27 January 2021

PIDs for software - shedding some light on a dark puzzle

Daniel S. Katz

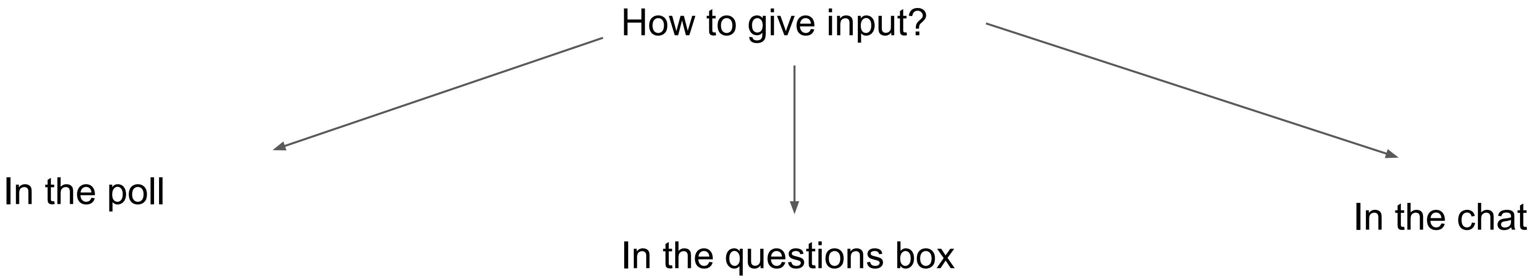
Chief Scientist, NCSA,
University of Illinois at
Urbana-Champaign

Morane Gruenpeter

Software Engineer
Metadata Specialist
Inria, Software Heritage

This session's goals

1. **Share findings** from the Software Source Code Identification Working Group
   DOI: 10.15497/RDA00053
2. **Learn** about attendees' backgrounds
3. **Show the complex puzzle** of identifying software
4. Get attendee opinions about **where we are**
5. Get attendee opinions about **where we should go**



How to give input?

In the poll

In the questions box

In the chat

PIDapalooza

FORCE11 The Future of Research Communications and e-Scholarship

RDA RESEARCH DATA ALLIANCE

Question 1: Are you **dealing** with software in your daily work?

- Yes, using,
- Yes, implementing
- Yes, developing
- Yes, designing
- Yes, testing
- Yes, all of the above
- No or Not yet

Software is all around us...

Apollo 11 Guidance Computer (~60.000 lines), 1969

"When I first got into it, nobody knew what it was that we were doing. It was like the Wild West."

Margaret Hamilton

Tim Berners-Lee invented the **World Wide Web, 1989**, while working at CERN

"When somebody has learned how to program a computer … **You're joining a group of people who can do incredible things. They can make the computer do anything they can imagine."**

From An Insight, An Idea with Tim Berners-Lee at 27:27 (25 January 2013)

DOI: 10.15497/RDA00053

Question 2: Have you **heard about** or **read** the RDA/FORCE11 Software Source Code Identification output?

- yes! I even read and commented during the community review,
- yes only heard about it,
- not yet :)

PIDapalooza

FORCE11 The Future of Research Communications and e-Scholarship

RDA RESEARCH DATA ALLIANCE

Software Source Code Identification Working Group

The SCID WG Goal: **capture and analyze** the software identification state-of-the-art in the scholarly ecosystem

Co-chairs

- Roberto Di Cosmo
- Martin Fenner
- Daniel S. Katz

Secretariat Liaison

- Stefanie Kethers

RDA page
https://www.rd-alliance.org/groups/software-source-code-identification-wg

Repository
https://github.com/force11/force11-rda-scidwg

Chronology...

