Implementing the RDA Data Citation Recommendations for Long Tail Research Data

Stefan Pröll

research data sharing without barriers
rd-alliance.org

Overview

- Introduction
- Recap of the WGDC Recommendations
- Long Tail Research Data
- SQL Prototype
- Git Prototype
- Conclusion

Data Driven Research

- Modern research is data driven
  - Results are based on data
  - But the results are still published in papers
- Data sets are often
  - Not available or accessible
  - Not cited
  - Ambiguous

→ Reproducibility is at risk

Data Citation

- Citing data may seem easy
  - from providing a URL in a footnote
  - via providing a reference in the bibliography section
  - to assigning a PID (DOI, ARK, …) to dataset in a repository
- What's the problem?

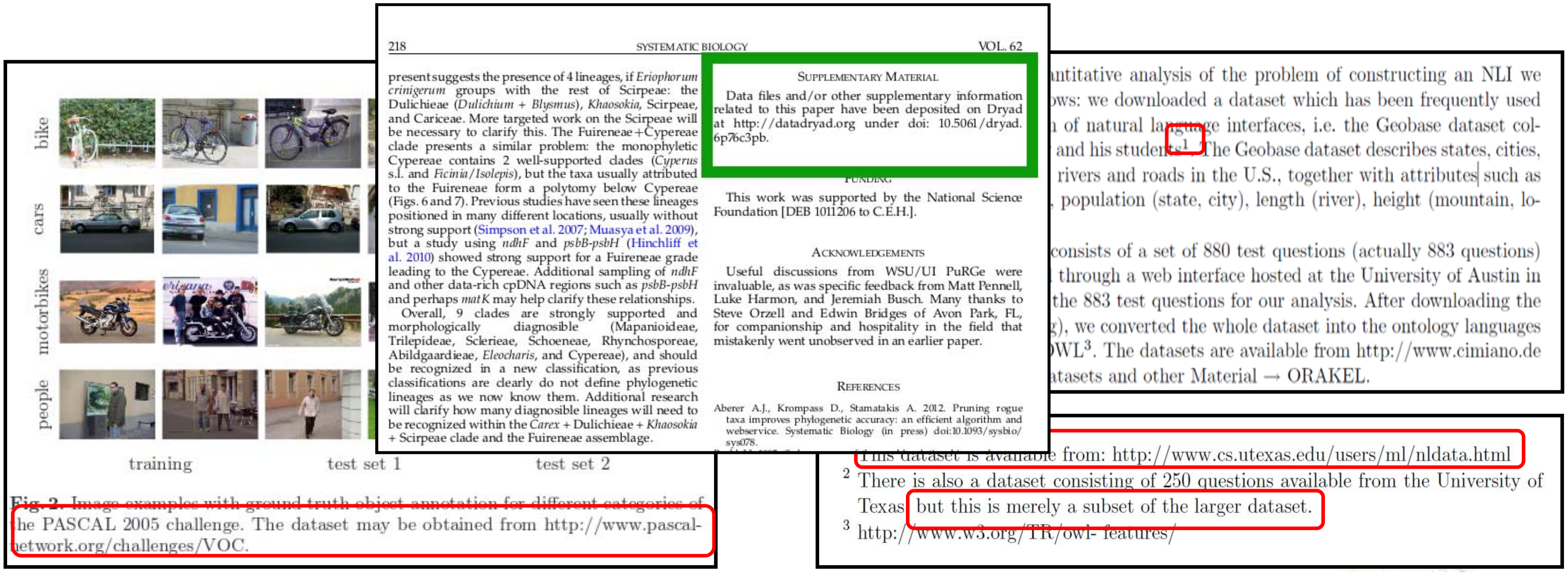


Fig. 2. Image examples with ground truth object annotation for different categories of the PASCAL 2005 challenge. The dataset may be obtained from http://www.pascal-network.org/challenges/VOC.

218 SYSTEMATIC BIOLOGY VOL. 62

present suggests the presence of 4 lineages, if *Eriophorum crinigerum* groups with the rest of Scirpeae: the Dulichieae (*Dulichium* + *Blysmus*), *Khaosokia*, Scirpeae, and Cariceae. More targeted work on the Scirpeae will be necessary to clarify this. The Fuireneae+Cypereae clade presents a similar problem: the monophyletic Cypereae contains 2 well-supported clades (*Cyperus* s.l. and *Ficinia*/*Isolepis*), but the taxa usually attributed to the Fuireneae form a polytomy below Cypereae (Figs. 6 and 7). Previous studies have seen these lineages positioned in many different locations, usually without strong support (Simpson et al. 2007; Muasya et al. 2009), but a study using *ndhF* and *psbB-psbH* (Hinchliff et al. 2010) showed strong support for a Fuireneae grade leading to the Cypereae. Additional sampling of *ndhF* and other data-rich cpDNA regions such as *psbB-psbH* and perhaps *matK* may help clarify these relationships.

Overall, 9 clades are strongly supported and morphologically diagnosible (Mapanioideae, Trilepideae, Sclerieae, Schoeneae, Rhynchosporeae, Abildgaardieae, *Eleocharis*, and Cypereae), and should be recognized in a new classification, as previous classifications are clearly do not define phylogenetic lineages as we now know them. Additional research will clarify how many diagnosible lineages will need to be recognized within the *Carex* + Dulichieae + *Khaosokia* + Scirpeae clade and the Fuireneae assemblage.

SUPPLEMENTARY MATERIAL

Data files and/or other supplementary information related to this paper have been deposited on Dryad at http://datadryad.org under doi: 10.5061/dryad.6p76c3pb.

FUNDING

This work was supported by the National Science Foundation [DEB 1011206 to C.E.H.].

ACKNOWLEDGEMENTS

Useful discussions from WSU/UI PuRGe were invaluable, as was specific feedback from Matt Pennell, Luke Harmon, and Jeremiah Busch. Many thanks to Steve Orzell and Edwin Bridges of Avon Park, FL, for companionship and hospitality in the field that mistakenly went unobserved in an earlier paper.

REFERENCES

Aberer A.J., Krompass D., Stamatakis A. 2012. Pruning rogue taxa improves phylogenetic accuracy: an efficient algorithm and webservice. Systematic Biology (in press) doi:10.1093/sysbio/sys078.

antitative analysis of the problem of constructing an NLI we ows: we downloaded a dataset which has been frequently used n of natural language interfaces, i.e. the Geobase dataset col- and his students[1]. The Geobase dataset describes states, cities, rivers and roads in the U.S., together with attributes such as , population (state, city), length (river), height (mountain, lo-

consists of a set of 880 test questions (actually 883 questions) through a web interface hosted at the University of Austin in the 883 test questions for our analysis. After downloading the g), we converted the whole dataset into the ontology languages OWL[3]. The datasets are available from http://www.cimiano.de atasets and other Material → ORAKEL.

