RDA Global Adoption week

15 - 19 June 2020

→ The RDA Global Adoption Week: 15-19 June 2020
→ focused on **five areas of the research data lifecycle**

Day & Topic	Sessions
Monday, 15th June 2020 - Data Management Planning	14:00 UTC + 23:00 UTC
Tuesday, 16th June 2020 - Data Description	06:00 UTC + 14:00 UTC
Wednesday, 17th June 2020 - Identify, Store and Preserve	07:00 UTC + 14:00 UTC
Thursday, 18th June 2020 - Disseminate, Link and Find	07:00 UTC + 12:00 UTC
Friday, 19th June 2020 - Policy, Legal Compliance and Capacity	05:00 UTC + 13:00 UTC

Originally planned for the RDA 15th Plenary, the Adoption Week aims to **demonstrate the wide variety of RDA adoptable and adopted solutions to data sharing challenges** across research practices, domains and geographies.

Purpose of the week:

- Learn about RDA Outputs
- Converse with speakers from all around the world who have created and implemented them
- Determine how best to integrate those data sharing solutions into your own projects

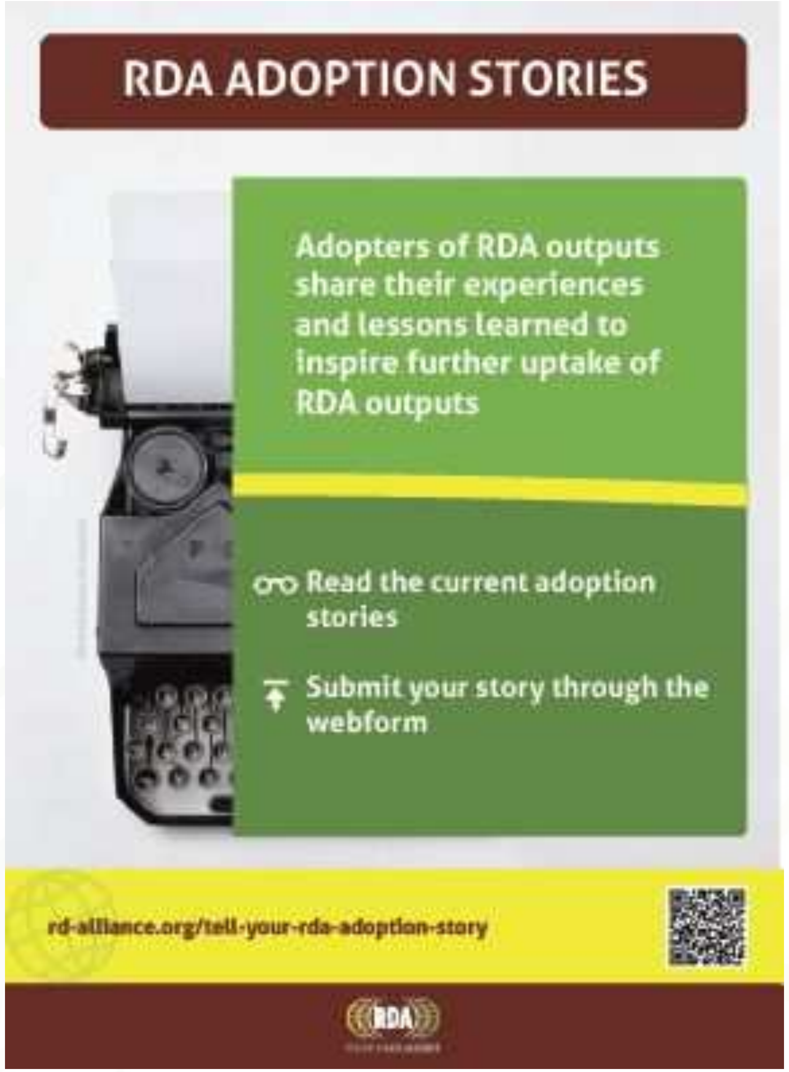
Recommendations and outputs catalogue

- RDA Outputs are classified as **RDA Recommendations** *(official, endorsed results of RDA Groups)*, **Supporting Outputs** *(useful solutions from our RDA Working and Interest Groups)* or **other Outputs**
- **They can be searched according to their status, Data Life Cycle topics or scientific domain**

rd-alliance.org/recommendations-and-outputs/catalogue

Tell your adoption story

- **Are you an adopter?** RDA is actively seeking new adoption stories to inspire the further uptake of RDA outputs.
- **Submit your story here:** **https://www.rd-alliance.org/tell-your-rda-adoption-story**



CODATA Data Science Journal CfP

- **RDA special collection themes:**
  - Results produced by an IG or WG;
  - Description of an Adoption Case outlining how a specific recommendation or output has been implemented;
  - Other types of work related to RDA activities.
- RDA Europe 4.0 still has funds available for the publication of articles in DSJ
- Open to all interested applicants regardless of their geographical provenance.
- **Deadline 17 July**

RESEARCH DATA ALLIANCE

Thursday 18th June

07:00 UTC

Disseminate, Link & Find

An increasing number of publishers and journals are implementing policies that require or recommend published articles to be accompanied by the underlying research data.

1. **Data Discovery Paradigms IG**
   - **Survey on the practices in data search services**
     Mingfang Wu (ARDC)
   - **Eleven quick tips and User requirements and recommendations**
     Fotis Psomopoulos (INAB CERTH)

Followed by Q&A

2. **FAIR data maturity model: specification and guidelines**

   Keith Russell (ARDC)

Followed by Q&A

3. **Workflows for Research Data Publishing: Models and Key Components Recommendation -** Introducing Maneage: customizable framework for managing data lineage

   Mohammad Akhlaghi (IAC)