03/2018 Spawned at RDA **P11** in Berlin from the

- RDA Software Source Code IG &
- FORCE11 Software Citation Implementation WG

10/2018 - TAB endorsement

4/2019 - RDA **P13**, Philadelphia

- **WG kick-off**

10/2019 - **FORCE2019**, Edinburgh Full day hackathon on research software

03/2020 - RDA VP15 session online

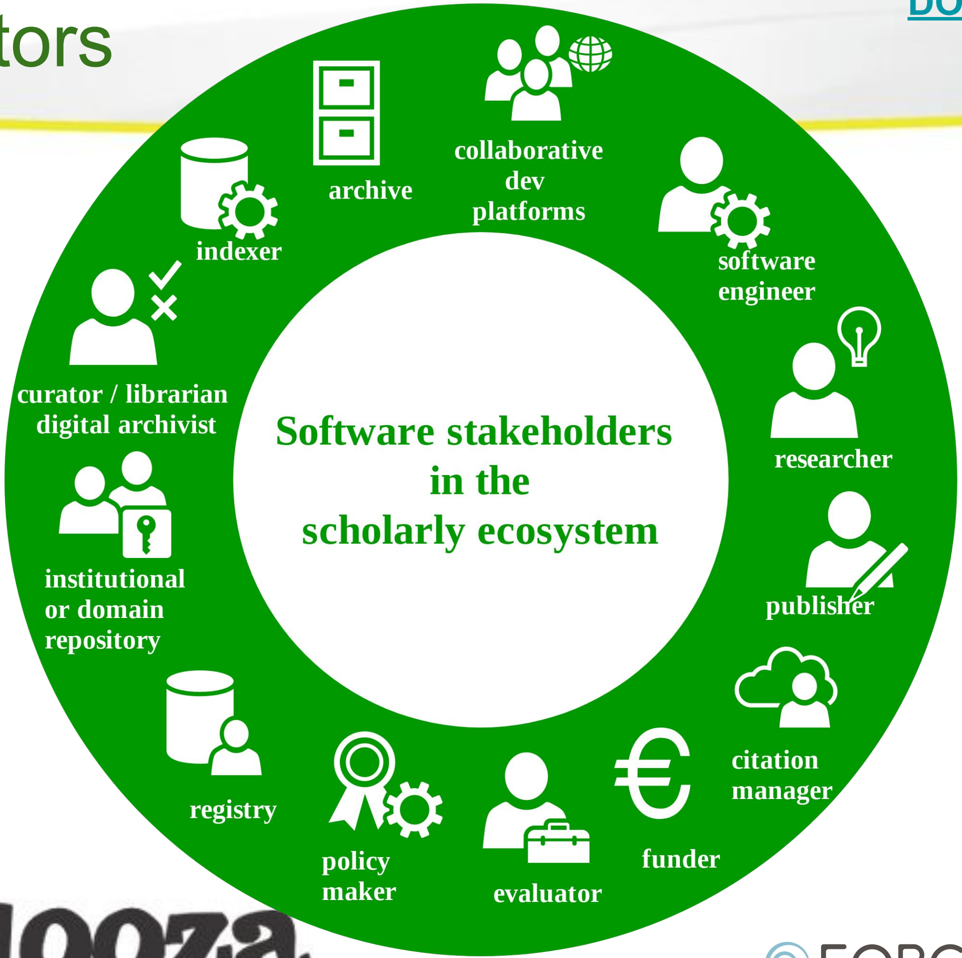
07/2020 - Output in community review **DOI:10.15497/RDA00053**

Question 3: What kind of **stakeholder** are you in the scholarly ecosystem?

- a researcher, a research software engineer,
- a publisher, a citation manager,
- a funder, an evaluator,
- a registry admin, a curator
- or some other role

DOI: 10.15497/RDA00053

Definition: Actors


Software stakeholders in the scholarly ecosystem
archive
collaborative dev platforms
software engineer
researcher
publisher
citation manager
funder
evaluator
policy maker
registry
institutional or domain repository
curator / librarian digital archivist
indexer

Question 4: Do you **need to identify** software in your work?

- Yes, in my articles, reports and other writings
- Yes, in my data
- Yes, in my software
- Yes, in different settings
- Not at the moment

Question 5: What do you want to see when you click on a software PID?

- a project, a module,
- a release, a source code artifact, an executable,
- a docs page, a tutorial,
- a docker image
- or depending on the use-case at hand

DOI: 10.15497/RDA00053

Identification target - what do we want to identify?