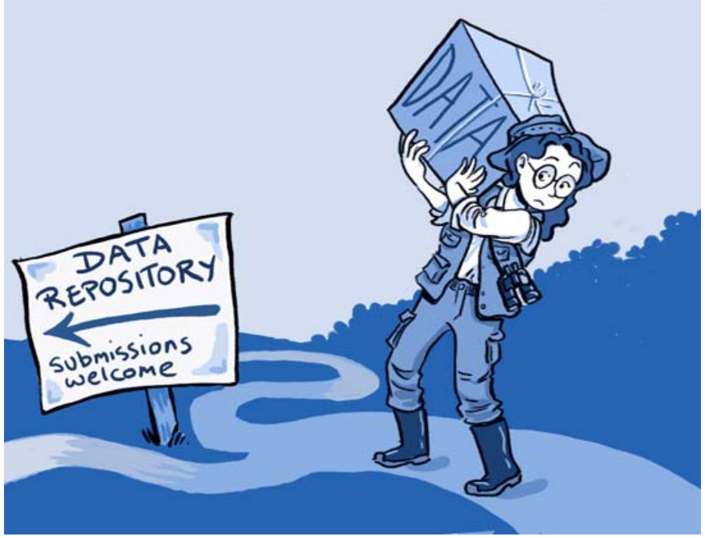
[1] This dataset is available from: http://www.cs.utexas.edu/users/ml/nldata.html

[2] There is also a dataset consisting of 250 questions available from the University of Texas but this is merely a subset of the larger dataset.

[3] http://www.w3.org/TR/owl- features/

Main Challenges

- Scalability
  - More and more data sets
  - Growing amounts of data
  - Granularity
- Infrastructure
  - Sophisticated data management is not always available
  - Processes not defined well
- Dynamics
  - Frequent updates
  - Evolving data
- Precise identification
  - Ambiguity?



Src: CC BY 4.0, https://commons.wikimedia.org/w/index.php?curid=30978545

Granularity of Subsets

- What about the **granularity** of data to be identified?
  - Enormous amounts of data
  - Researchers use specific subsets of data
  - Need to identify precisely the subset used
- Current approaches
  - Storing a copy of subset as used in study -> scalability
  - Citing entire dataset, providing textual description of subset
  - -> imprecise (ambiguity)
  - Storing list of record identifiers in subset -> scalability,
  - not for arbitrary subsets (e.g. when not entire record selected)

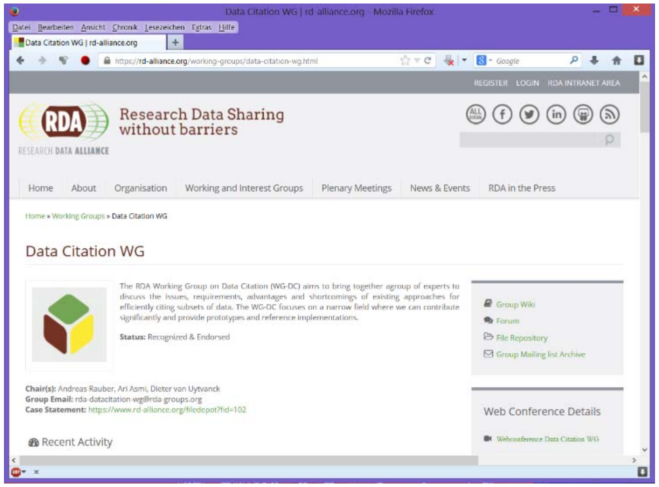
- Would like to be able to identify precisely the **subset of (dynamic) data used** in a process

Identification of Dynamic Data

- Citable datasets have to be static
- Fixed set of data, no changes:
  - no corrections to errors, no new data being added
- But: (research) data is **dynamic**
  - Adding new data, correcting errors, enhancing data quality, …
  - Changes sometimes highly dynamic, at irregular intervals
- Current approaches
  - Identifying entire data stream, without any versioning
  - Using “accessed at” date
  - “Artificial” versioning by identifying batches of data (e.g. annual), aggregating changes into releases (time-delayed!)
- Would like to identify precisely the **data as it existed at a specific point in time**

RDA WG Data Citation

- Research Data Alliance
- WG on **Data Citation:**
- **Making Dynamic Data Citeable**
- WG officially endorsed in March 2014
- Concentrating on the problems of
  - **large, dynamic (changing) datasets**
  - Focus! Identification of data!
  - Not: PID systems, metadata, citation string, attribution, …
  - Liaise with other WGs and initiatives on data citation
  - (CODATA, DataCite, Force11, …)


Data Citation WG | rd-alliance.org - Mozilla Firefox
Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe
Data Citation WG | rd-alliance.org
https://rd-alliance.org/working-groups/data-citation-wg.html
Google
REGISTER LOGIN RDA INTRANET AREA
RDA
RESEARCH DATA ALLIANCE
Research Data Sharing without barriers
Home About Organisation Working and Interest Groups Plenary Meetings News & Events RDA in the Press
Home » Working Groups » Data Citation WG

Data Citation WG

The RDA Working Group on Data Citation (WG-DC) aims to bring together agroup of experts to discuss the issues, requirements, advantages and shortcomings of existing approaches for efficiently citing subsets of data. The WG-DC focuses on a narrow field where we can contribute significantly and provide prototypes and reference implementations.

Status: Recognized & Endorsed

Chair(s): Andreas Rauber, Ari Asmi, Dieter van Uytvanck
Group Email: rda-datacitation-wg@rda-groups.org
Case Statement: https://www.rd-alliance.org/filedepot?fid=102

- https://rd-alliance.org/working-groups/data-citation-wg.html

Basic Principle

Idea: Versioned data + timestamped queries

- Data: timestamped and versioned (aka history)
- Query: Timestamped

- Access: Re-execute query on versioned data with the appropriate timestamp.

