
# From Code Repositories to Knowledge Graphs of Research Software Metadata

**Daniel Garijo, Ontology Engineering Group, Universidad Politécnica de Madrid, Spain**

daniel.garijo@upm.es

@dgarijov

# Research Software is one of the pillars of Open Science


Research Data

Research Software

Research Methods

# Research Software is one of the pillars of Open Science


Research Data

Research Software

Research Methods

figshare

github
SOCIAL CODE HOSTING

Apache

PLOS

DRYAD

The Dataverse Network Project

GitLab

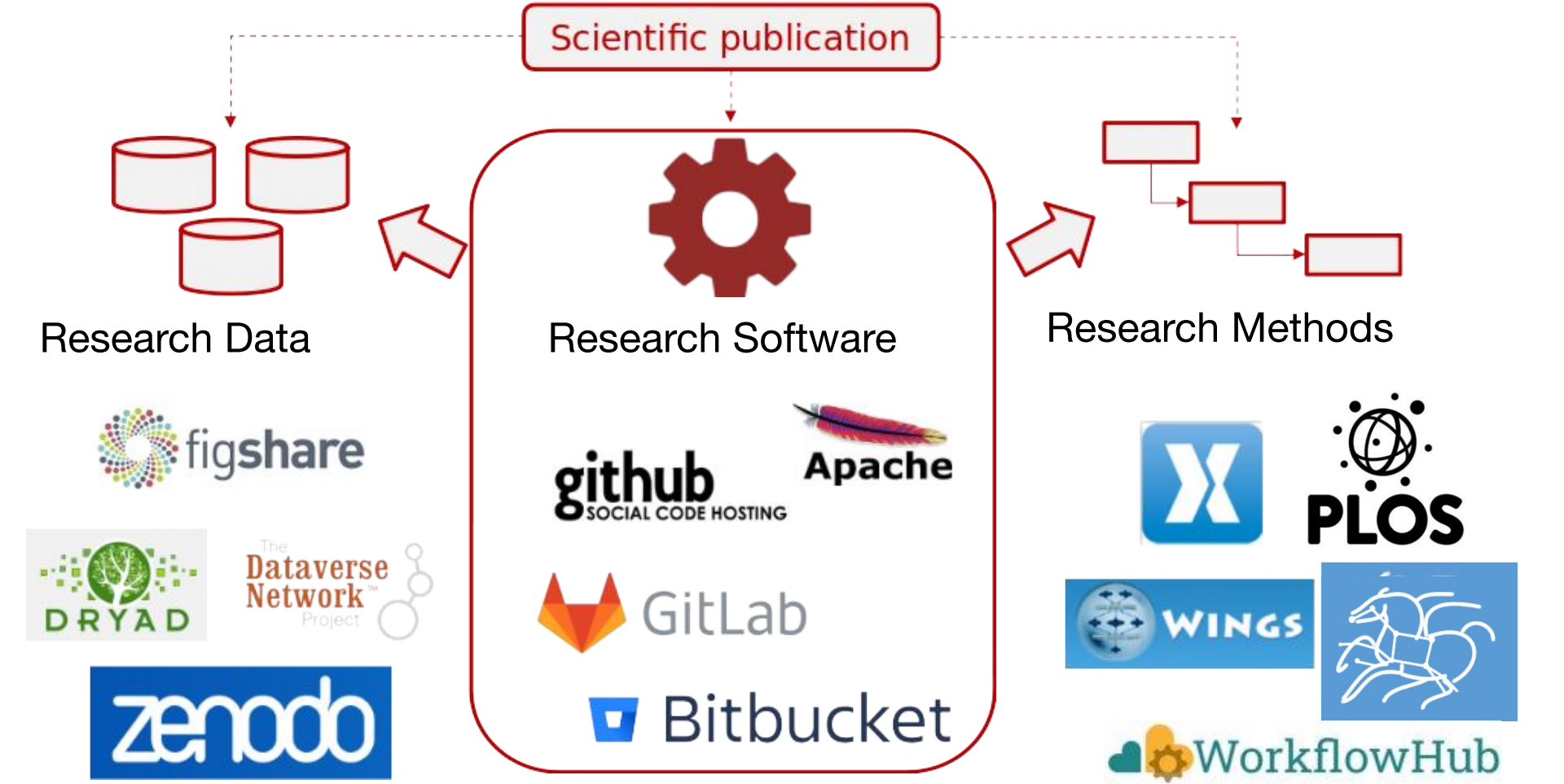
WINGS

zenodo

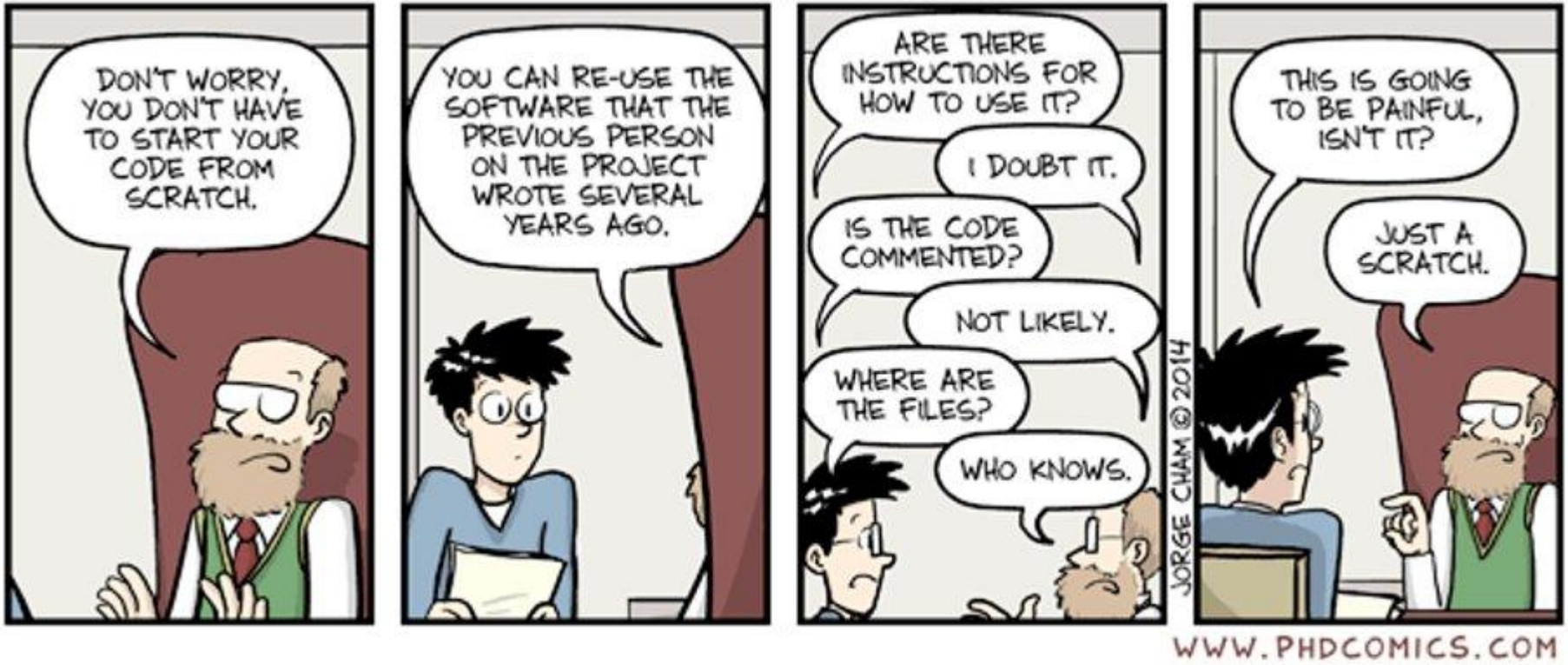
Bitbucket

WorkflowHub

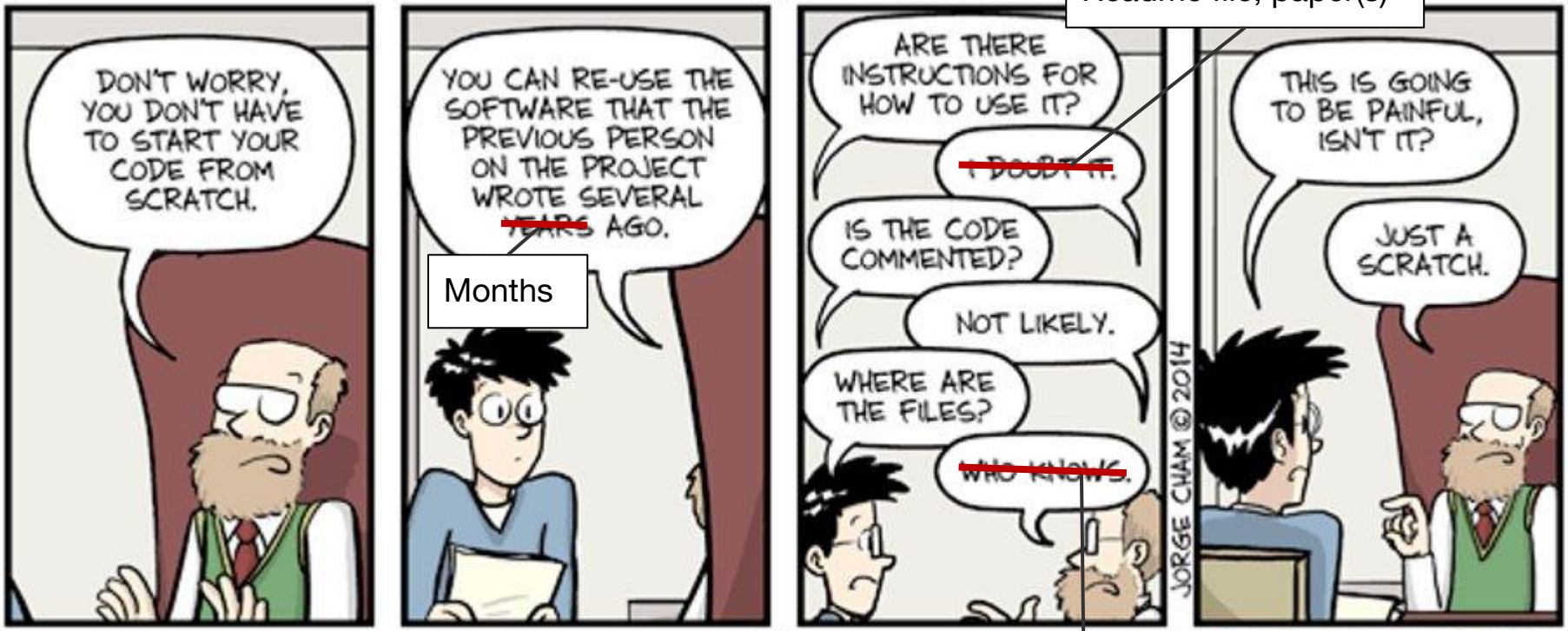
# Research Software is one of the pillars of Open Science



# Is this situation familiar?



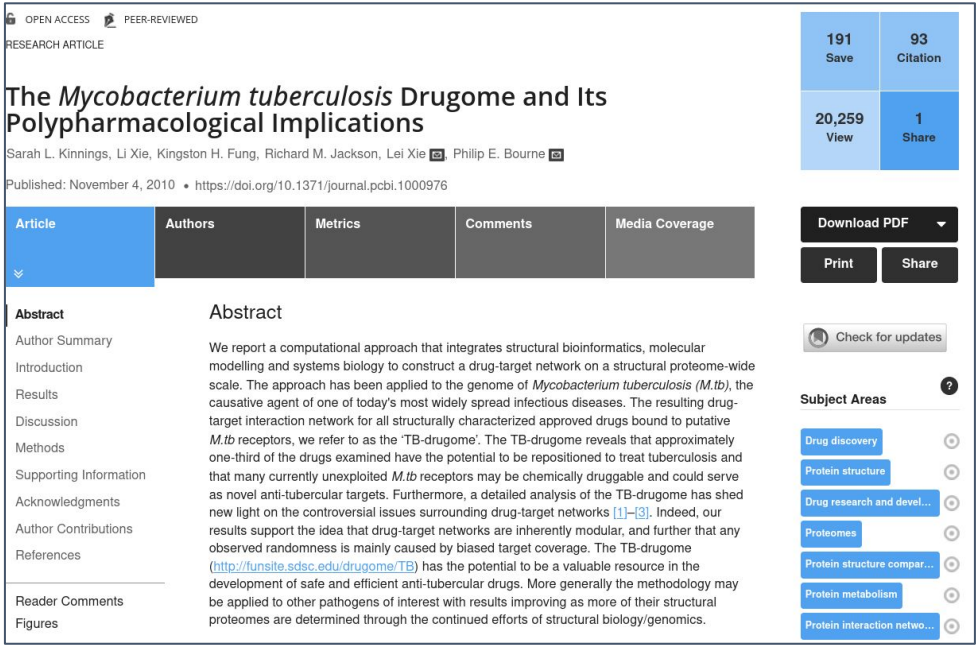
# Challenge 1: Lack of structured (machine-readable) documentation



# Challenge 2: Reusability takes time

In [1] we tried to reproduce an effort from **one** year before.