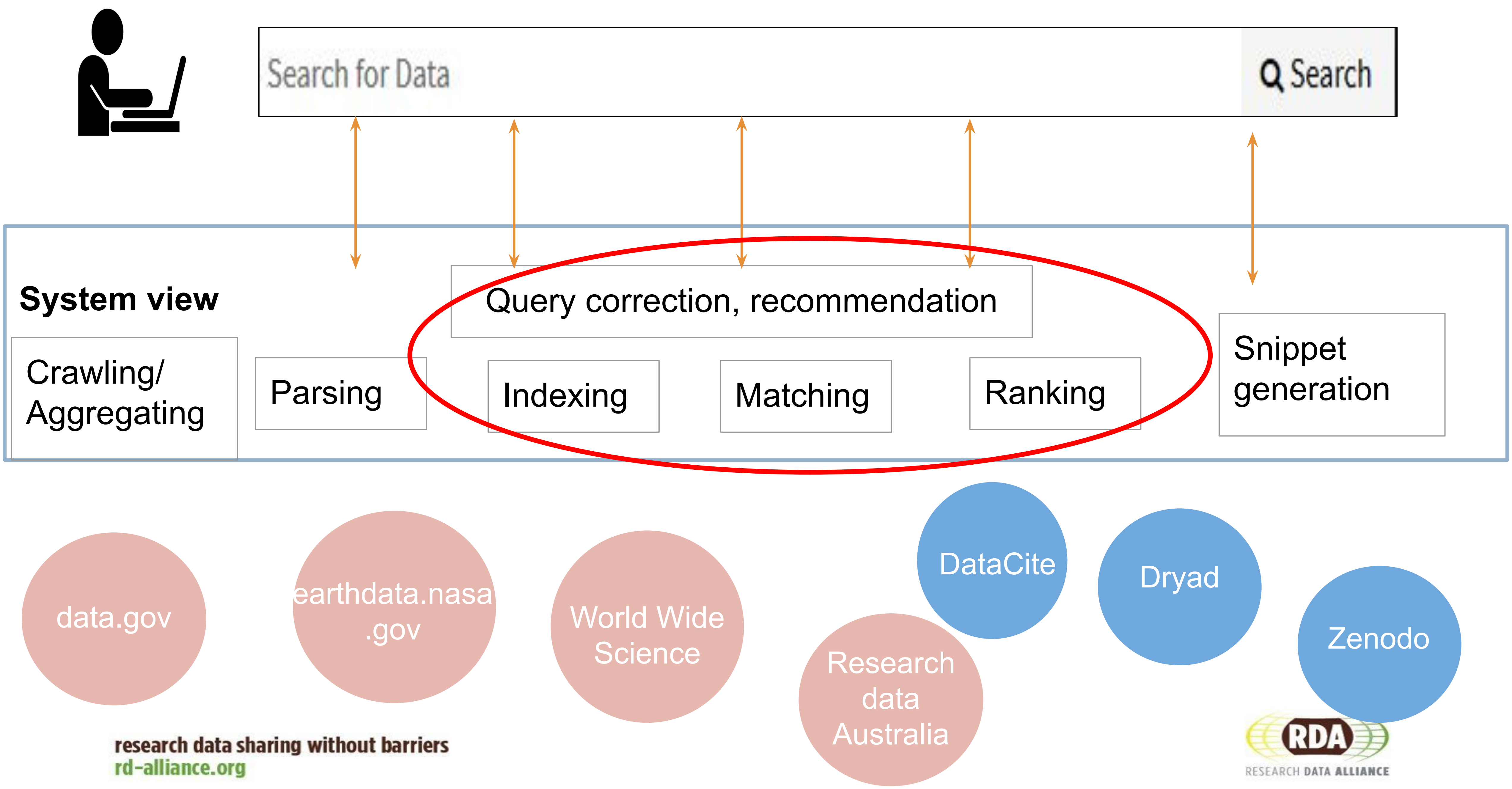
Followed by Q&A

Data Discovery Paradigms IG

Relevancy Ranking Task Force

RDA Global Adoption week
18 June 2020

research data sharing without barriers
rd-alliance.org


Search for Data
Search
System view
Crawling/
Aggregating
Parsing
Query correction, recommendation
Indexing
Matching
Ranking
Snippet
generation
data.gov
earthdata.nasa
.gov
World Wide
Science
Research
data
Australia
DataCite
Dryad
Zenodo

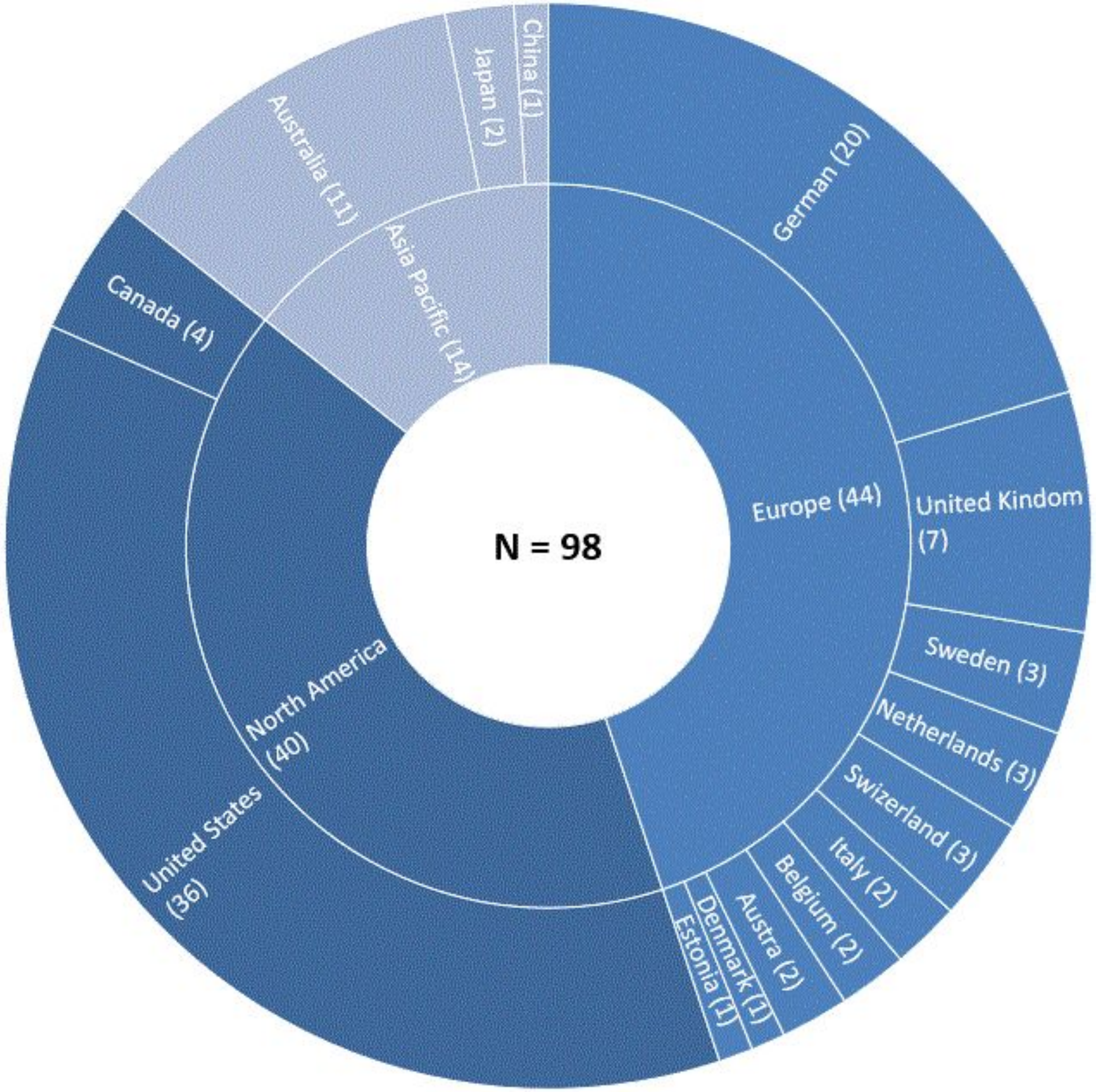
Survey goals

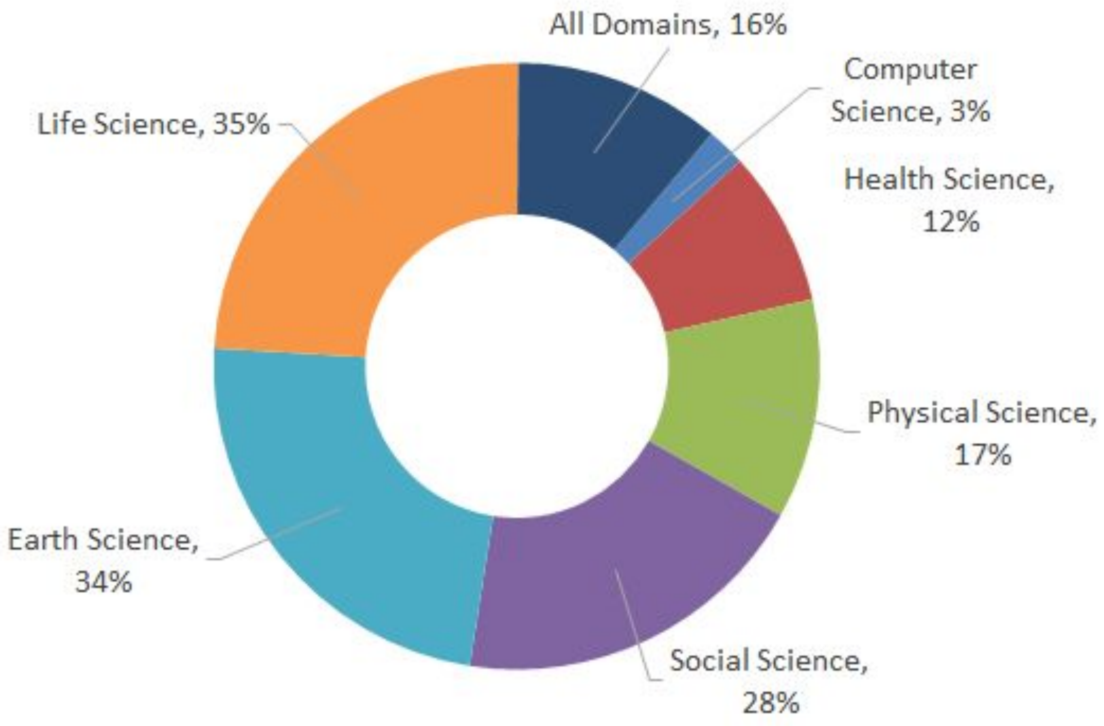
- Investigate what data search systems and ranking models have been deployed.
- Serve as a benchmark to be looked back on in future to assess how much and in what ways data search has improved.
- Identify potential collaborative projects from the Survey