- **Trick: Assign the PID to the query**

S. Pröll, A. Rauber. **Scalable Data Citation in Dynamic Large Databases: Model and Reference Implementation.** In IEEE Intl. Conf. on Big Data 2013 (IEEE BigData2013), 2013
http://www.ifs.tuwien.ac.at/~andi/publications/pdf/pro_ieeebigdata13.pdf

Data Citation – Output

14 Recommendations

- Grouped into 4 phases:
  - Preparing data and query store
  - Persistently identifying specific data sets
  - Resolving PIDs
  - Upon modifications to the data infrastructure
- 2-page flyer
- More detailed Technical Report:
- https://rd-alliance.org/group/data-citation-wg/wiki/wgdc-recommendations.html
- Reference implementations
- (SQL, CSV, XML) and Pilots

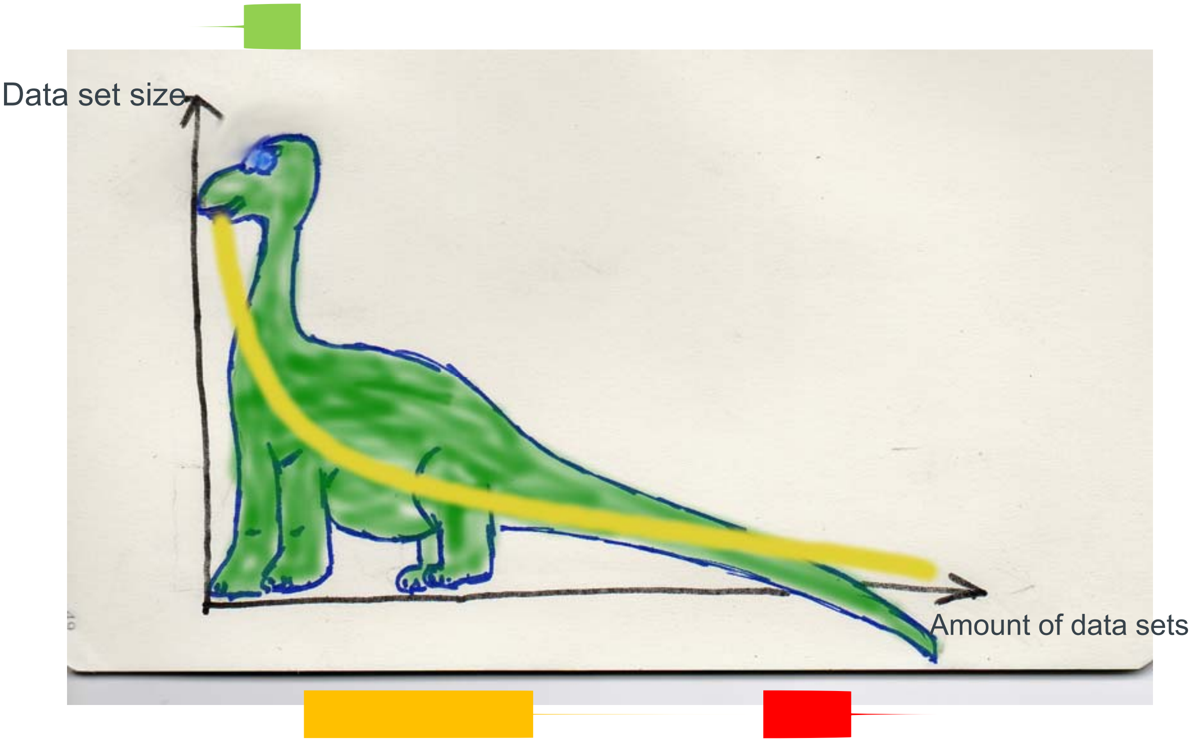
Data Citation of Evolving Data

Recommendations of the Working Group on Data Citation (WGDC)

Identification of Reproducible Subsets for Data Citation, Sharing and Re-Use

Long Tail Research Data

Big data,
well organized,
often used and cited



Less well organized,
non-standardised
no dedicated infrastructure

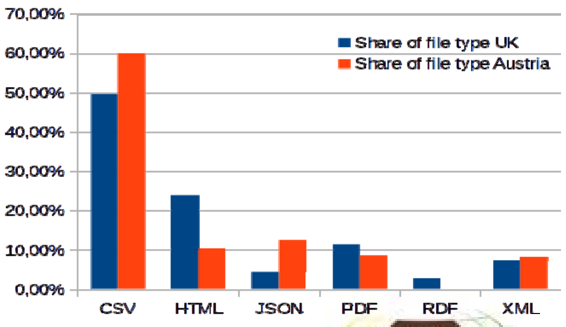
“Dark data”

[1] Heidorn, P. Bryan. "Shedding light on the dark data in the long tail of science." Library Trends 57.2 (2008): 280-299.

Dynamic Data Citation for CSV Data

- **Goals**:
  - Ensure cite-ability of CSV data
  - Enable subset citation
  - Support particularly small and large volume data
  - Support dynamically changing data
  - Establish links between data set and subsets
  - Scalable approach without storing copies of data exports

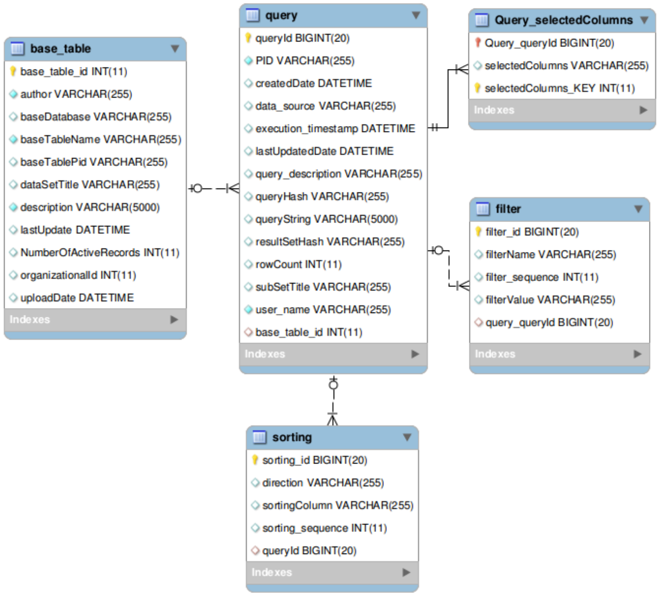
- **Why CSV data?**
  - Well understood and widely spread
  - Small and big data settings
  - Simple and flexible



Large Scale Research Settings

- Advanced data infrastructure
    - Large data sets
    - Database driven
    - Defined interfaces
    - Trained experts available
    -
- Required adaptions
    - Ingest CSV files
    - Capture subset process
    - Implement dedicated query store
    -