- All data were available online
- All tools were available online (except one, but authors had a replacement)
- > **250 hrs to full reproducibility**
- > **100 hrs** to get familiar with the tools and their I/O



OPEN ACCESS PEER-REVIEWED

RESEARCH ARTICLE

The *Mycobacterium tuberculosis* Drugome and Its Polypharmacological Implications

Sarah L. Kinnings, Li Xie, Kingston H. Fung, Richard M. Jackson, Lei Xie, Philip E. Bourne

Published: November 4, 2010 • https://doi.org/10.1371/journal.pcbi.1000976

191 Save | 93 Citation | 20,259 View | 1 Share

Article | Authors | Metrics | Comments | Media Coverage

Download PDF | Print | Share

Abstract
Author Summary
Introduction
Results
Discussion
Methods
Supporting Information
Acknowledgments
Author Contributions
References
Reader Comments
Figures

Abstract

We report a computational approach that integrates structural bioinformatics, molecular modelling and systems biology to construct a drug-target network on a structural proteome-wide scale. The approach has been applied to the genome of *Mycobacterium tuberculosis (M.tb)*, the causative agent of one of today's most widely spread infectious diseases. The resulting drug-target interaction network for all structurally characterized approved drugs bound to putative *M.tb* receptors, we refer to as the 'TB-drugome'. The TB-drugome reveals that approximately one-third of the drugs examined have the potential to be repositioned to treat tuberculosis and that many currently unexploited *M.tb* receptors may be chemically druggable and could serve as novel anti-tubercular targets. Furthermore, a detailed analysis of the TB-drugome has shed new light on the controversial issues surrounding drug-target networks [1]–[3]. Indeed, our results support the idea that drug-target networks are inherently modular, and further that any observed randomness is mainly caused by biased target coverage. The TB-drugome (http://funsite.sdsc.edu/drugome/TB) has the potential to be a valuable resource in the development of safe and efficient anti-tubercular drugs. More generally the methodology may be applied to other pathogens of interest with results improving as more of their structural proteomes are determined through the continued efforts of structural biology/genomics.

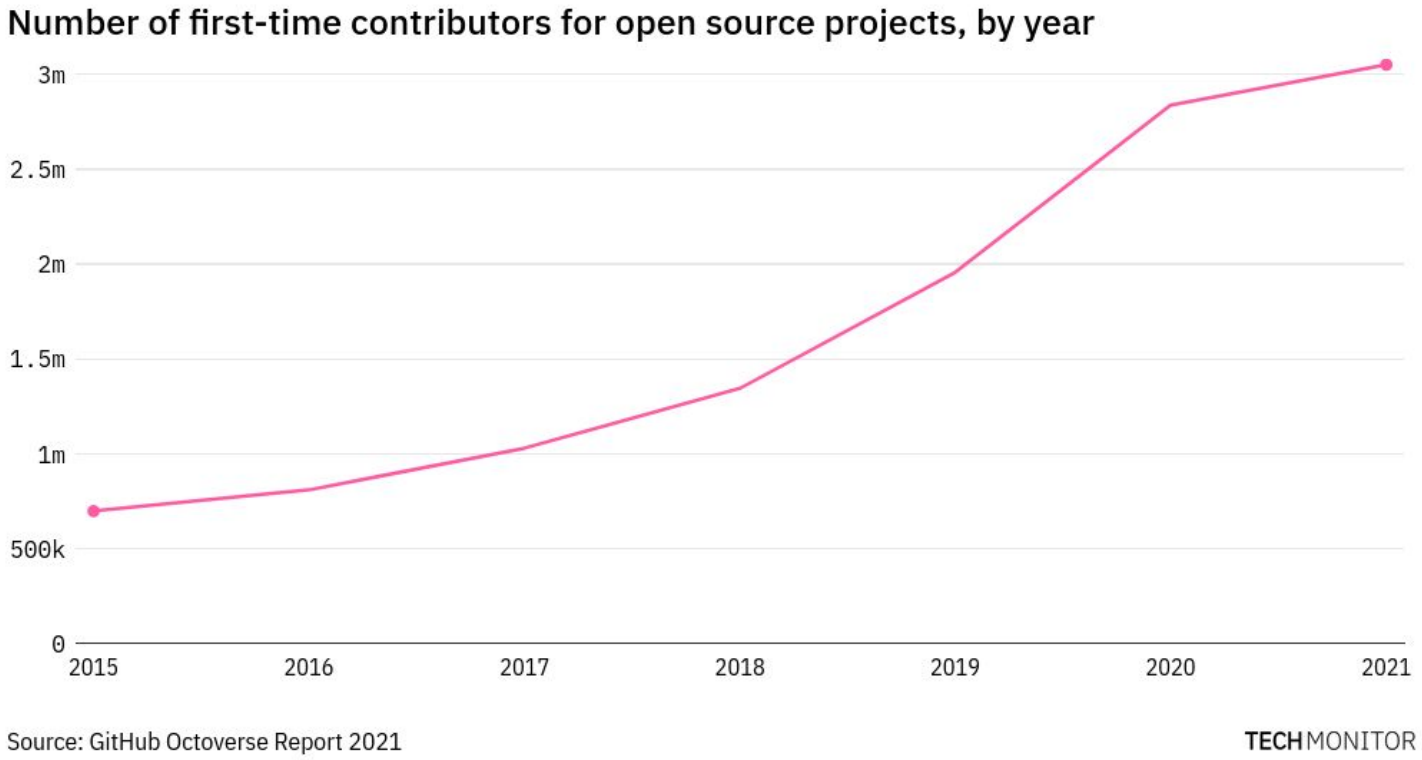
Check for updates

Subject Areas

Drug discovery | Protein structure | Drug research and devel... | Proteomes | Protein structure compar... | Protein metabolism | Protein interaction netwo...

[1] Garijo, D., Kinnings, S., Xie, L., Xie, L., Zhang, Y., Bourne, P. E., & Gil, Y. (2013). Quantifying reproducibility in computational biology: the case of the tuberculosis drugome. *PloS one*, *8*(11), e80278.

Millions of open-source repositories are updated/created every year



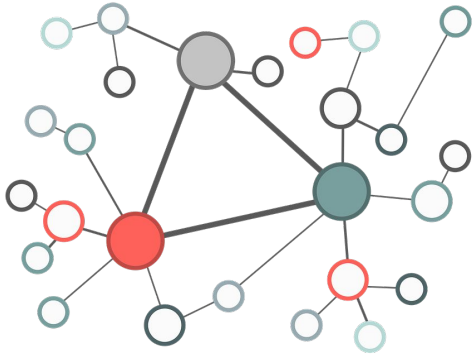
[Hucka et al]: Scientists still rely on three main methods for searching new software:

- Survey
- Recommendation from a colleague
- Search engine

M. Hucka and M. J. Graham, “Software search is not a science, even among scientists: A survey of how scientists and engineers find software,” Journal of Systems and Software, vol. 141, pp. 171–191, 2018.