Survey Design (33 Questions)

1. What are characteristics of each repositories? (5)
2. What are system configurations (e.g., ranking model, index methods, query methods)? (7)
3. What are evaluation methods and benchmark? (10)
4. What methods have been used to boost search-ability to web search engines? (2)
5. What other technologies or system configurations have been employed? (5)
6. Wish list for future activities for the RDA relevance task force (2)

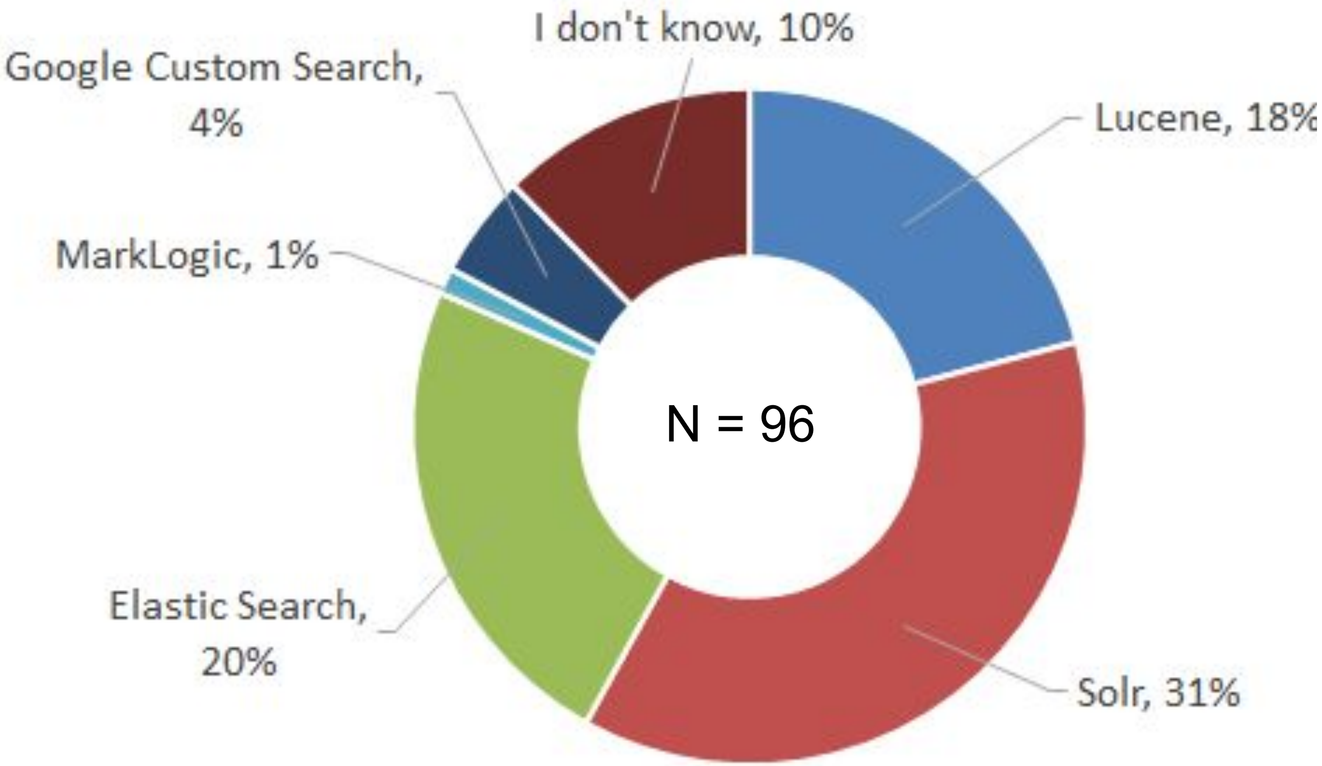
Participants background

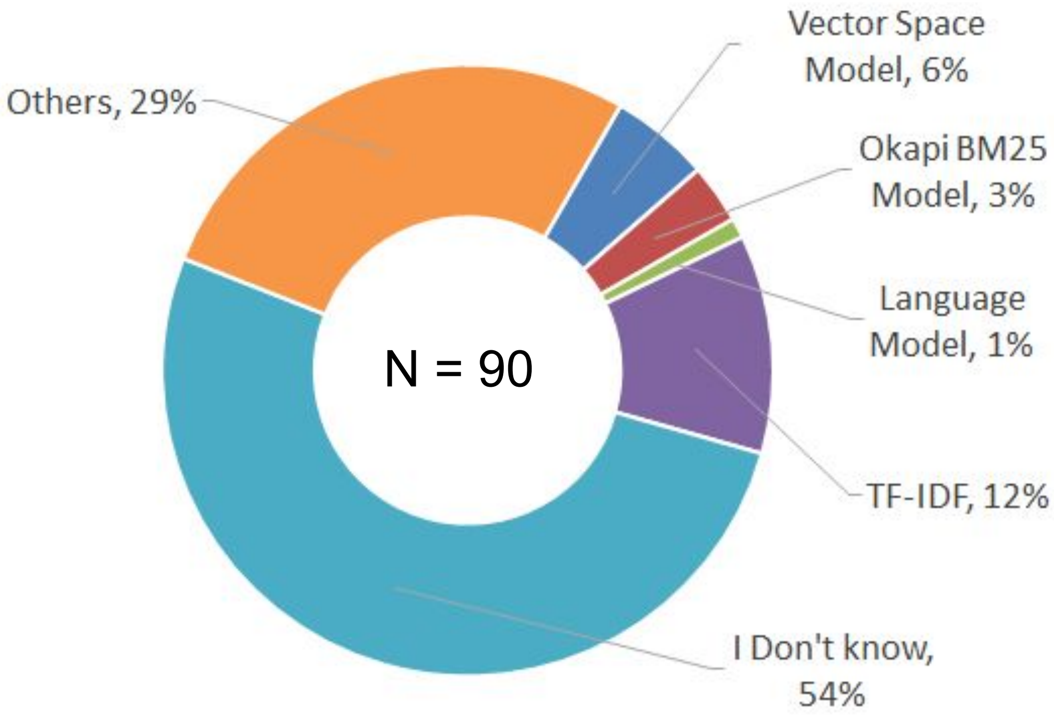




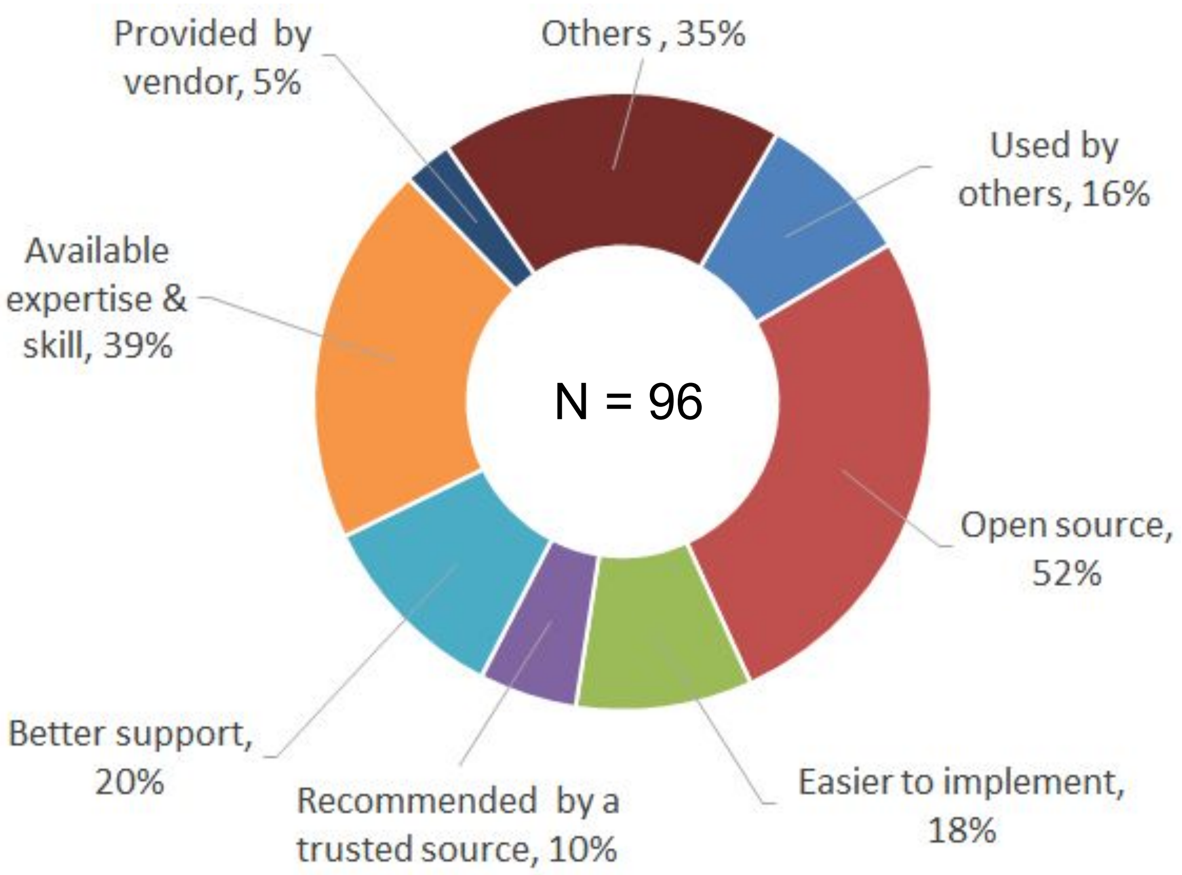
Survey result highlights ...

Data repositories use common search systems





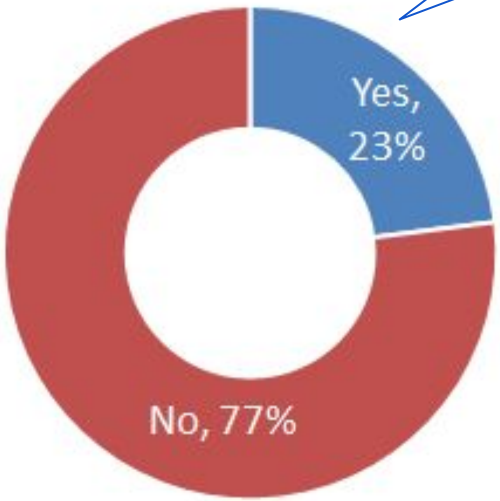
Open source and available skills are top reasons for choosing a search system