❼ SQL Prototype



base_table
base_table_id INT(11)
author VARCHAR(255)
baseDatabase VARCHAR(255)
baseTableName VARCHAR(255)
baseTablePid VARCHAR(255)
dataSetTitle VARCHAR(255)
description VARCHAR(5000)
lastUpdate DATETIME
NumberOfActiveRecords INT(11)
organizationalId INT(11)
uploadDate DATETIME
Indexes

query
queryId BIGINT(20)
PID VARCHAR(255)
createdDate DATETIME
data_source VARCHAR(255)
execution_timestamp DATETIME
lastUpdatedDate DATETIME
query_description VARCHAR(255)
queryHash VARCHAR(255)
queryString VARCHAR(5000)
resultSetHash VARCHAR(255)
rowCount INT(11)
subSetTitle VARCHAR(255)
user_name VARCHAR(255)
base_table_id INT(11)
Indexes

Query_selectedColumns
Query_queryId BIGINT(20)
selectedColumns VARCHAR(255)
selectedColumns_KEY INT(11)
Indexes

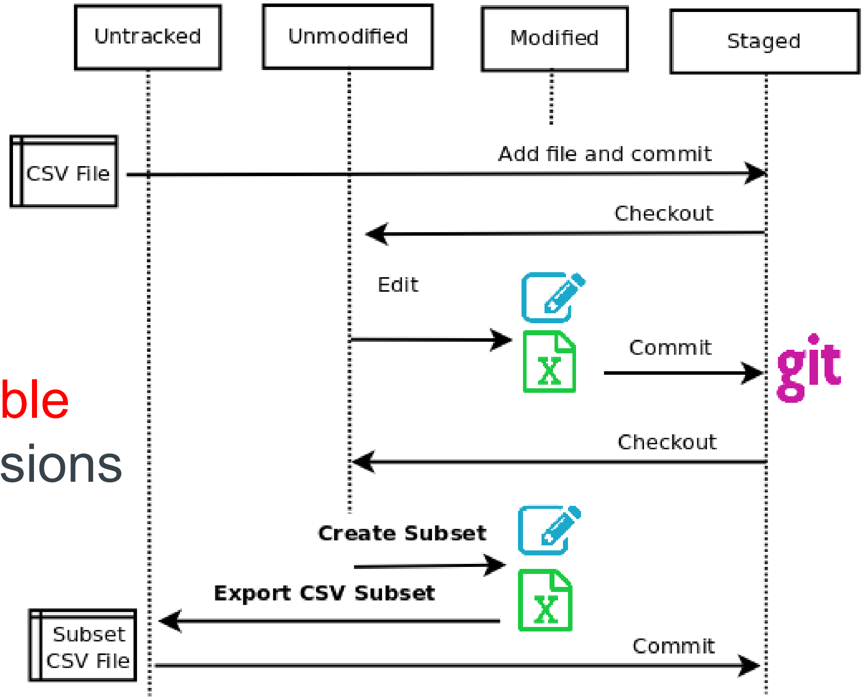
filter
filter_id BIGINT(20)
filterName VARCHAR(255)
filter_sequence INT(11)
filterValue VARCHAR(255)
query_queryId BIGINT(20)
Indexes

sorting
sorting_id BIGINT(20)
direction VARCHAR(255)
sortingColumn VARCHAR(255)
sorting_sequence INT(11)
queryId BIGINT(20)
Indexes

Small Scale Research Settings

- Local workstations
  - Smaller data sets
  - Local storage and tools
  - Scripting languages
- Required adaptions
  - Data versioning, e.g. with Git
  - Store scripts versioned as well
  - Make subset creation reproducible
  - Document software and OS versions
  - Share repositories

❼ **Git Prototype**



Prototype Implementations

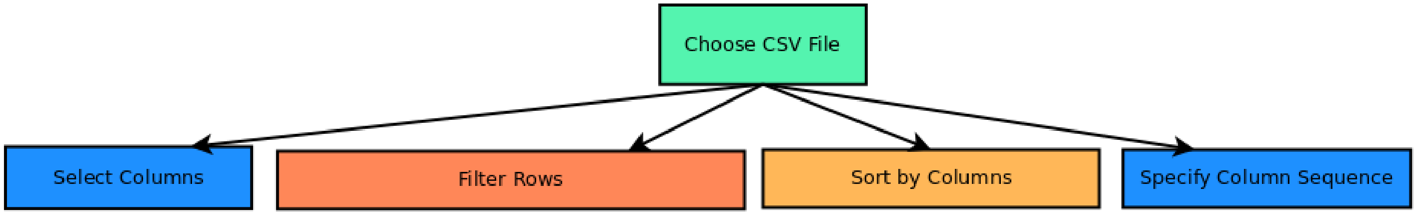
- SQL based Prototype
  - A) Migrates CSV data into relational database

- Git based Prototypes
  - A) Git as backend only
  - B) Using branches for data and scripts

- Data backend responsible for versioning data sets
- Subsets are created with scripts or queries

Reproducible Subsets with SQL

- CSV files have the same structure as relational database tables
- Subsetting process via SQL SELECT statements