1. Structured representation
2. Reuse
3. Compare
4. Search

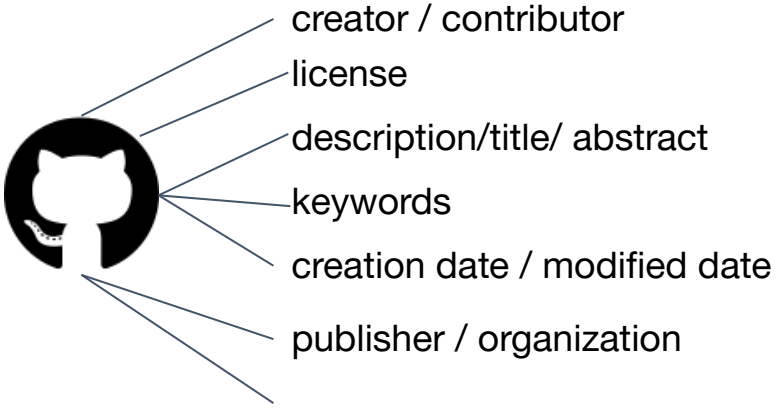
1. Structured representation
2. Reuse
3. Compare
4. Search



**Knowledge Graphs**

# **Creating KGs of Research Software metadata**:
Representing RS at different levels of abstraction

Why is it needed?
- Search
- Compare

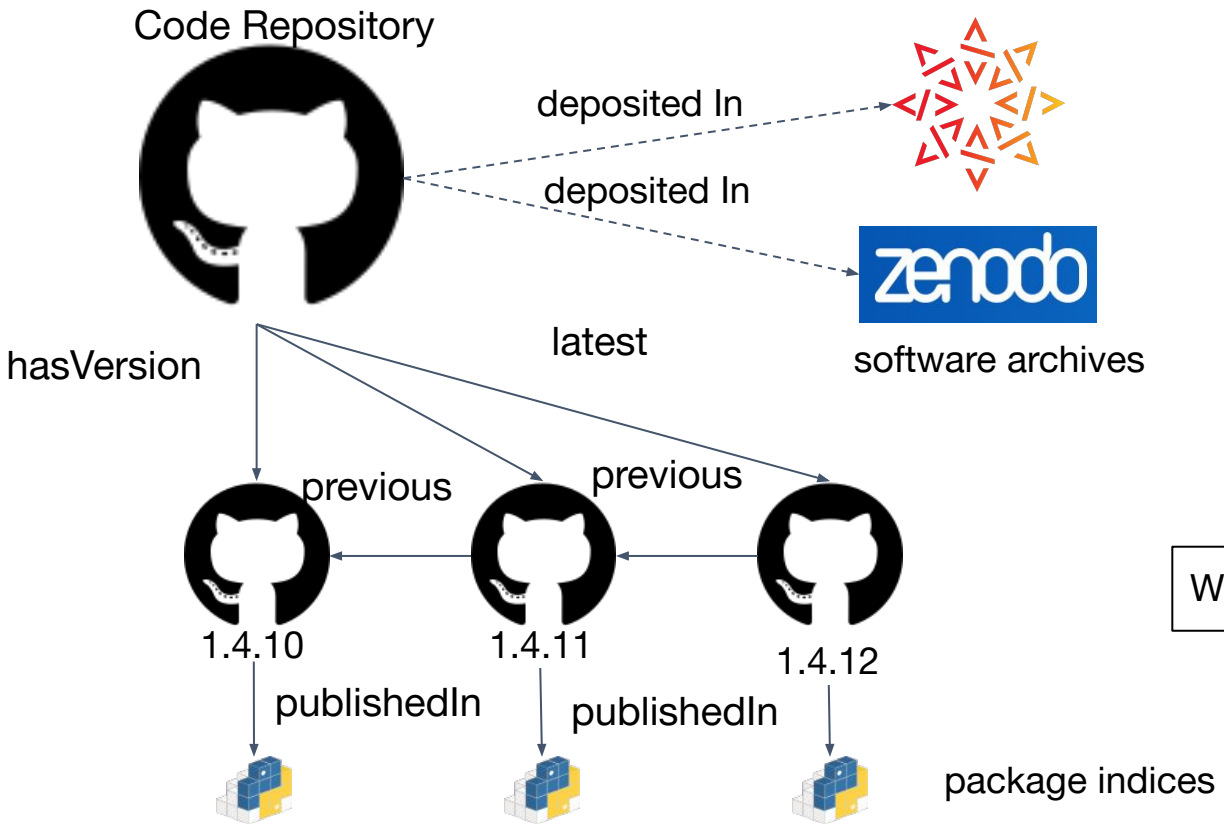



ORCIDs not available

CodeMeta Schema.org

https://w3id.org/okn/o/sd
https://w3id.org/software-types/

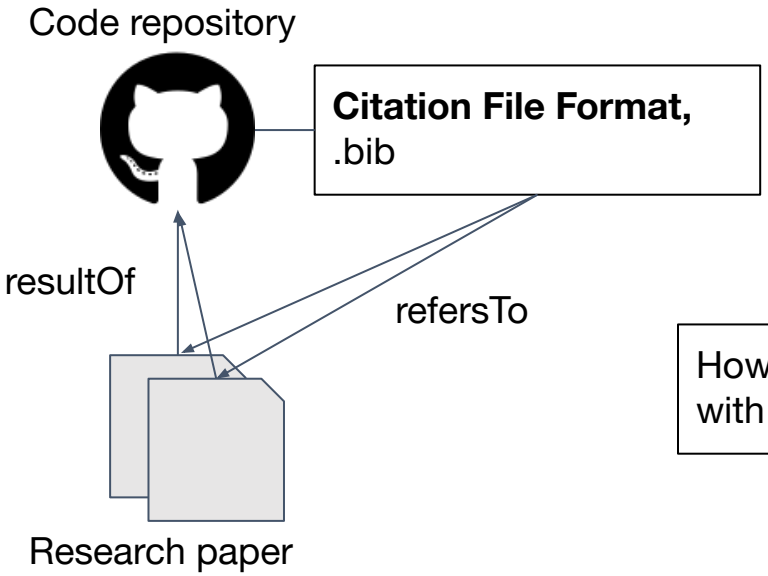
# Software and versions




Why is it needed?
- Search
- Compare
- Reuse

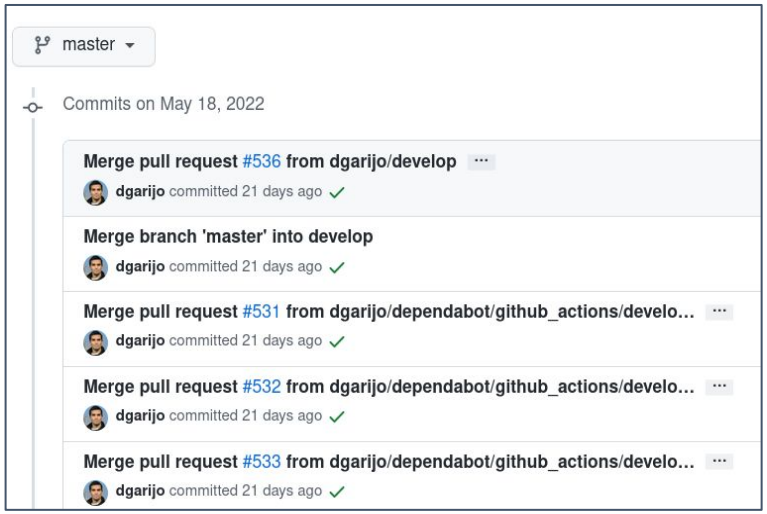
Which identifiers to use?

Why is it needed?
- Credit
- Trust




How to reconcile code authors with paper authors?

