(Graph Technologies in the Humanities 2020):211–244, Mar. 2022. Publisher: Zenodo.
doi:10.5281/zenodo.6380914

[ST25] L. K. Schubert, F. Thiery. Keeping it REAL: Sustainability Challenges for Research Software in Computational Archaeology. *Squirrel Papers* 7(3):L4, Feb. 2025.
doi:10.5281/zenodo.14898291

[STT22] S. C. Schmidt, F. Thiery, M. Trognitz. Practices of Linked Open Data in Archaeology and Their Realisation in Wikidata. *Digital* 2(3):333–364, June 2022.
doi:10.3390/digital2030019

[TH20a] F. Thiery, T. Homburg. oghamextractor. *GitHub* ogi-ogham, Aug. 2020.
doi:10.5281/zenodo.3969514

[TH20b] F. Thiery, T. Homburg. QGIS - A SPARQLing Unicorn? Eine Einführung in Linked Open Geodata zur Integration von RDF in QGIS Plugins. In *FOSSGIS 2020 - Anwenderkonferenz für Freie und Open Source Software für Geoinformationssysteme*. Zenodo, Mar. 2020.
doi:10.5281/zenodo.3719127

[TH24a] F. Thiery, T. Homburg. The SPARQL Unicorn Ontology documentation: Exposing RDF geodata using static GeoAPIs. *Tagungsband FOSSGIS-Konferenz 2024* 2024:82–90, Mar. 2024.
doi:10.5281/zenodo.10570985

[TH24b] F. Thiery, T. Homburg. SPARQLing Unicorn QGIS Plugin. *Squirrel Papers* 6(2):#1, Mar. 2024. v0.17.
doi:10.5281/zenodo.10779466

[Thi22] F. Thiery. SPARQLing Ogham – Irische Ogham-Steine als Linked Open Data. *ZfdG - Zeitschrift für digitale Geisteswissenschaften* Fabrikation von Erkenntnis – Experimente in den Digital Humanities.(Sonderband 5), 2022. Medium: HTML,XML,PDF Publisher: Zeitschrift für digitale Geisteswissenschaften - ZfdG Version Number: 1.0.
doi:10.17175/SB005_010

[Thi24a] F. Thiery. LinkedOpenOgham / ogham-lod. *Squirrel Papers* 5(6):#7, May 2024. GitHub-Repository.
https://github.com/LinkedOpenOgham/ogham-lod

[Thi24b] F. Thiery. unicorn-ontdoc-template-squirrels. Research-Squirrel-Engineers/unicorn-ontdoc-template-squirrels/, May 2024.
https://github.com/Research-Squirrel-Engineers/unicorn-ontdoc-template-squirrels/

[TM22] F. Thiery, A. Mees. CeraTyOnt @ AMT. *Squirrel Papers* 4(2):#2, Oct. 2022. v1.0.
doi:10.5281/zenodo.7143093

[TM23a] F. Thiery, A. Mees. Taming Ambiguity - Dealing with doubts in archaeological datasets using LOD. *CAA 2018: Human History and Digital Future*, Oct. 2023. Publisher: Tübingen University Press.
doi:10.15496/PUBLIKATION-87762

[TM23b] F. Thiery, A. W. Mees. Alligator Method. *Squirrel Papers* 5(1):#2, Feb. 2023.
doi:10.5281/zenodo.7624870

[TM24a] F. Thiery, A. Mees. alligator - Allen Transformer. *Squirrel Papers* 6(2):#4, Apr. 2024. v1.1.
doi:10.5281/zenodo.2540709

[TM24b] F. Thiery, A. W. Mees. alligator - Allen Transformer. leiza-rse/alligator, Apr. 2024.
https://github.com/leiza-rse/alligator

[TS23a] F. Thiery, F. Schenk. Campanian Ignimbrite Geo Locations. Research-Squirrel-Engineers/campanian-ignimbrite-geo, Dec. 2023.
https://github.com/Research-Squirrel-Engineers/campanian-ignimbrite-geo

[TS23b] F. Thiery, F. Schenk. Campanian Ignimbrite Geo Locations. *Squirrel Papers* 5(2):#2, 2023.
doi:10.5281/zenodo.10361309

[TS23c] F. Thiery, F. Schenk. CI Site 52: Urluia (Romania). *Squirrel Papers* Research Squirrel Engineers, via @campanian-ignimbrite-geo, 2023.
http://fuzzy-sl.squirrel.link/data/cisite_52

[TS23d] F. Thiery, F. Schenk. How to locate the Campanian Ignimbrite site Urluia based on literature? How to provide and publish this data in a FAIR way? *Squirrel Papers* 5(1):#5, 2023.
doi:10.5281/zenodo.10262720

[TS23e] F. Thiery, S. C. Schmidt. Research-Squirrel-Engineers / bb-5kbc. *Squirrel Papers* 5(6):#3, Nov. 2023. GitHub-Repository.
https://github.com/Research-Squirrel-Engineers/bb-5kbc/

[TSB+24] F. Thiery, F. Schenk, S. Baars, K. Tolle, P. Thiery. Modellierung von Fuzzyness / Wobbliness in Geodaten. *Tagungsband FOSSGIS-Konferenz 2024* 2024:64–73, Mar. 2024.
doi:10.5281/zenodo.10571859

[TSHT20] F. Thiery, S. C. Schmidt, T. Homburg, M. Trognitz. The SPARQL Unicorn: An introduction. *Squirrel Papers* 2(1):#1, Apr. 2020.
doi:10.5281/zenodo.3742185

[UTM19] M. Unold, F. Thiery, A. Mees. Academic Meta Tool. Ein Web-Tool zur Modellierung von Vagheit. *ZfdG - Zeitschrift für digitale Geisteswissenschaften* Die Modellierung des Zweifels – Schlüsselideen und -konzepte zur graphbasierten Modellierung von Unsicherheiten.(Sonderband 4), 2019.
doi:10.17175/SB004_004

[WDA+16] M. D. Wilkinson, M. Dumontier, I. J. Aalbersberg, G. Appleton, M. Axton, A. Baak, N. Blomberg, J.-W. Boiten, L. B. Da Silva Santos, P. E. Bourne, J. Bouwman, A. J. Brookes, T. Clark, M. Crosas, I. Dillo, O. Dumon, S. Edmunds, C. T. Evelo, R. Finkers, A. Gonzalez-Beltran, A. J. Gray, P. Groth, C. Goble, J. S. Grethe, J. Heringa, P. A. 'T Hoen, R. Hooft, T. Kuhn, R. Kok, J. Kok, S. J. Lusher, M. E. Martone, A. Mons, A. L. Packer, B. Persson, P. Rocca-Serra, M. Roos, R. Van Schaik, S.-A. Sansone, E. Schultes, T. Sengstag, T. Slater, G. Strawn, M. A. Swertz, M. Thompson, J. Van Der Lei, E. Van Mulligen, J. Velterop, A. Waagmeester, P. Wittenburg, K. Wolstencroft, J. Zhao, B. Mons. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data* 3(1):160018,

Mar. 2016.
doi:10.1038/sdata.2016.18