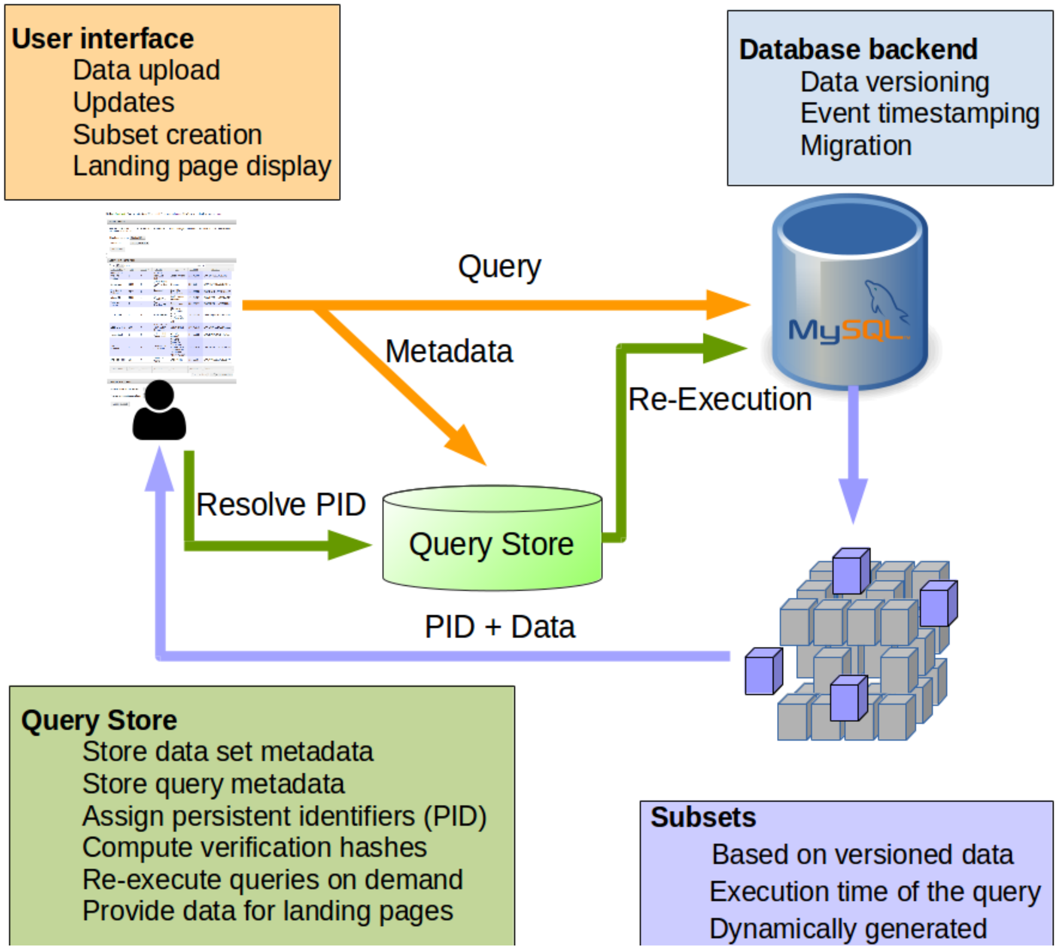


SELECT columns FROM table WHERE conditions HAVING conditions ORDER BY column

Data Citation – Deployment

- Researcher uses workbench or tool to identify subset of data
- Upon executing selection („download“) user gets
  - Data (package, access API, …)
  - PID (e.g. DOI) (Query is time-stamped and stored)
  - Hash value computed over the data for local storage
  - Recommended citation text (e.g. BibTeX)
  - Query string
- PID resolves to landing page
  - Provides detailed metadata, link to parent data set, subset,…
  - Option to retrieve original data OR current version OR changes
- Upon activating PID associated with a data citation
  - Query is re-executed against time-stamped and versioned DB
  - Results as above are returned
- Query store aggregates data usage

Reproducible Subsets with SQL Prototype



User interface
- Data upload
- Updates
- Subset creation
- Landing page display

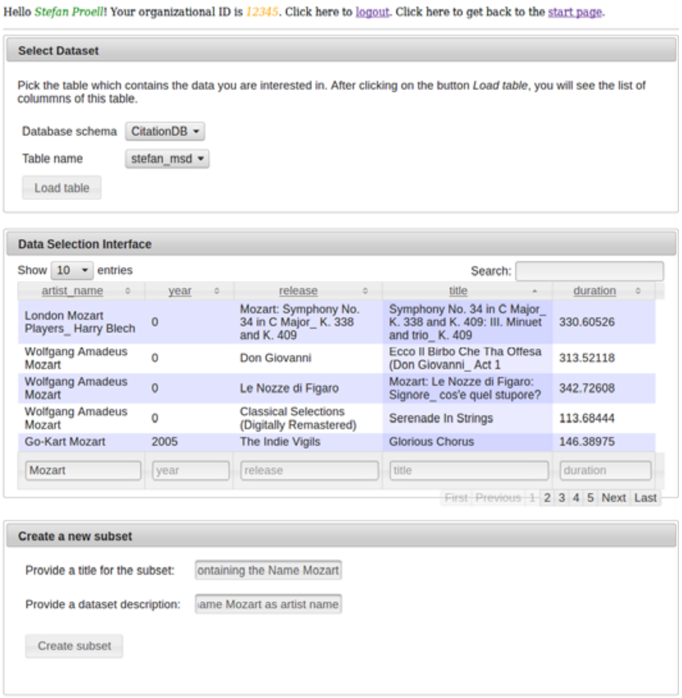
Database backend
- Data versioning
- Event timestamping
- Migration

Query Store
- Store data set metadata
- Store query metadata
- Assign persistent identifiers (PID)
- Compute verification hashes
- Re-execute queries on demand
- Provide data for landing pages

Subsets
- Based on versioned data
- Execution time of the query
- Dynamically generated

Implementation Overview

- Presentation layer
  - Web interface
- Application server layer
  - CSV module
  - Query store module
  - Persistent identification module
  - Result set verification module
- Data server layer
  - Database module



Hello *Stefan Proell*! Your organizational ID is *12345*. Click here to logout. Click here to get back to the start page.

Select Dataset

Pick the table which contains the data you are interested in. After clicking on the button *Load table*, you will see the list of colummns of this table.

Database schema: CitationDB

Table name: stefan_msd

Data Selection Interface

Show 10 entries

Search:

artist_name	year	release	title	duration
London Mozart Players_ Harry Blech	0	Mozart: Symphony No. 34 in C Major_ K. 338 and K. 409	Symphony No. 34 in C Major_ K. 338 and K. 409: III. Minuet and trio_ K. 409	330.60526
Wolfgang Amadeus Mozart	0	Don Giovanni	Ecco Il Birbo Che Tha Offesa (Don Giovanni_ Act 1	313.52118
Wolfgang Amadeus Mozart	0	Le Nozze di Figaro	Mozart: Le Nozze di Figaro: Signore_ cos'e quel stupore?	342.72608
Wolfgang Amadeus Mozart	0	Classical Selections (Digitally Remastered)	Serenade In Strings	113.68444
Go-Kart Mozart	2005	The Indie Vigils	Glorious Chorus	146.38975
Mozart	year	release	title	duration

Create a new subset

Provide a title for the subset: ontaining the Name Mozart

Provide a dataset description: ame Mozart as artist name

- Technologies: Java 8, Maven 3, MySQL 5.7, Hikari CP, JSF, Primefaces, jQuery

Prototype Demo

- Demo SQL Prototype
-

Videos available at: http://www.datacitation.eu/

Git as Data Backend

- Git
  - Distributed source code management software
  - Version control
  - Track changes
  - Ideal for text based file formats
- Advantages of Git
  - Local install possible
  - Available for all platforms
  - Repositories can be easily shared
  - Does not require central administration
  - Open source