# Provenance and code contributions

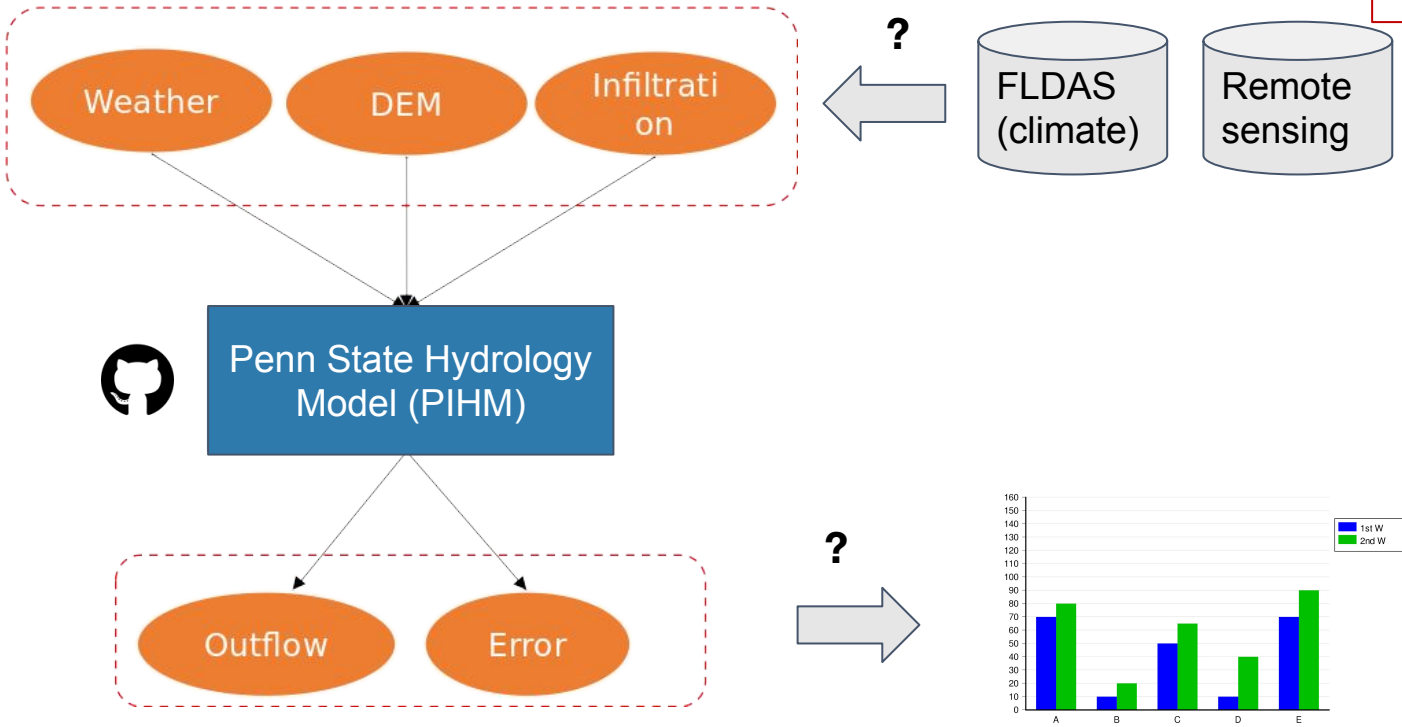



Why is it needed?
- Credit
- Responsibility
- Accountability

Contributions?
Development process?

## Describing inputs, outputs and their structure

Why is it needed?
- Reuse
- Reproducibility



# Dependencies

```
install_requires = [
    "bs4==0.0.1",
    "Click==7.0",
    "click-option-group==0.5.3",
    "markdown==3.3.6",
    "matplotlib==3.5.0",
    "nltk==3.6.6",
    "numpy==1.22.0",
    "pandas==1.3.4",
    "rdflib==6.0.2",
    "rdflib-jsonld==0.6.2",
    "requests>=2.22.0",
    "scikit-learn==1.0",
    "textblob==0.17.1",
    "validators==0.18.2",
    "xgboost==1.5.0"
]
```

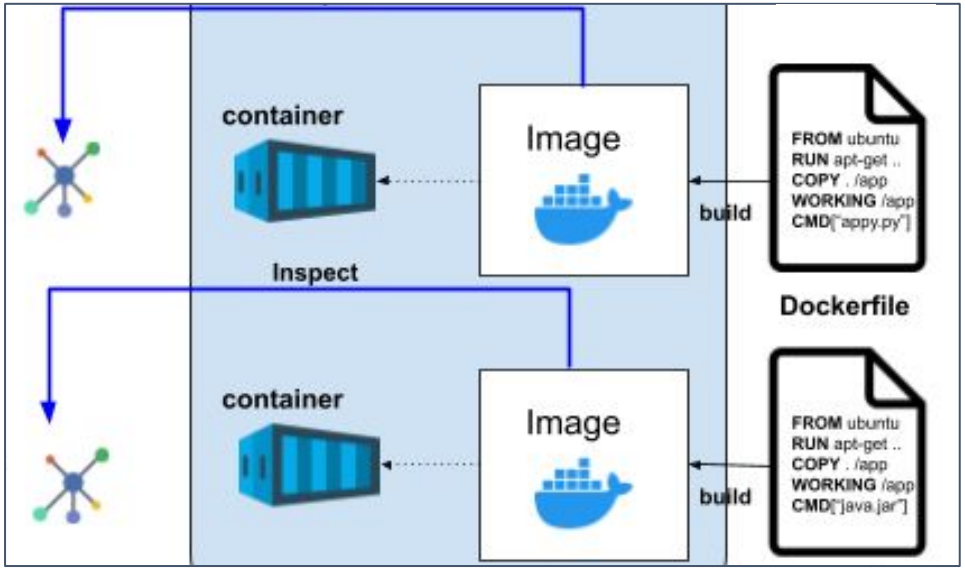
...