Software concept / project / collection

Description in registry, a homepage or any other form of metadata record

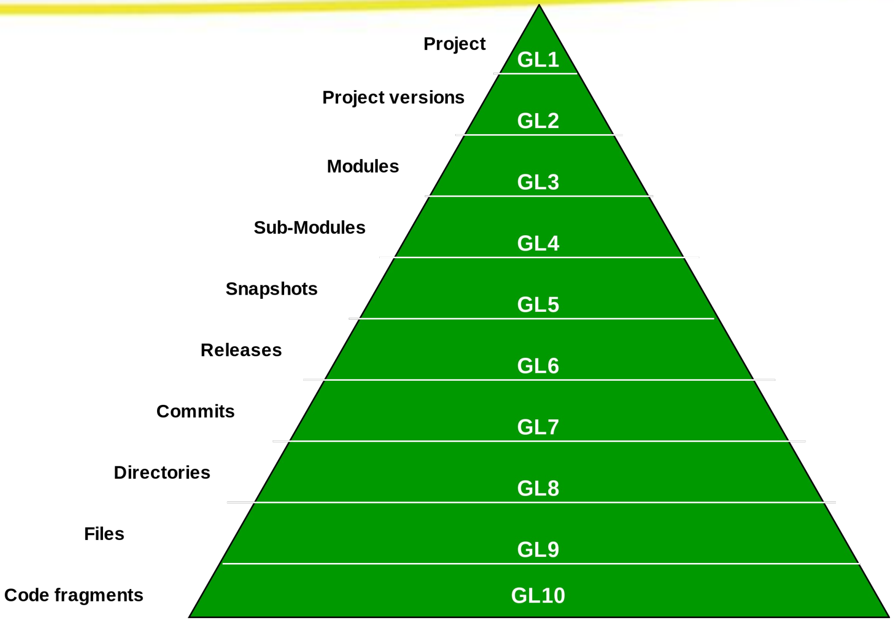
- Project versions (for example Python2 and Python3)
- Modules
- Sub-modules

Software artifact

- Executable (download link)
- Software source code
    - Dynamic artifact - current development code (on collaborative development platform)
    - Archived copy
        - Snapshot (all branches, all dev history)
        - Release / Package
        - Commit- a specific point in development history
        - Directory
        - File
        - Algorithm

Software context

- Complementary artifacts
    - the software environment, tutorial (Jupyter notebook), Data (input/output data), etc.
- Articles
- Documentation



Project — GL1
Project versions — GL2
Modules — GL3
Sub-Modules — GL4
Snapshots — GL5
Releases — GL6
Commits — GL7
Directories — GL8
Files — GL9
Code fragments — GL10

GL= Granularity Level

PIDapalooza

FORCE11
The Future of Research Communications and e-Scholarship

RDA
RESEARCH DATA ALLIANCE

Question 6: What’s the most important **use case** in your workflow?

- Get credit / Give credit
- To reproduce an experiment
- Curate software metadata
- Access the software source code to use it
- None of the above

DOI: 10.15497/RDA00053

The use cases collection (a small excerpt)

Actor	Use case description	Action	Identification target
Archive	Identify all the software artifacts I hold	Archiving, referencing	Release and smaller artifacts
Citation manager	Curate the software citation entries	Credit	Project, release
Curator / librarian / digital archivist	Catalog and browse the development history of legacy software source code for preservation purposes	Archiving	Project, release and smaller artifacts depending on the reference
Publisher	Create/retrieve identifiers quickly for use in the paper for all software including commercial packages.	Referencing, describing	Any item (all granularity levels)
Registry	Identify and curate the software entries I hold	Archiving, referencing, describing, credit	Project
Researcher as a software user (RSU)	Access and use SSC no longer available on a collaborative platform	Archiving	Snapshot, release, revision, directory

Question 7: Are you using a **software identifier**?