Query Store + Git

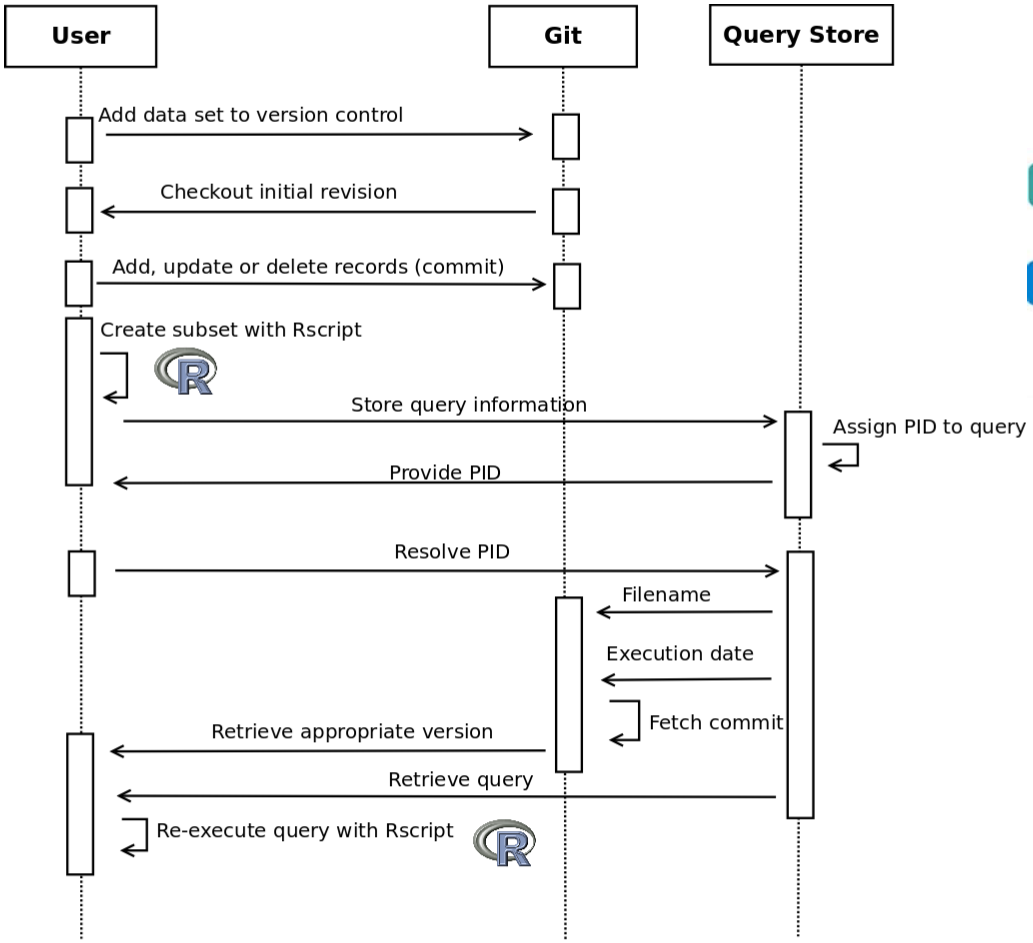
- Provide the same interface
  - Data selection with GUI
  - Git as backend
  - Query store preserves CSV2SQL query
  - Re-execution on top of CSV file revision
  -
  -
- Git as Data Backend
  - Ideal for text based formats
  - Simple query translation via the interface
  - Version all changes by commiting
  - Sharing via repositories (e.g. Github)

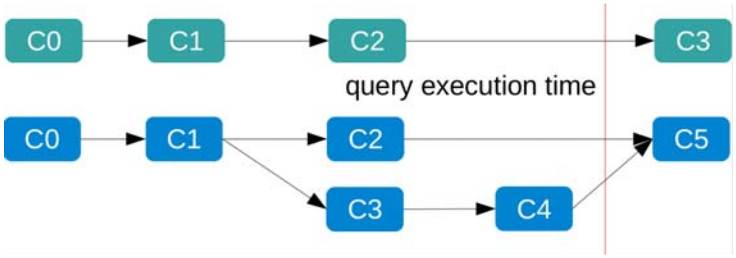
Scripts and Git Branches

- Subsets are created with a scripting language (e.g. R)
  - Select columns, filter records and sort result set
  - Script produces CSV file
- Users store the subsetting script also in Git
  - Subsetting process can be automatically executed
  - The subsetting script is also stored in Git
  - Metadata file describes script execution, language version, etc
- Use Git to retrieve proper data set version and re-execute script on retrieved file

- Advantage: Simple method, Integration with a Query Store
- Disadvantage: Git commit history contains data set and script files

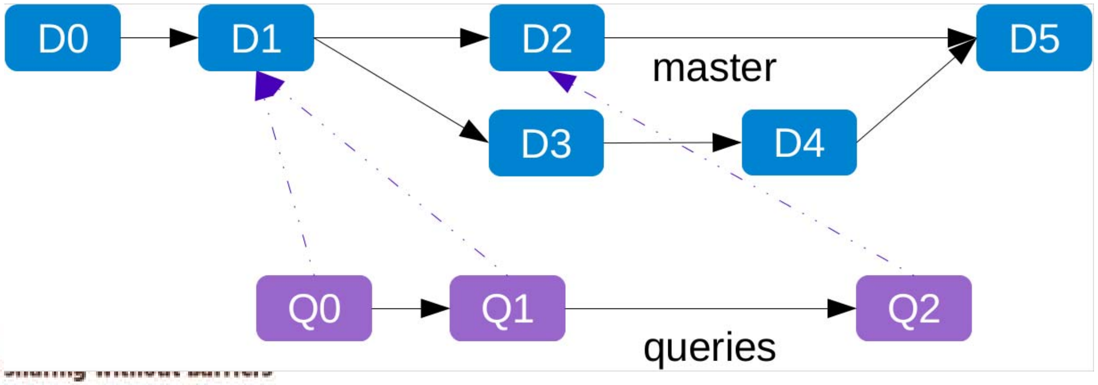
Reproducible Subsets with Git


User
Git
Query Store
Add data set to version control
Checkout initial revision
Add, update or delete records (commit)
Create subset with Rscript
Store query information
Assign PID to query
Provide PID
Resolve PID
Filename
Execution date
Fetch commit
Retrieve appropriate version
Retrieve query
Re-execute query with Rscript