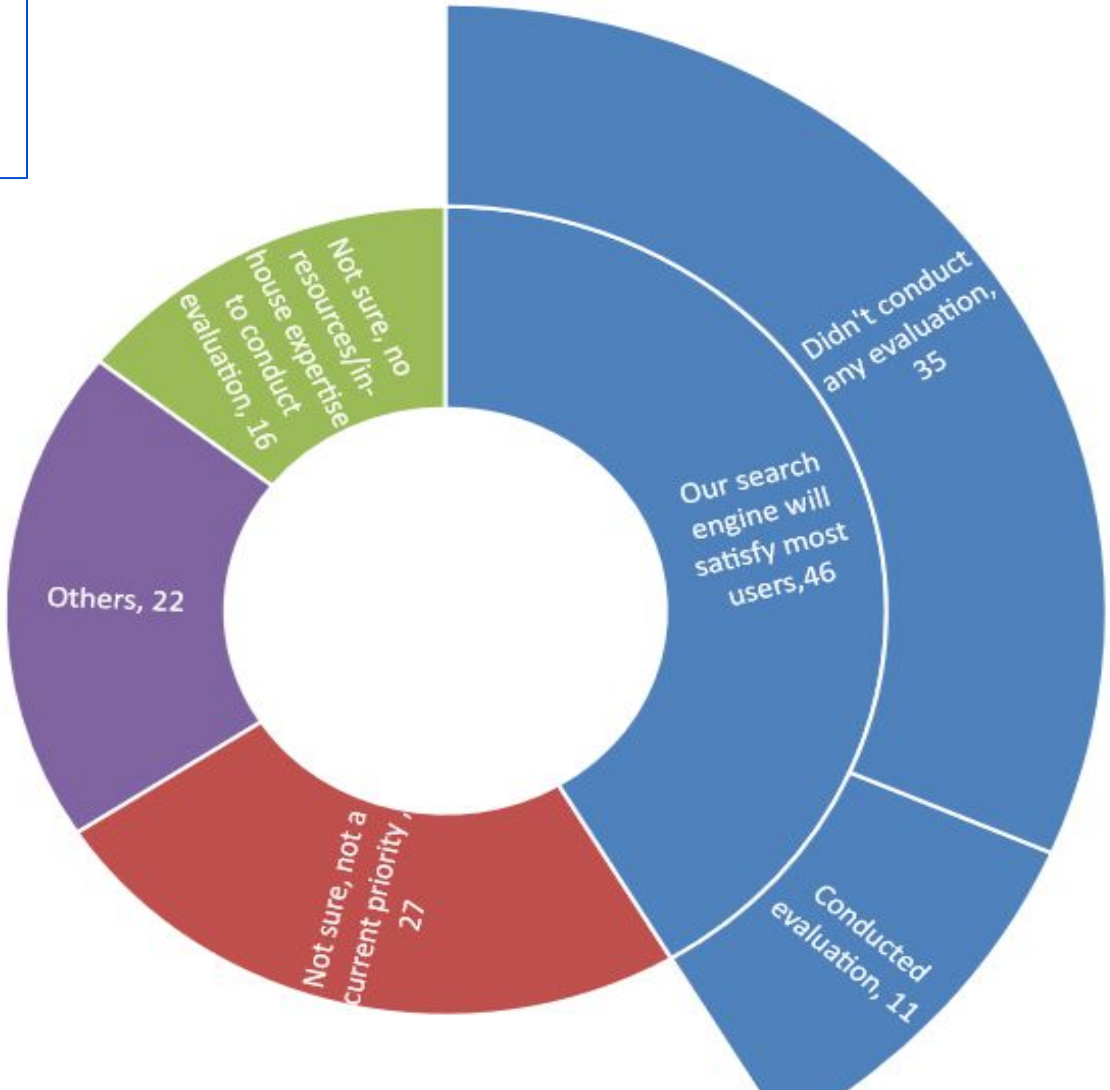


Majority didn’t conduct any kind of evaluations

9 Created test collection
11 Informal evaluation
6 Log analysis

No performance measure was provided





Summary

- Repositories desire guidelines for improving relevancy ranking in their data search system, with small repositories having the greatest need.
- Repositories understand that their search systems need to be evaluated and improved, but often lack the resources (time and/or expertise) to explore and evaluate the available options.
- The study concludes that there is an opportunity for people working in the search space to collaborate, to build test collections and other efforts that offer the greatest improvements in search services at the lowest cost.