- Yes, all the time
- Yes, sometimes
- Maybe, I’m not sure
- No, it is too difficult to do

DOI: 10.15497/RDA00053

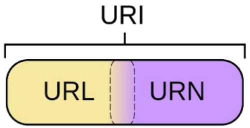
Landscape of Identifiers schemes (small excerpt)

ARCHIVES HAL OUVERTES

HAL-ID

doi

Digital Object Identifier



URI

URL

URN

WIKIDATA

Wiki Item identifier (Qxxx)

ASCL.net
Astrophysics Source Code Library

ARK
Archival Resource Key

Software Identification

RRID

Handle
Handle System identifiers

git

swMATH
an information service for mathematical software

SWHID
Software Heritage identifiers

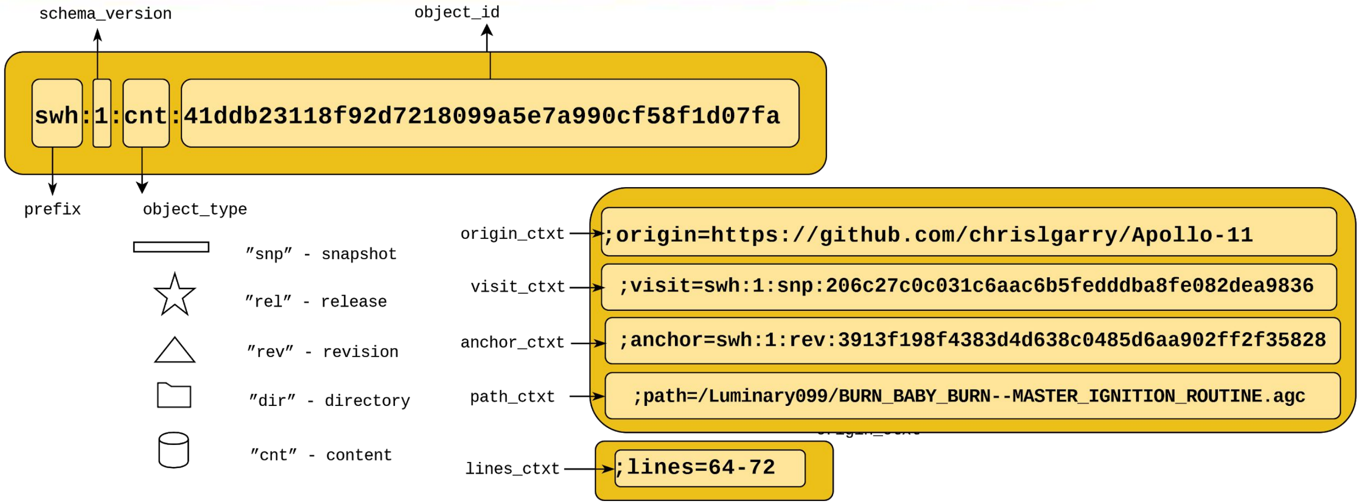
PIDapalooza

FORCE11
The Future of Research Communications and e-Scholarship

RDA
RESEARCH DATA ALLIANCE

Intrinsic identifier: the Software Heritage ID (SWHID)

- **Intrinsic**: compute a unique **digital fingerprint**
- **cryptographically strong** identifiers
- **decentralised**:
  - do not need a registry,
  - agreement on a standard