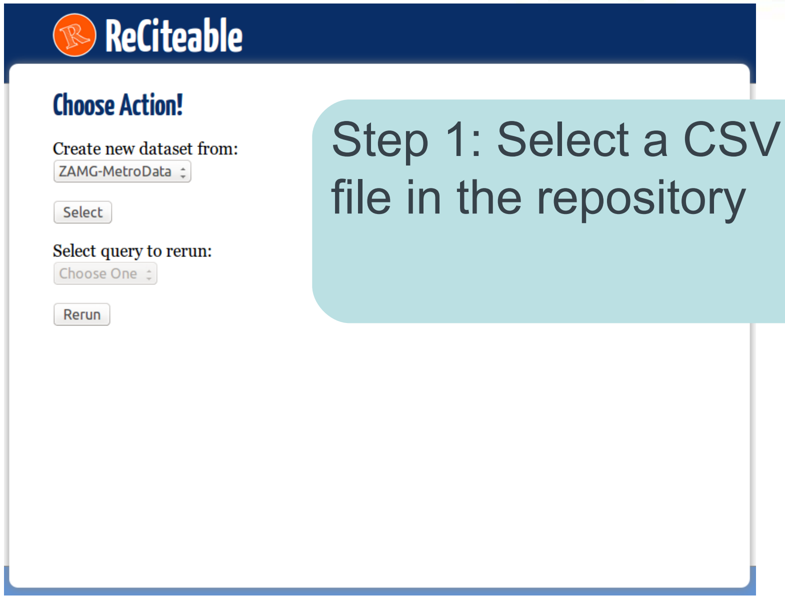


Reproducible Subsets with Git Branches

- Using the Git branching model
  - Branches allow separation of data and scripts
  - Keeps commit history clean
    - Allows merging of data files
  - Use commit hash for identification
    - Assigned PID hashed with SHA1
    - Use hash of PID as filename
  - Orphaned branch for queries and metadata files



Reproducible Subsets with Git Prototype

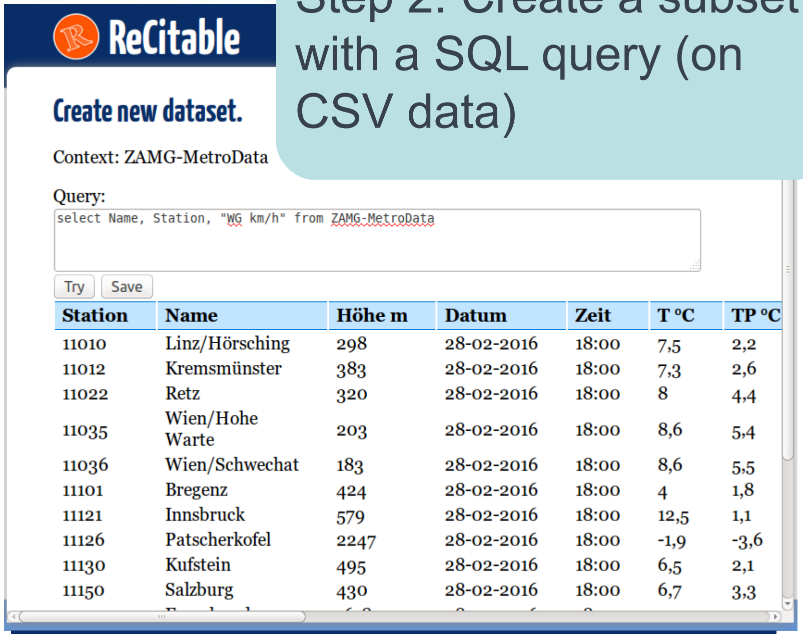


Choose Action!

Create new dataset from: ZAMG-MetroData

Select query to rerun: Choose One

Step 1: Select a CSV file in the repository



Create new dataset.

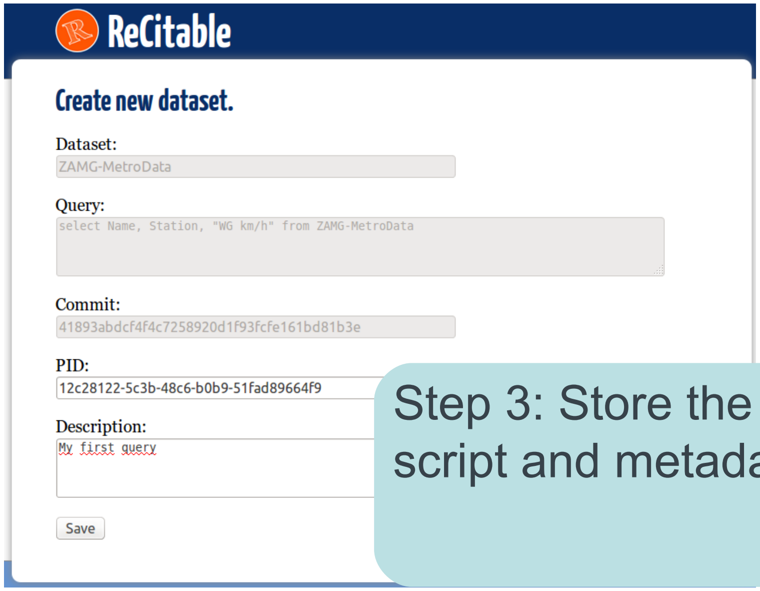
Context: ZAMG-MetroData

Query:

```
select Name, Station, "WG km/h" from ZAMG-MetroData
```

Station	Name	Höhe m	Datum	Zeit	T °C	TP °C
11010	Linz/Hörsching	298	28-02-2016	18:00	7,5	2,2
11012	Kremsmünster	383	28-02-2016	18:00	7,3	2,6
11022	Retz	320	28-02-2016	18:00	8	4,4
11035	Wien/Hohe Warte	203	28-02-2016	18:00	8,6	5,4
11036	Wien/Schwechat	183	28-02-2016	18:00	8,6	5,5
11101	Bregenz	424	28-02-2016	18:00	4	1,8
11121	Innsbruck	579	28-02-2016	18:00	12,5	1,1
11126	Patscherkofel	2247	28-02-2016	18:00	-1,9	-3,6
11130	Kufstein	495	28-02-2016	18:00	6,5	2,1
11150	Salzburg	430	28-02-2016	18:00	6,7	3,3

Step 2: Create a subset with a SQL query (on CSV data)



Create new dataset.