Khalsa, SiriJodha; Cotroneo, Peter; Wu, Mingfang (2018), “A survey of current practices in data search services”, Mendeley Data, v1 http://dx.doi.org/10.17632/7j43z6n22z.1

Thank you

Contact:

mingfang.wu@ardc.edu.au
sjsk@nsidc.org
fpsom@certh.gr

RESEARCH DATA ALLIANCE

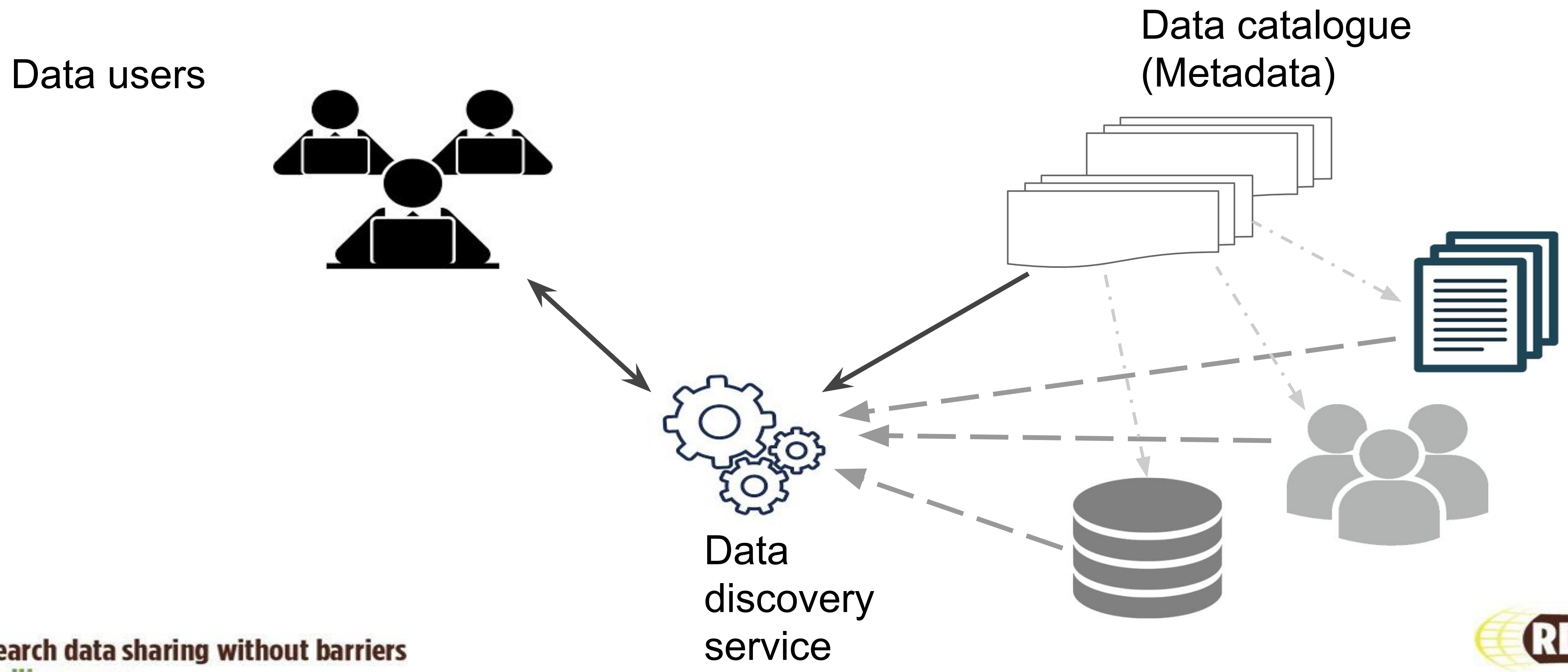
Data Discovery Paradigms Interest Group

RDA Global Adoption week
18 June 2020

research data sharing without barriers
rd-alliance.org

DDP Interest Group: Motivation

Helping to make research data **Findable** to support users in discovering data.



DDP Interest Group: Objective

- Provide a forum where representatives across the spectrum of stakeholders and roles can explore how to improve data discovery.

- Produce actionable recommendations for data producers, data repositories, data services providers and data seekers.

Output I - Eleven quick tips for finding research data

Tip 1: Think about the data you need and why you need them.
Tip 2: Select the most appropriate resource.
Tip 3: Construct your query strategically.
Tip 4: Make the repository work for you.
Tip 5: Refine your search.
Tip 6: Assess data relevance and fitness-for-use.
Tip 7: Save your search and data- source details.
Tip 8: Look for data services, not just data.
Tip 9: Monitor the latest data.
Tip 10: Treat sensitive data responsibly.
Tip 11: Give back (cite and share data).

Best practices for data seeker

Can be used for learning and research skills training

Gregory K, Khalsa SJ, Michener WK, Psomopoulos FE, de Waard A, Wu M (2018) Eleven quick tips for finding research data. PLoS Comput Biol 14(4): e1006038. https://doi.org/10.1371/journal.pcbi.1006038

(8124 views, 2345 downloads)

Output 2 - User Requirements for a data repository

Nine requirements (from 79 use cases)

- Indication of data availability
- Connection of data with person/institution/paper/citations/grants
- Fully annotated data
- Filtering of data based on specific criteria on multiple fields at the same time
- Cross-referencing of data
- Visual analytics/inspections of data/thumbnail preview
- Sharing data in a collaborative environment
- Accompanying educational/training material
- Portal functionality similar to other established academic portals

Data repository operators can use the requirements for the following purposes:

- As a checklist for designing and implementing a data service portal.
- For existing data discovery services, the list of requirements can be used as guidelines for heuristic evaluation of a specific data discovery service, and therefore plan for future improvements when necessary.
- In the era of big data, research on data discovery paradigms is at an all-time high. A user's perspective provides a strong foundation on which to construct the paradigms of the future.

Output 2 - Recommendations for Data Repositories to make data discovery

Recommendations:

- Multiple query interfaces
- Multiple access points
- Assessable search result
- Readable and analysable metadata records
- Available bibliographic references
- Available data usage statistics
- Consistent interface
- Identifiable duplicats
- Findable from web search engines
- Interoperability with other repositories

Data repositories can take the ten recommendations:

- As guidelines when implementing a new repository
- As a checklist when conducting heuristic evaluation of an existing repository.

Data repositories can implement all or prioritise their implementation based on their user needs and available resources.

Use cases published to Zenodo https://doi.org/10.5281/zenodo.1050976 (124 views, 73 downloads)

Output 2 - User Requirements and Recommendations for Data Repositories

	REQ1: Data availability	REQ2: Connection of data	REQ3: Annotations	REQ4: Filtering	REQ5: Cross-referencing	REQ6: Inspection of data	REQ7: Collaborative environment	REQ9: Similarity across portals	REQ8: Training material
REC 1: Query interfaces				✓		✓		✓	Ten simple rules for finding data
REC 2: Multiple access points		✓		✓		✓		✓	
REC 3: Summarize search results	✓		✓			✓			
REC 4: Metadata records readable		✓	✓						
REC 5: Bibliographic references							✓		
REC 6: Usage statistics			✓						
REC 7: Consistentcy								✓	
REC 8: Identify duplicates		✓			✓				
REC 9: Findability from web SEs	Support data searches from web search engines								
REC 10: Interoperability	The Fair Data Principles								

Wu, M., Psomopoulos, F., Khalsa, S.J. and de Waard, A., 2019. Data Discovery Paradigms: User Requirements and Recommendations for Data Repositories. Data Science Journal, 18(1), DOI: http://doi.org/10.5334/dsj-2019-003

(1432 views, 396 downloads

RDA RESEARCH DATA ALLIANCE

Contact

fpsom@certh.gr

mingfang.wu@ardc.edu.au

sjsk@nsidc.org

https://www.rd-alliance.org/groups/data-discovery-paradigms-ig

Adoption of the FAIR Data Maturity Model

18 June 2020

Context

The principles are **NOT** strict

- **Ambiguity**
- Wide range of **interpretations** of FAIRness

Different **FAIR Assessment** Frameworks

- Different metrics
- No comparison of results
- No benchmark

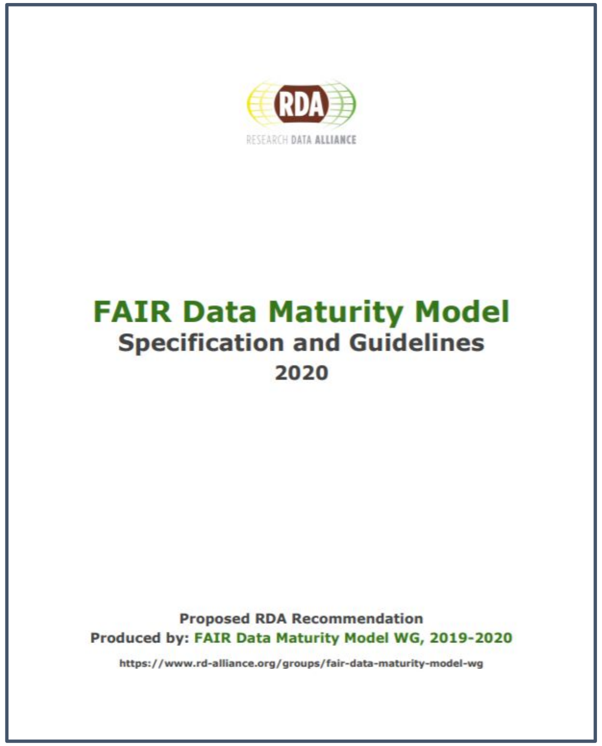
FAIR

SOLUTION is to bring together **stakeholders** to build on **existing approaches** and **expertise**