schema_version

object_id

swh:1:cnt:41ddb23118f92d7218099a5e7a990cf58f1d07fa

prefix

object_type

"snp" - snapshot

"rel" - release

"rev" - revision

"dir" - directory

"cnt" - content

origin_ctxt ;origin=https://github.com/chrislgarry/Apollo-11

visit_ctxt ;visit=swh:1:snp:206c27c0c031c6aac6b5fedddba8fe082dea9836

anchor_ctxt ;anchor=swh:1:rev:3913f198f4383d4d638c0485d6aa902ff2f35828

path_ctxt ;path=/Luminary099/BURN_BABY_BURN--MASTER_IGNITION_ROUTINE.agc

lines_ctxt ;lines=64-72

Documentation

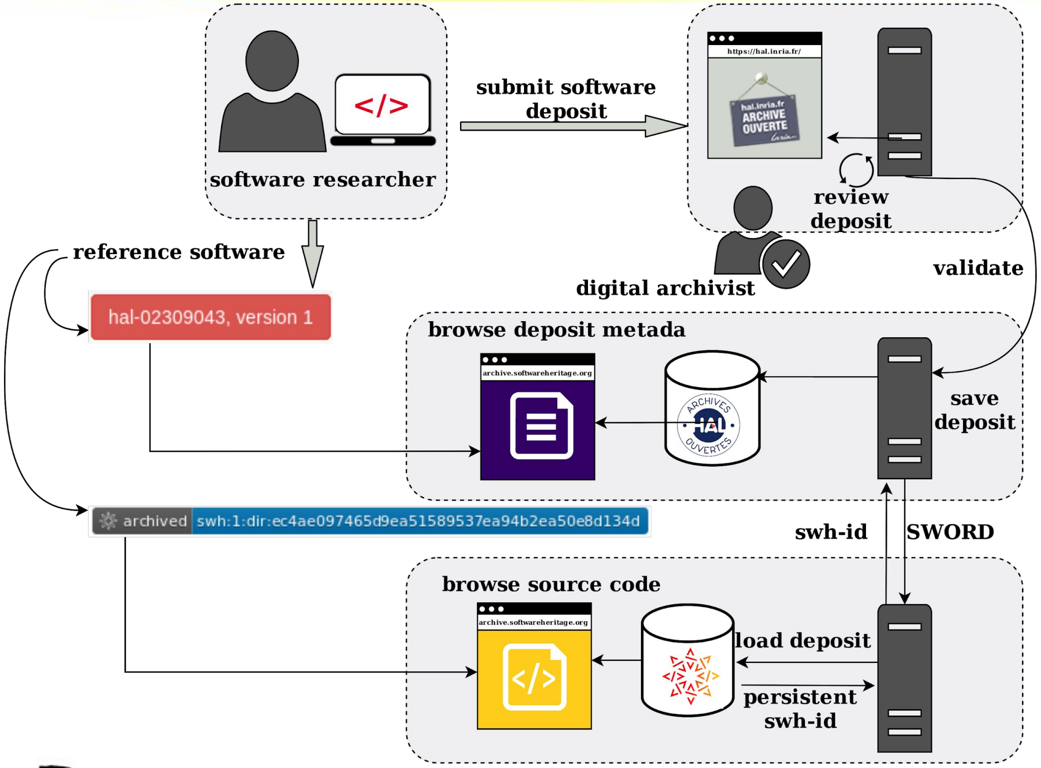
DOI: 10.15497/RDA00053

Extrinsic identifier: the **HAL ID**

Deposit guide

[Describe][Cite]

[Archive][Reference]