...

Why is it needed?

- Reuse
- Security

Software images are created from configuration files (e.g., Dockerfiles)




Why is it needed?
- Reuse
- Security

Fig. by Jhon Toledo

Initial effort transforming part of DockerHub: https://dockerpedia.inf.utfsm.cl/

Osorio, M., Buil-Aranda, C., Santana-Perez, I., & Garijo, D. (2022). DockerPedia: A Knowledge Graph of Software Images and Their Metadata. *International Journal of Software Engineering and Knowledge Engineering*, *32*(01), 71-89.

# **Creating KGs of Research Software metadata**:
Knowledge extraction

# Research Software metadata is not ~~abundant~~ machine readable

Can you please describe your software component with metadata?

I already did! Did you read the project readme?

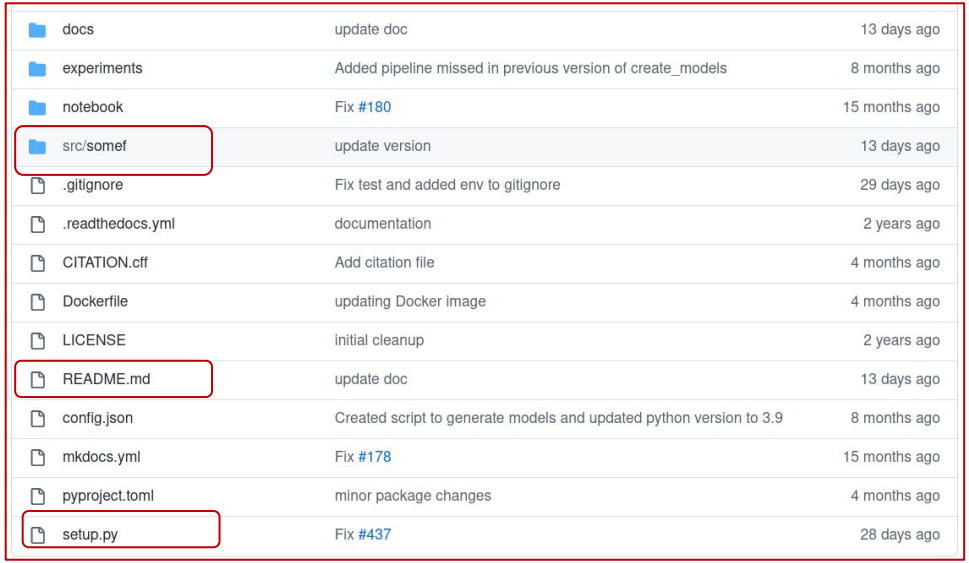
Did you see the online documentation?

Perhaps the you saw the paper?

Many domain-specific registries are curated by hand by experts

- Documentation
  - Text classification
  - Named entity recognition and relation extraction

- Code
  - Static code analysis

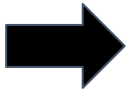


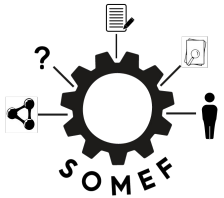
| | | |
|---|---|---|
| docs | update doc | 13 days ago |
| experiments | Added pipeline missed in previous version of create_models | 8 months ago |
| notebook | Fix #180 | 15 months ago |
| src/somef | update version | 13 days ago |
| .gitignore | Fix test and added env to gitignore | 29 days ago |
| .readthedocs.yml | documentation | 2 years ago |
| CITATION.cff | Add citation file | 4 months ago |
| Dockerfile | updating Docker image | 4 months ago |
| LICENSE | initial cleanup | 2 years ago |
| README.md | update doc | 13 days ago |
| config.json | Created script to generate models and updated python version to 3.9 | 8 months ago |
| mkdocs.yml | Fix #178 | 15 months ago |
| pyproject.toml | minor package changes | 4 months ago |
| setup.py | Fix #437 | 28 days ago |

# Text classification: Software Metadata Extraction Framework

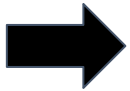
https://github.com/KnowledgeCaptureAndDiscovery/somef/

Repository

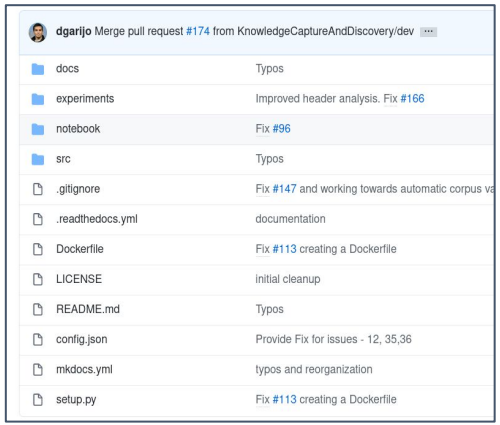





Extraction



Results (Metadata)




- **Readme Analysis**
  - Supervised classification
  - Regular expressions
  - Header analysis
- **File exploration**
  - Notebooks
  - Dockerfiles
  - Documentation
- **GitHub API**

CodeMeta



Kelley, A., & Garijo, D. (2021). A framework for creating knowledge graphs of scientific software metadata. *Quantitative Science Studies*, 1-37.

# SOMEF: Recognizing Metadata Categories

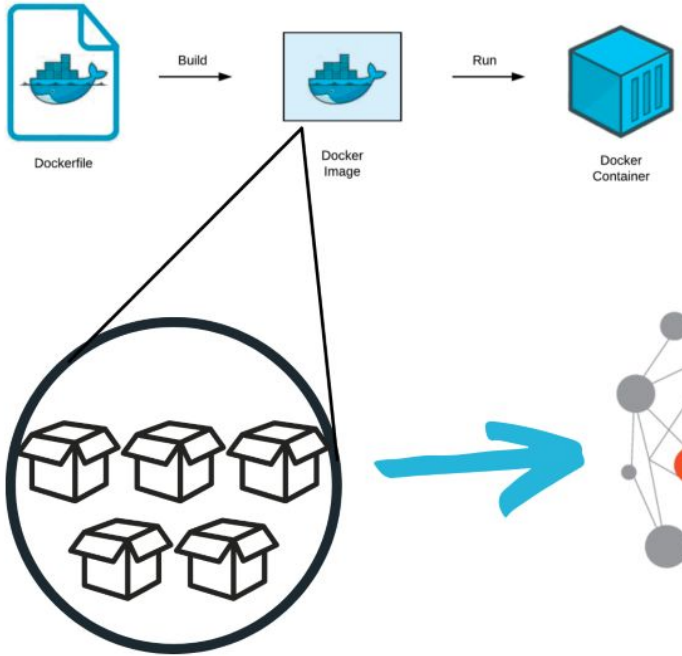
- **Name** (GA)
- **Full title** (RE)
- **Description** (SC, HA)
- **Citation** (SC, RE, HA)
- **Installation instructions** (SC, HA)
- **Invocation** (SC)
- **Usage examples** (HA)
- **Documentation** (HA, FE)
- **Requirements** (HA)
- **Contributors** (HA)
- **FAQ** (HA)
- **Support** (HA)
- **License** (GA, HA, FE)
- **Stars** (GA)