- Set of **core assessment criteria** for FAIRness
- FAIR **data maturity model & toolset**
- FAIR data **checklist**
- RDA recommendation

Join the **RDA** Working Group: RDA WG web page | GitHub

Public review period complete now to council


RESEARCH DATA ALLIANCE

FAIR Data Maturity Model
Specification and Guidelines
2020

Proposed RDA Recommendation
Produced by: FAIR Data Maturity Model WG, 2019-2020
https://www.rd-alliance.org/groups/fair-data-maturity-model-wg

THANKS TO ALL REVIEWERS

3600+ page views

14 comments

https://www.rd-alliance.org/group/fair-data-maturity-model-wg/outcomes/fair-data-maturity-model-specification-and-guidelines

Adoption examples

Early adopters – Experience sharing

- **Ge Peng** | NOAA
- **Anusuriya Devaraju** | FAIRsFAIR

… will share their relevant experience with regard to the adoption of the FDMM and answer to the following questions;

1. What is the level of adoption at your organisation? (E.g., pilot, production, ...)
2. Do you plan to continue to use the Recommendation?
3. Did you need to modify the Recommendation for your use?
4. Can you give an estimate of how much time / effort you have spent on the adoption so far?
5. What's your overall experience? (E.g., Very Good, Good, Fair, Poor)
6. Would you do it again?

Evaluating the FAIRness of Environmental Data

– Application of the RDA FAIR Data Maturity Indicators

Ge Peng, PhD

Cooperative Institute for Satellite Earth System Studies (CISESS) Between
U.S. National Oceanic and Atmospheric Administration (NOAA) and North Carolina State University
at NOAA National Centers for Environmental Information (NCEI)

#9 Workshop of the RDA FAIR Data Maturity Model Working Group, May 20–21, 2020

Purposes of Pilot Application

- Examine the relevancy of the RDA FAIR DMIs (v0.04)
- **Baseline the FAIRness of NCEI managed data**
  - In particular, *OneStop*-Ready datasets,
    - *OneStop* project was Initiated in 2015 to improve discovery and access services for NOAA datasets.
  - What worked?
- **Identify potential gaps & define path forward in NCEI data sharing practices**

Adopting OAIS RM & DSMM Helped!

Mapping FAIR Data Principles to NCEI/CICS-NC Data Stewardship Maturity Matrix (DSMM)

FAIR Data Principles (Wilkinson et al. 2016)	DSMM Key Components								
	Preservability	Accessibility	Usability	Production Sustainability	Data Quality Assurance	Data Quality Control/Monitoring	Data Quality Assessment	Transparency /Traceability	Data Integrity
F1. (meta)data are assigned a globally unique and eternally persistent identifier								L3	
F2. data are described with rich metadata (defined by R1 below)	L3		L3					L5	
F3. metadata clearly and explicitly include the identifier of the data it describes	L3		L3					L3	
F4. (meta)data are registered or indexed in a searchable resource		L2 & L3							
A1. (meta)data are retrievable by their identifier using a standardised communications protocol		L2 & L3	L3					L3	
A1.1. the protocol is open, free, and universally implementable		L3							
A1.2. the protocol allows for an authentication and authorization procedure, where necessary		L3							
A2. metadata are accessible, even when the data are no longer available		L2							
I1. (meta)data use a formal, accessible, shared, and broadly applicable language for knowledge representation	L3		L3						
I2. (meta)data use vocabularies that follow FAIR principles		L4							
I3. (meta)data include qualified references to other (meta)data	L3		L3						
R1. meta(data) are richly described with a plurality of accurate and relevant attributes	L3		L3						
R1.1. (meta)data are released with a clear and accessible data usage licence	*		*						
R1.2. (meta)data are associated with detailed provenance									
R1.3. (meta)data meet domain-relevant community standards	L3		L3						

* Can be easily implemented via relevant metadata entity and modified document template

(Version: v00r01 20200403; POC: gpeng@ncsu.edu; CC-BY 4.0)

Many data stewardship quality attributes are not explicitly addressed by the FAIR Data Principles.

- Most of data are open by default,
- Use agreements or use constraints,
- CC license not yet explicitly included.

Path Forward

Improving the FAIRness of NCEI & NOAA Data

- **Explicitly include** a data usage license, e.g. CC-BY 4.0; CC0, in the metadata record:
  - Discussions are on-going,
  - Procedure is under development.

Extending the Application Scope – under discussion

- **Assess:** 200+ additional NCEI datasets,
  - produced by NCEI's Center for Weather and Climate, *various stages of OneStop-ready.*
- **Examine** the scalability of the evaluation.

Integrating Assessment Results - Fairly

- Community guidelines – consistently curating and representing dataset quality information,
- Virtual workshop on July 13, 2020 – bringing together international domain experts,
- Contact me at gpeng@ncsu.edu if interested in participating or contributing.

RDA FAIR Data Maturity Model Adoption (Impression and Experience)

Anusuriya Devaraju & Hervé L'Hours
(on behalf of FAIRsFAIR)

FAIRsFAIR "Fostering FAIR Data Practices in Europe" has received funding from the European Union's Horizon 2020 project call H2020-INFRAEOSC-2018-2020 Grant agreement 831558

Repository Certification

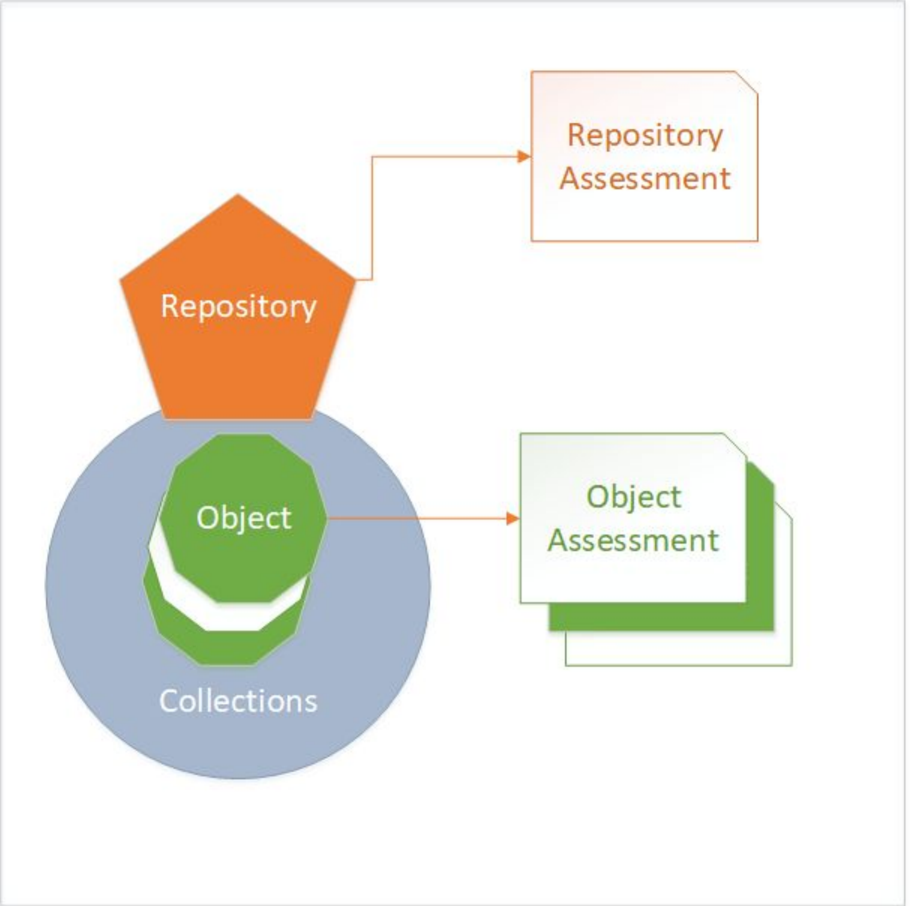
- CoreTrustSeal follows a self-assessment and peer review model
- FAIRsFAIR is offering support with a CoreTrustSeal+FAIR angle
- Map object characteristics to where repositories can enable FAIR