software researcher

submit software deposit

https://hal.inria.fr/

hal.inria.fr ARCHIVE OUVERTE

review deposit

digital archivist

validate

reference software

hal-02309043, version 1

browse deposit metada

archive.softwareheritage.org

save deposit

archived swh:1:dir:ec4ae097465d9ea51589537ea94b2ea50e8d134d

swh-id

SWORD

browse source code

archive.softwareheritage.org

load deposit

persistent swh-id

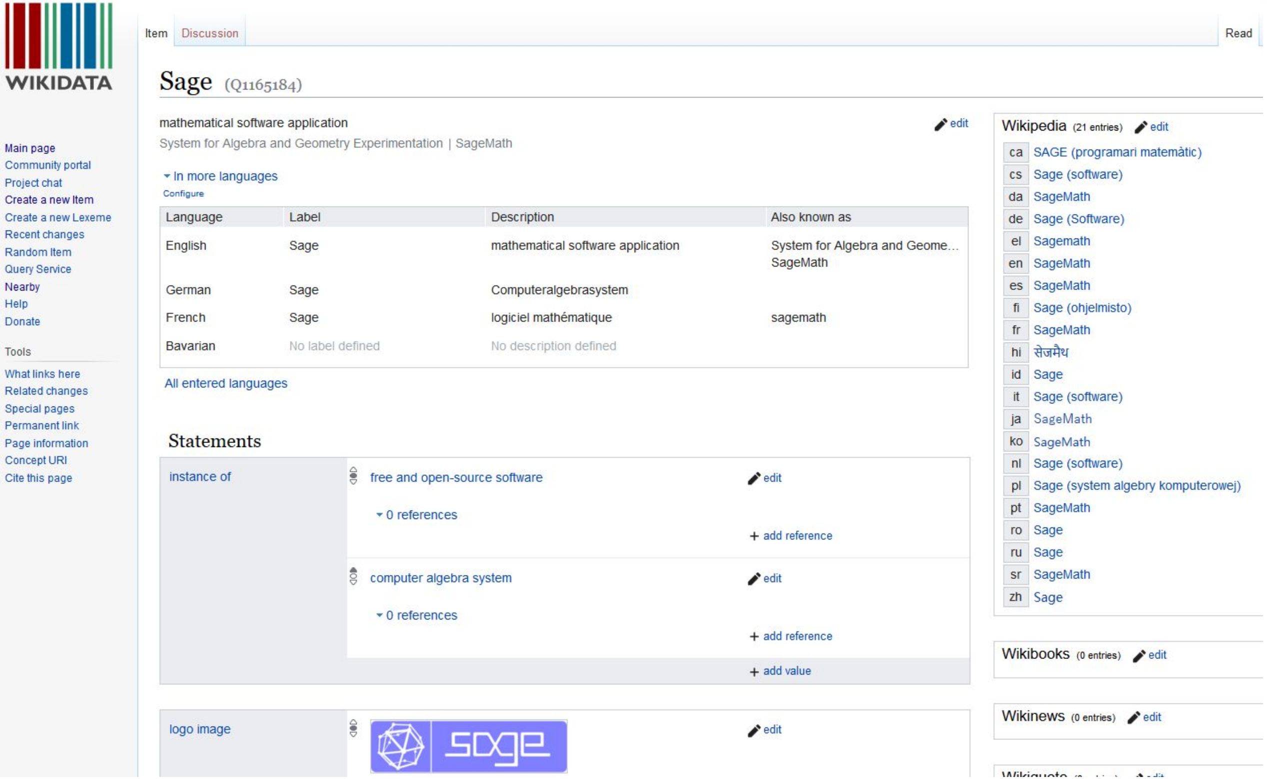
DOI: 10.15497/RDA00053

Wikidata entities for identifying projects

Q1165184=SageMath.

A few examples of external identifiers properties of used on software entities:

- Arch package sagemath
- Debian stable package
- Fedora package
- Free Software Directory entry
- Freebase
- Gentoo package
- Open Hub ID sage
- Quora topic
- Ubuntu Package
- swMATH work ID 825
- SWHID snapshot (15.11.2020)
- and many more



WIKIDATA

Main page
Community portal
Project chat
Create a new Item
Create a new Lexeme
Recent changes
Random Item
Query Service
Nearby
Help
Donate

Tools

What links here
Related changes
Special pages
Permanent link
Page information
Concept URI
Cite this page

Item | Discussion | Read

Sage (Q1165184)

mathematical software application ✎ edit
System for Algebra and Geometry Experimentation | SageMath

▾ In more languages
Configure

Language	Label	Description	Also known as
English	Sage	mathematical software application	System for Algebra and Geome… SageMath
German	Sage	Computeralgebrasystem	
French	Sage	logiciel mathématique	sagemath
Bavarian	No label defined	No description defined	

All entered languages

Statements

instance of
- free and open-source software ✎ edit
  ▾ 0 references
  + add reference
- computer algebra system ✎ edit
  ▾ 0 references
  + add reference

+ add value

logo image ✎ edit

Wikipedia (21 entries) ✎ edit

ca	SAGE (programari matemàtic)
cs	Sage (software)
da	SageMath
de	Sage (Software)
el	Sagemath
en	SageMath
es	SageMath
fi	Sage (ohjelmisto)
fr	SageMath
hi	सेजमैथ
id	Sage
it	Sage (software)
ja	SageMath
ko	SageMath
nl	Sage (software)
pl	Sage (system algebry komputerowej)
pt	SageMath
ro	Sage
ru	Sage
sr	SageMath
zh	Sage