**Method used (provenance)**:

- Supervised Classification (SC)
- Header Analysis and Synset comparison (HA)
- File Exploration (FE)
- Regular Expressions (RE)
- GitHub API (GA)

- **Contact** (HA)
- **Download URL** (HA, GA)
- **DOI** (RE)
- **DockerFile** (FE)
- **Notebooks** (FE)
- **Executable notebooks (Binder, Collab)** (RE)
- **Owner**: (GA)
- **Keywords** (GA)
- **Source code** (GA)
- **Releases** (GA)
- **Changelog** (GA)
- **Issue tracker** (GA)
- **Programming languages** (GA)
- **Acknowledgements** (HA)
- **Logos** (RE)
- **Images** (RE)
- **Shell scripts** (FE)
- **Code of conduct** (FE)
- **Repository status** (RE)
- **Arxiv links** (RE)
- **Support channels** (RE)
- **Software category** (SC) **(Work in progress)**
- **…**

# Converting containers and code into KGs




C2T
CONTAINERS TO TRIPLES

docker RDF

Inspect4py

https://github.com/SoftwareUnderstanding/inspect4py

https://osoc-es.github.io/c2t/website/

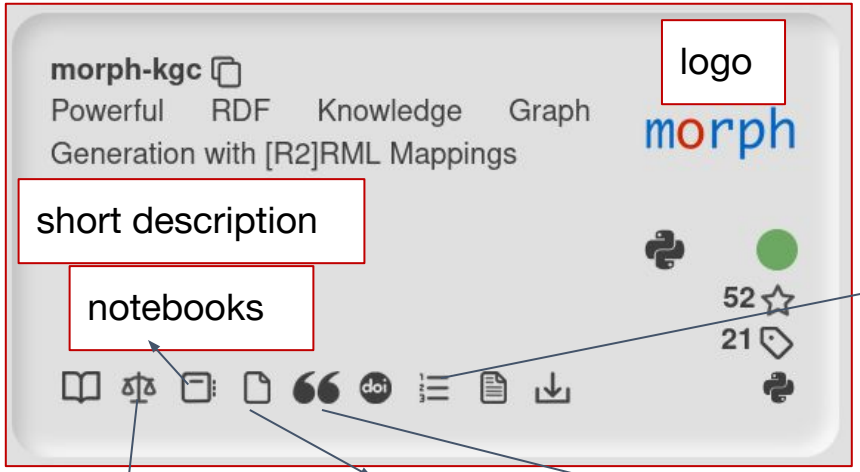
# **Creating KGs of Research Software metadata**:
Benefits

# Early result: Automated software catalogs


Useful for
- Comparison
- Exploring
- Reuse

Alpha available at: https://software.oeg.fi.upm.es/ Github: https://github.com/oeg-upm/soca

# A software repository at a glance




## License

**Apache License 2.0**

**Description:**

A permissive license whose main conditions require preservation of copyright and license notices. Contributors provide an express grant of patent rights. Licensed works, modifications, and larger works may be distributed under different terms and without source code.

**Permissions:**

1. Commercial-use
2. Modifications
3. Distribution
4. Patent-use
5. Private-use

{Morph-KGC: Scalable Knowledge Graph Materialization with Mapping Partitions}

## Citation

```
@article{arenas2022morph,
  title   = {{Morph-KGC: Scalable Knowledge Graph Materialization with
  author  = {Arenas-Guerrero, Julián and Chaves-Fraga, David and Toledo
  journal = {Semantic Web},
  year    = {2022},
  url     = {http://www.semantic-web-journal.net/system/files/swj3135.p
}
```

## Usage

Learn quickly with the tutorial in **Google Colaboratory**!

**PyPi** is the fastest way to install Morph-KGC:

```
pip install morph-kgc
```

We recommend to use **virtual environments** to install Morph-KGC.

To run the engine via **command line** you just need to execute the following:

```
python3 -m morph_kgc config.ini
```

Check the **documentation** to can see how to generate the configuration **INI file**.

**Here** you can also see an example INI file.

It is also possible to run Morph-KGC as a **library** with **RDFLib** and **Oxigraph**:

```
import morph_kgc

# generate the triples and load them to an RDFLib graph
g_rdflib = morph_kgc.materialize('/path/to/config.ini')
# work with the RDFLib graph
q_res = g_rdflib.query(' SELECT DISTINCT ?classes WHERE { ?s a ?classes } ')

# generate the triples and load them to Oxigraph
g_oxigraph = morph_kgc.materialize_oxigraph('/path/to/config.ini')
# work with Oxigraph
q_res = graph.query(' SELECT DISTINCT ?classes WHERE { ?s a ?classes } ')

# the methods above also accept the config as a string
config = """
        [DataSource1]
        mappings: /path/to/mapping/mapping_file.rml.ttl
        db_url: mysql+pymysql://user:password@localhos
    """
g_rdflib = morph_kgc.materialize(config)
```