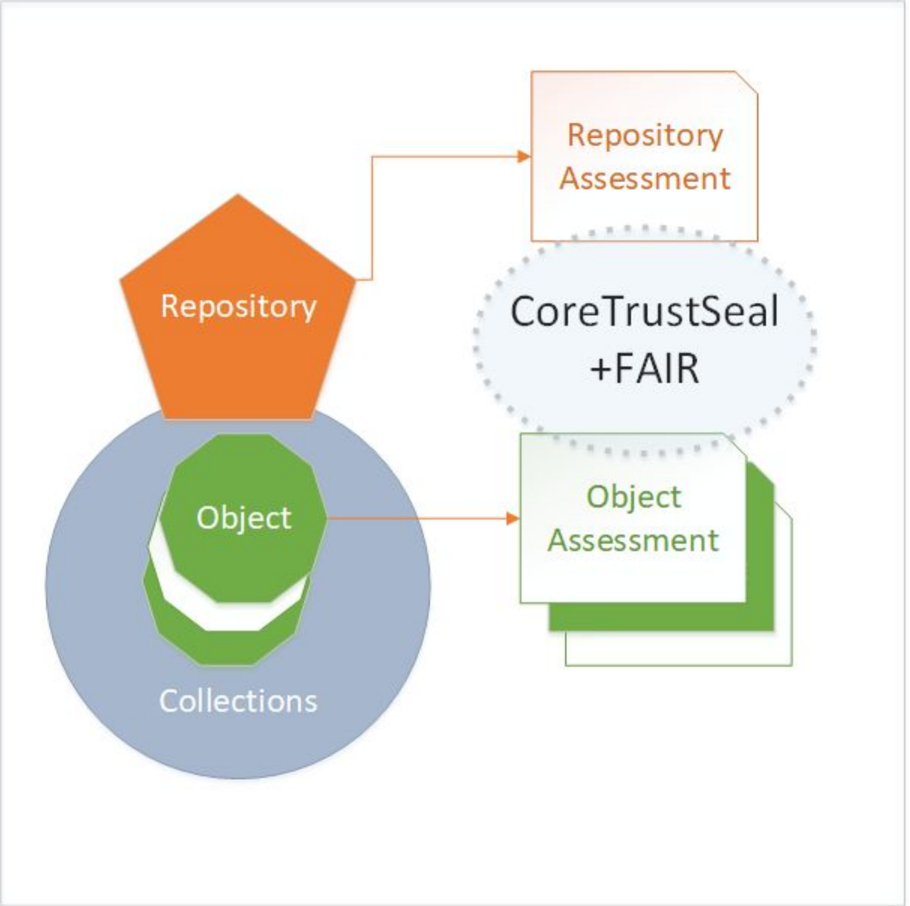
Repository Certification

- CoreTrustSeal follows a self-assessment and peer review model
- FAIRsFAIR is offering support with a CoreTrustSeal+FAIR angle
- Map object characteristics to where repositories can enable FAIR

Later:

- **Integrate object evaluation outcomes**


Repository Assessment
Repository
Object
Object Assessment
Collections


Repository
Repository Assessment
CoreTrustSeal +FAIR
Object
Object Assessment
Collections

Overall Adoption Experience

- The recommendation should be used as a starting reference point for data FAIRness assessment.
- Presentation - specification and guidelines are well structured!
- ‘What’ aspect of FAIR assessment
  - Descriptions of indicators are very helpful!
  - Suggestion - Include priority level next to each of the indicators.
  - Essential I-indicators missing (needs further work or not important?)
- ‘How’ aspect of FAIR assessment
  - Context matters (e.g., practices, data types)
  - Assessment details not always provide sufficient detail to implement tests.
  - Potential supporting technologies and services should be described.

Next steps

- Reach out to your communities as for the publishing of the **FAIR data maturity model: specification and guidelines** (i.e. RDA recommendation)
- Continuously provide feedback to the Editorial Team and pass on information with regards to the use of the **FAIR data maturity model: specification and guidelines** (i.e. RDA recommendation)

The editorial team will look into a release calendar and change management schedule

WORKSHOP #10

Possibly **September 2020**

Thank you!

Introducing Maneage:
Customizable framework for managing data lineage

[RDA Europe Adoption grant recipient. Submitted to IEEE CiSE (arXiv:2006.03018), Comments welcome]

Mohammad Akhlaghi

Instituto de Astrofísica de Canarias (IAC), Tenerife, Spain

RDA Global Adoption week

June 18th, 2020

Most recent slides available in link below (this PDF is built from Git commit d1faba6):

https://maneage.org/pdf/slides-intro-short.pdf

MINISTERIO DE CIENCIA, INNOVACIÓN Y UNIVERSIDADES

SUNDIAL
H2020 Innovative Training Network

This project has received financial support from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No. 721463 to the SUNDIAL ITN network.

Unión Europea
Fondo Europeo de Desarrollo Regional
"Una manera de hacer Europa"

This project received financial support from the European Union's Horizon 2020 research and innovation grant agreement No. 777388 to the RDA EU 4.0 project.

RESEARCH DATA ALLIANCE EUROPE

Universidad de La Laguna

Gobierno de Canarias
Consejería de Economía, Conocimiento y Empleo

Challenges of the RDA-WDS Publishing Data Workflows WG (DOI:10.1007/s00799-016-0178-2)

Challenges (also relevant to researchers, not just repositories)

- *Bi-directional linking*: how to link data and publications.
- *Software management:* how to manage, preserve, publish and cite software?
- *Metrics:* how often are data used.
- *Incentives to researchers:* how to communicate benefits of following good practices to researchers.

"*We would like to see a workflow that results in all* **scholarly objects being connected***, linked, citable, and persistent to allow researchers to navigate smoothly and to* **enable reproducible research***. This includes* **linkages** *between documentation, code, data, and journal articles in an integrated environment. Furthermore, in the ideal workflow, all of these objects need to be* **well documented** *to enable other researchers (or citizen scientists etc) to reuse the data for new discoveries.*"

General outline of a project (after data collection)

Existing solutions:

- Virtual machines
- Containers (e.g., Docker)
- OSs (e.g., Nix, GNU Guix)

Config environment?

Config options?

Repository?

Dep. versions?

What version?

Dependencies?

Software → Build

Hardware/data

Data base, or PID?

Calibration/version?

Integrity?

Confirmation bias?

Human error?

Runtime options?

What order?

→ Run software on data → Paper

Environment update?

In sync with coauthors?

History recorded?

Cited software?

Report this info?

Sync with analysis?

https://heywhatwhatdidyousay.wordpress.com
http://pngimages.net

Green boxes with sharp corners: *source*/input components/files.
Blue boxes with rounded corners: *built* components.
Red boxes with dashed borders: questions that must be clarified for each phase.