Wikibooks (0 entries) ✎ edit

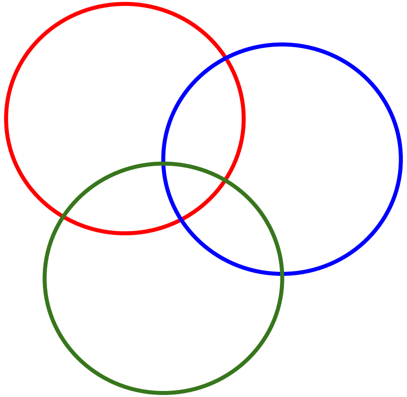
Wikinews (0 entries) ✎ edit

The Future of Research Communications and e-Scholarship

DOI: 10.15497/RDA00053

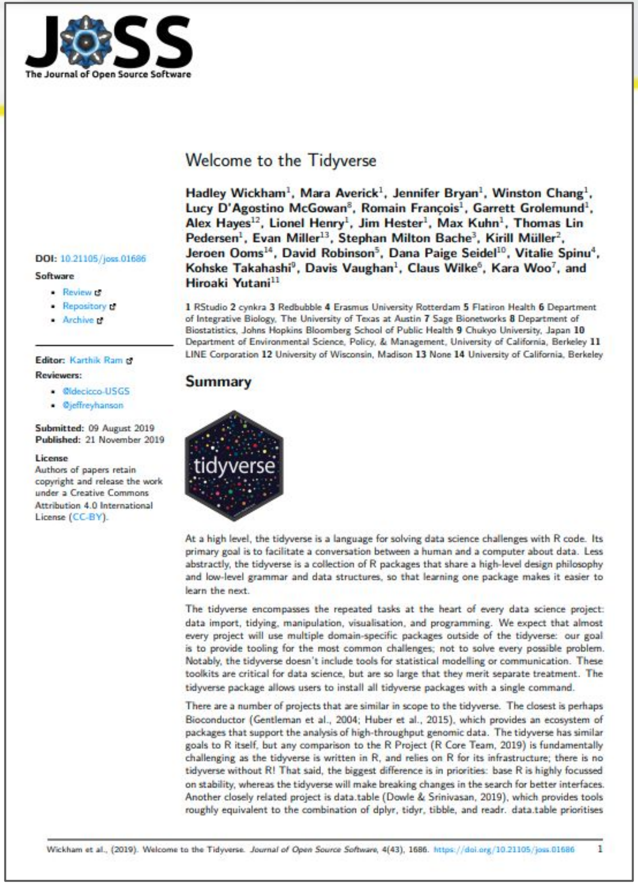
Identifiers from Registries

- To meet discipline-specific needs, discipline-specific registries have appeared
- **RRID** (life sciences), **ASCL** (astronomy), **swMath** (math)
- These let the community register different types of objects (including software) with an organization that curates the entries
- The entries then can be used in papers, including in citations
- But each is independent, and because software can cross disciplines, the registries can have overlapping contents



DOI on JOSS articles

- Journal of Open Source Software (JOSS, https://joss.theoj.org)
  - A developer friendly, open access journal for research software packages
- Research software packages are identified with:
  - The article DOI: JOSS publishes a short paper with a Crossref DOI (https://doi.org/10.21105/joss.01686)
  - The software DOI: The author deposits the software in a repository that provides a DOI (e.g., Zenodo) (https://doi.org/10.5281/zenodo.3547813)
- As a result, there are 2 DOIs.
- Links from the paper and metadata DOI to:
  - the software deposit and its DOI,
  - the live version of the software (e.g., on GitHub)