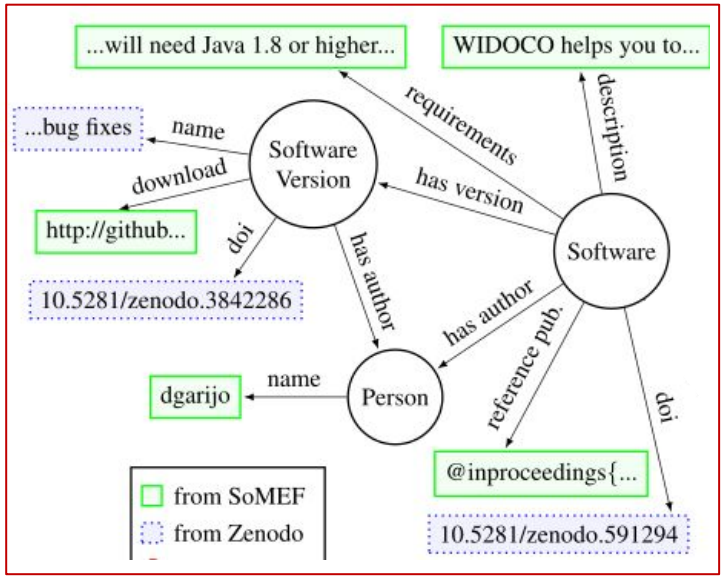
**How to use it**

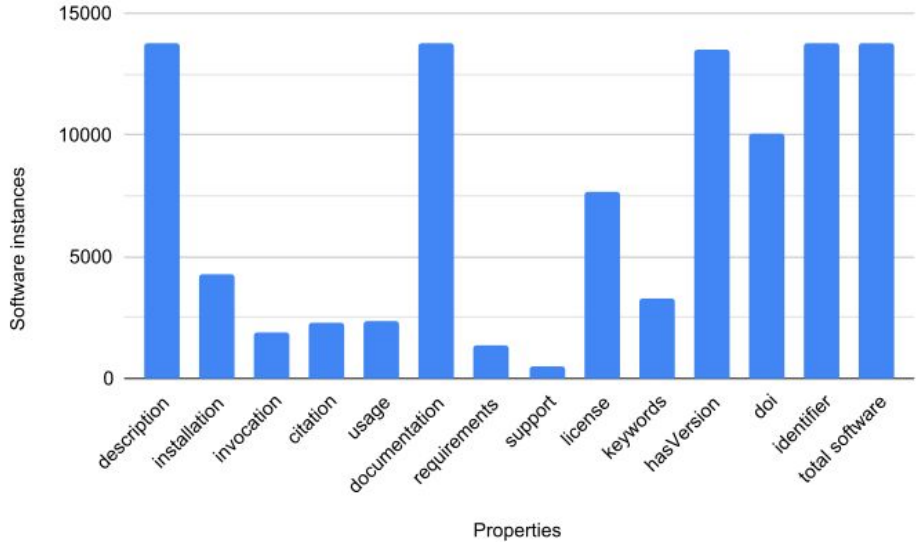
```
python /morph-kgc/oeg-upm_morph-kgc/morph-kgc-main/src/morph_kgc/main.py
```

invocation

Extracting KGs from thousands of Open Source repositories

- Zenodo software (> 12000)
- Measuring best practices based on metadata



# Summing up

Research software is a critical asset for **Open Science**

- Access information in structured, homogeneous manner
- Reusability
- Comparison
- Search



Pending Challenges**:**

- Automated metadata extraction from existing sources
- Curation
- Reconciliation of entities (through KGs like Wikidata)
- Representing all levels of granularity

# Acknowledgements





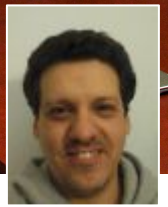






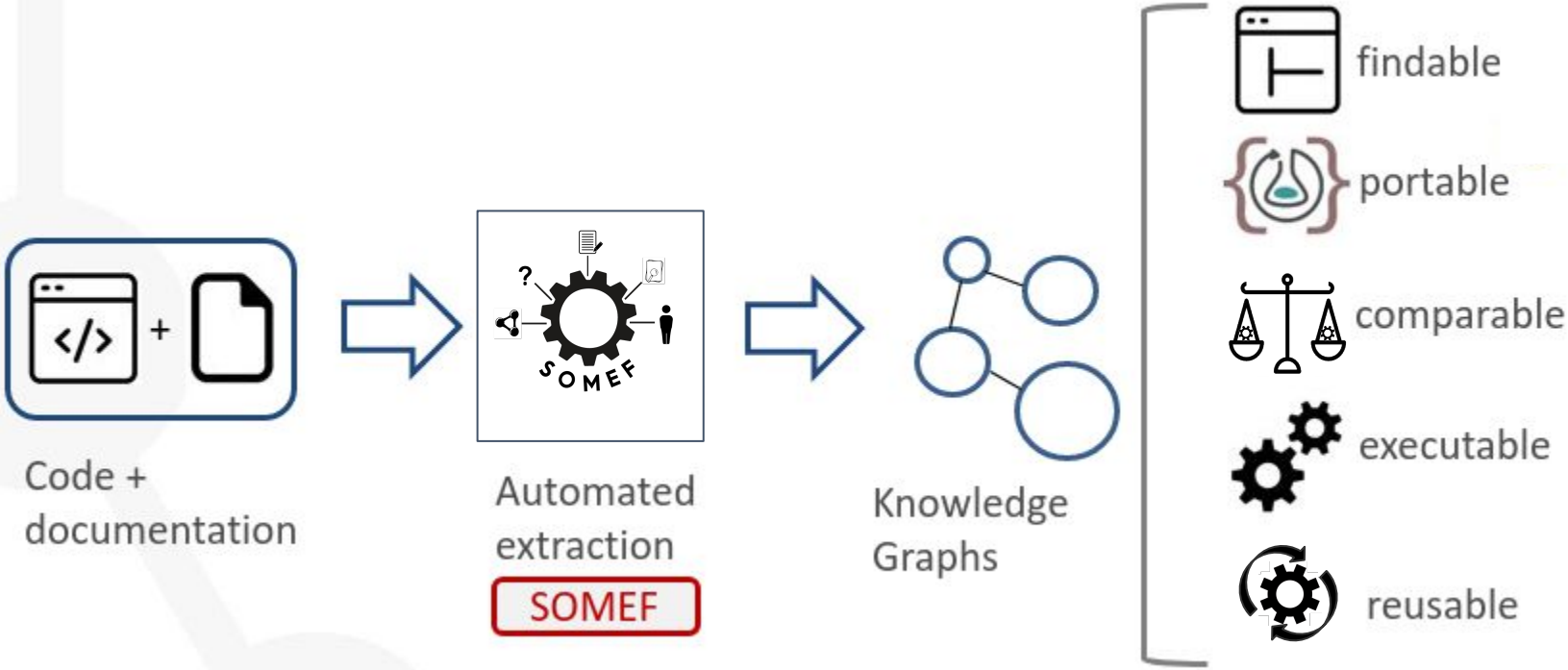





Thanks to Yolanda Gil, Varun Ratnakar, Maximiliano Osorio, Hernán Vargas, Deborah Khider, Allen Mao, Aidan Kelley, Haripriya Dharmala, Jiajing Wang, Rosa Filgueira, Pablo Calleja, Oscar Corcho, Laura Camacho, Jhon Toledo, Miguel Angel García, Esteban Gonzalez, Elena Montiel, Elvira Amador & all the students at UPM and USC who participated in the initiatives mentioned in this presentation

**This work has been supported by the Madrid Government (Comunidad de Madrid-Spain) under the Multiannual Agreement with Universidad Politécnica de Madrid in the line Support for R&D projects for Beatriz Galindo researchers, in the context of the V PRICIT (Regional Programme of Research and Technological Innovation)**




Let's create **machine-actionable** software metadata to promote Open Science!