Dataset: ZAMG-MetroData

Query:

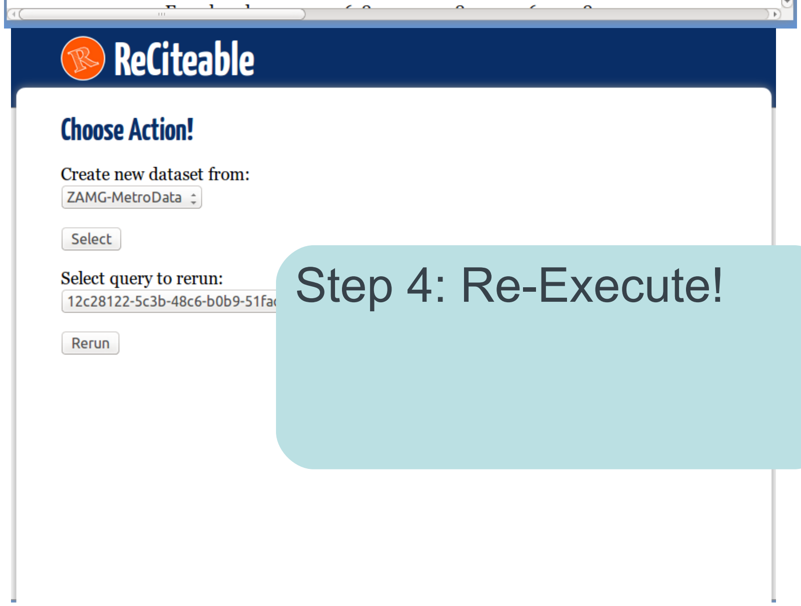
```
select Name, Station, "WG km/h" from ZAMG-MetroData
```

Commit: 41893abdcf4f4c7258920d1f93fcfe161bd81b3e

PID: 12c28122-5c3b-48c6-b0b9-51fad89664f9

Description: My first query

Step 3: Store the query script and metadata



Choose Action!

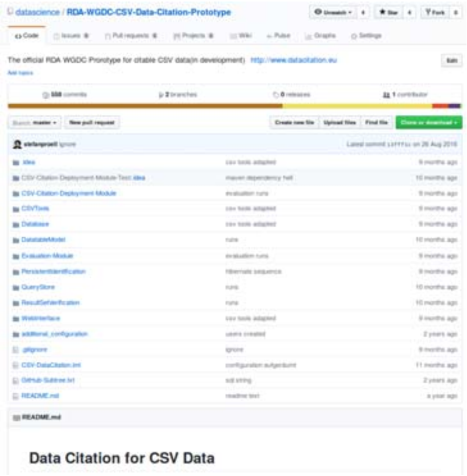
Create new dataset from: ZAMG-MetroData

Select query to rerun: 12c28122-5c3b-48c6-b0b9-51fa

Step 4: Re-Execute!

RDA WGDC Prototypes

- SQL Backend
- Git Backend
- Source code of all prototypes available at Github
- https://www.github.com/datascience



datascience / RDA-WGDC-CSV-Data-Citation-Prototype

The official RDA WGDC Prorotype for citable CSV data(in development) http://www.datacitation.eu

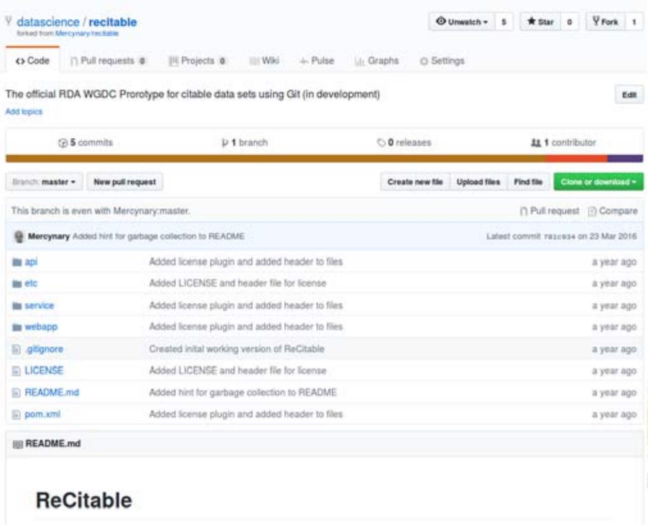
558 commits · 2 branches · 0 releases · 1 contributor

stefanproell ignore — Latest commit [illegible] on 26 Aug 2016

.idea	csv tools adapted	9 months ago
CSV-Citation-Deployment-Module-Test/.idea	maven dependency hell	10 months ago
CSV-Citation-Deployment-Module	evaluation runs	9 months ago
CSVTools	csv tools adapted	9 months ago
Database	csv tools adapted	9 months ago
DatatableModel	runs	10 months ago
Evaluation-Module	evaluation runs	9 months ago
PersistentIdentification	hibernate sequence	9 months ago
QueryStore	runs	10 months ago
ResultSetVerification	runs	10 months ago
WebInterface	csv tools adapted	9 months ago
additional_configuration	users created	2 years ago
.gitignore	ignore	9 months ago
CSV-DataCitation.iml	configuration aufgeräumt	11 months ago
GitHub-Subtree.txt	sql string	2 years ago
README.md	readme text	a year ago

README.md

Data Citation for CSV Data



datascience / recitable
forked from Mercynary/recitable

The official RDA WGDC Prorotype for citable data sets using Git (in development)

5 commits · 1 branch · 0 releases · 1 contributor

This branch is even with Mercynary:master.

Mercynary Added hint for garbage collection to README — Latest commit 781c934 on 23 Mar 2016

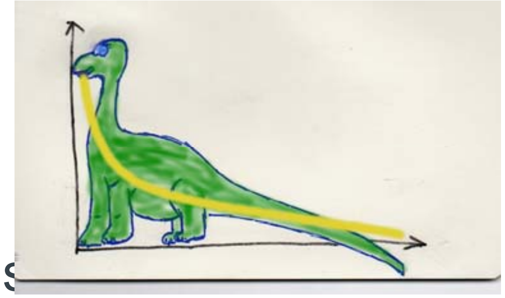
api	Added license plugin and added header to files	a year ago
etc	Added LICENSE and header file for license	a year ago
service	Added license plugin and added header to files	a year ago
webapp	Added license plugin and added header to files	a year ago
.gitignore	Created inital working version of ReCitable	a year ago
LICENSE	Added LICENSE and header file for license	a year ago
README.md	Added hint for garbage collection to README	a year ago
pom.xml	Added license plugin and added header to files	a year ago

README.md

ReCitable

Conclusion

- Query based data citation for evolving research data
  - Enhances reproducibility
  - Relies on data versioning and query (script) timestamping
- Implementation in small scale settings
  - Git repositories can be easily shared
  - Metadata included
- Implementation in large scale settings
  - Versioning often already available
  - Interfaces for subsetting processes can be us
    implementation



Thank You

Questions?
Comments?

Thank you very much for your attention!
stefan.proell@tuwien.ac.at
www.datacitation.eu
🐦 @stefanproell