JOSS
The Journal of Open Source Software

Welcome to the Tidyverse

Hadley Wickham[1], Mara Averick[1], Jennifer Bryan[1], Winston Chang[1], Lucy D'Agostino McGowan[8], Romain François[1], Garrett Grolemund[1], Alex Hayes[12], Lionel Henry[1], Jim Hester[1], Max Kuhn[1], Thomas Lin Pedersen[1], Evan Miller[13], Stephan Milton Bache[3], Kirill Müller[2], Jeroen Ooms[14], David Robinson[5], Dana Paige Seidel[10], Vitalie Spinu[4], Kohske Takahashi[9], Davis Vaughan[1], Claus Wilke[6], Kara Woo[7], and Hiroaki Yutani[11]

1 RStudio **2** cynkra **3** Redbubble **4** Erasmus University Rotterdam **5** Flatiron Health **6** Department of Integrative Biology, The University of Texas at Austin **7** Sage Bionetworks **8** Department of Biostatistics, Johns Hopkins Bloomberg School of Public Health **9** Chukyo University, Japan **10** Department of Environmental Science, Policy, & Management, University of California, Berkeley **11** LINE Corporation **12** University of Wisconsin, Madison **13** None **14** University of California, Berkeley

DOI: 10.21105/joss.01686

Software

- Review
- Repository
- Archive

Editor: Karthik Ram

Reviewers:

- @ldecicco-USGS
- @jeffreyhanson

Submitted: 09 August 2019
Published: 21 November 2019

License
Authors of papers retain copyright and release the work under a Creative Commons Attribution 4.0 International License (CC-BY).

Summary

tidyverse

At a high level, the tidyverse is a language for solving data science challenges with R code. Its primary goal is to facilitate a conversation between a human and a computer about data. Less abstractly, the tidyverse is a collection of R packages that share a high-level design philosophy and low-level grammar and data structures, so that learning one package makes it easier to learn the next.

The tidyverse encompasses the repeated tasks at the heart of every data science project: data import, tidying, manipulation, visualisation, and programming. We expect that almost every project will use multiple domain-specific packages outside of the tidyverse: our goal is to provide tooling for the most common challenges; not to solve every possible problem. Notably, the tidyverse doesn't include tools for statistical modelling or communication. These toolkits are critical for data science, but are so large that they merit separate treatment. The tidyverse package allows users to install all tidyverse packages with a single command.

There are a number of projects that are similar in scope to the tidyverse. The closest is perhaps Bioconductor (Gentleman et al., 2004; Huber et al., 2015), which provides an ecosystem of packages that support the analysis of high-throughput genomic data. The tidyverse has similar goals to R itself, but any comparison to the R Project (R Core Team, 2019) is fundamentally challenging as the tidyverse is written in R, and relies on R for its infrastructure; there is no tidyverse without R! That said, the biggest difference is in priorities: base R is highly focussed on stability, whereas the tidyverse will make breaking changes in the search for better interfaces. Another closely related project is data.table (Dowle & Srinivasan, 2019), which provides tools roughly equivalent to the combination of dplyr, tidyr, tibble, and readr. data.table prioritises

Wickham et al., (2019). Welcome to the Tidyverse. *Journal of Open Source Software*, 4(43), 1686. https://doi.org/10.21105/joss.01686 1

Summary

Granularity level (GL)	ID target	Extrinsic identifiers									Intrinsic identifiers	
		ASCL	*ARK*	*DOI*	*HAL*	*URL*	*RRID*	*SwMath*	*Wikidata*		*Hash*	*SWH*
									entity	*property*		
GL1	project	X	X	X	X	X	X	X	X			
GL2	project version		X						X			
GL3	module		X						X			
GL4	repository		X			X				X		
GL5	repository snapshot		X							X		X
GL6	release		X	X**						X	X	X
GL7	commit		X							X	X	X
GL8	directory		X	X**	X*					X	X	X
GL9	file		X	X**							X	X
GL10	Code fragment		X									X

PIDapalooza

FORCE11
The Future of Research Communications and e-Scholarship

RDA
RESEARCH DATA ALLIANCE

Final question (8): Given this mess (dark puzzle), how can we clean it up?

(considering communities, technologies, etc.)

PIDapalooza

FORCE11
The Future of Research Communications and e-Scholarship

RDA
RESEARCH DATA ALLIANCE