Science is a tricky business

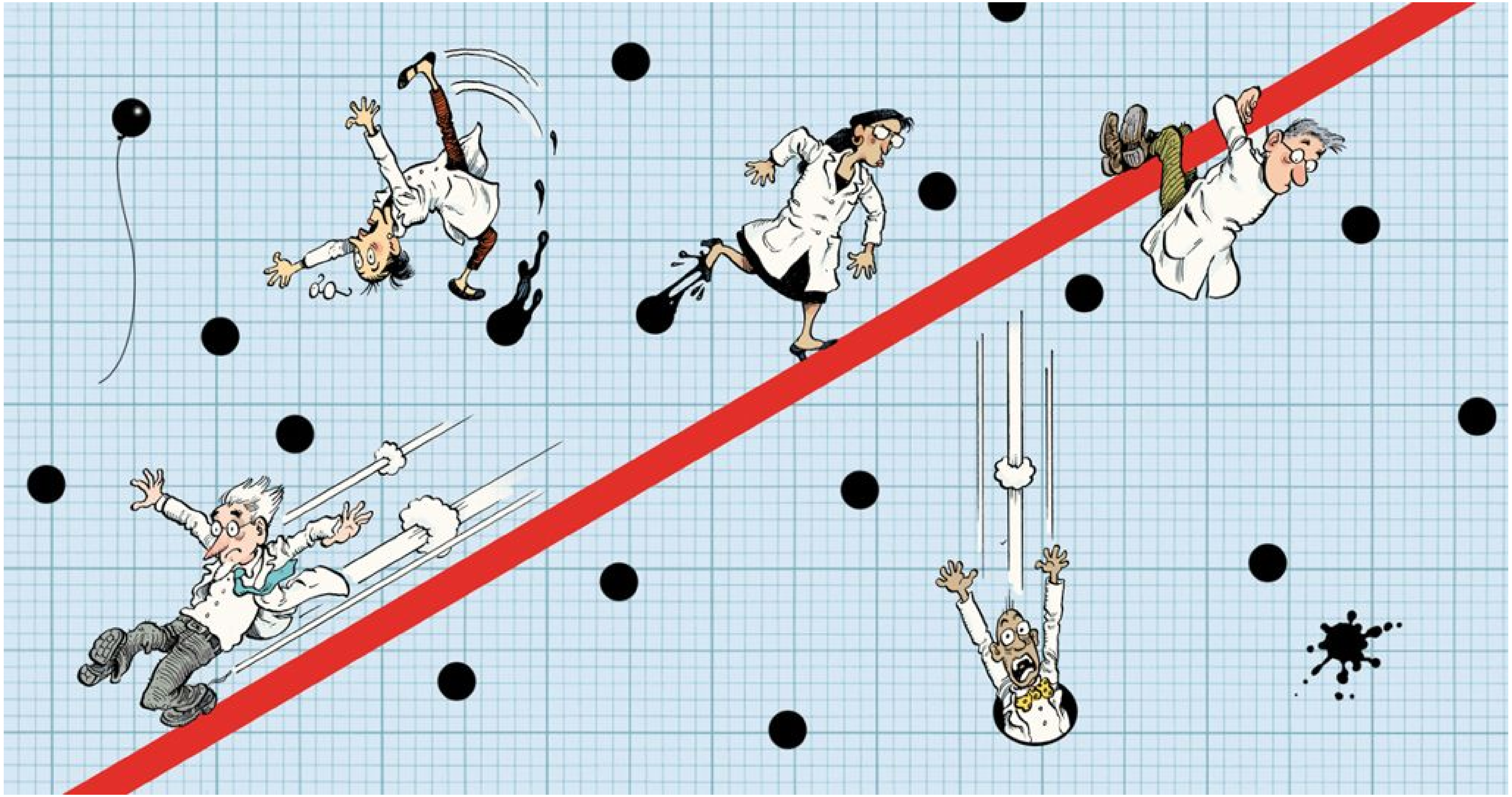


Image from nature.com ("Five ways to fix statistics", Nov 2017)

Data analysis [...] is a human behavior. Researchers who hunt hard enough will turn up a result that fits statistical criteria, but their discovery will probably be a false positive.

Five ways to fix statistics, Nature, 551, Nov 2017.

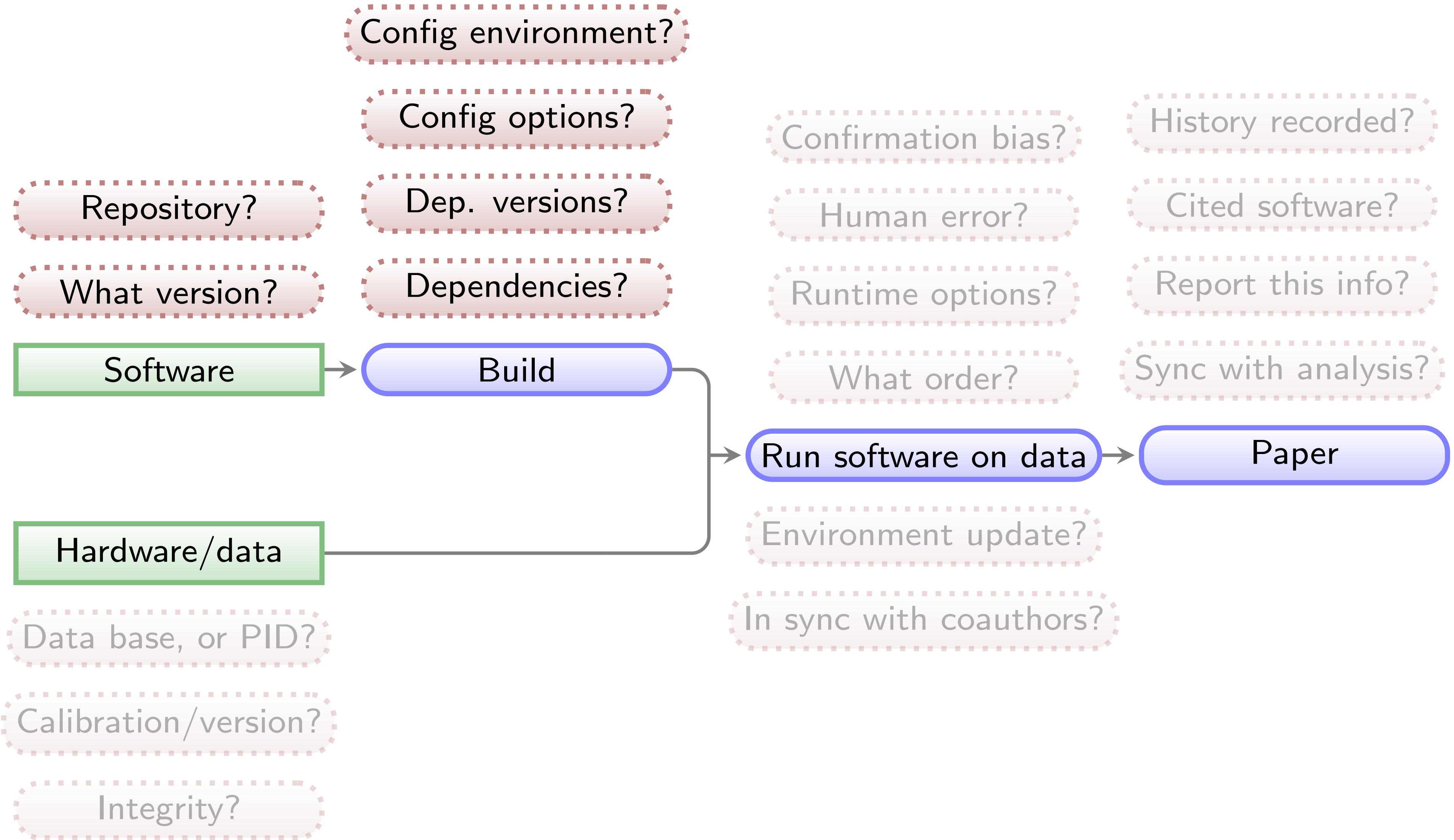
Founding criteria

Basic/simple principle:

Science is defined by its METHOD, not its result.

- **Complete/self-contained:**
  - Only dependency should be POSIX tools (discards Conda or Jupyter which need Python).
  - Must not require root permissions (discards tools like Docker or Nix/Guix).
  - Should be non-interactive or runnable in batch (user interaction is an incompleteness).
  - Should be usable without internet connection.
- **Modularity:** Parts of the project should be re-usable in other projects.
- **Plain text:** Project's source should be in plain-text (binary formats need special software)
  - This includes high-level analysis.
  - It is easily publishable (very low volume of $\times$100KB), archivable, and parse-able.
  - Version control (e.g., with Git) can track project's history.
- **Minimal complexity:** Occum's rasor: "Never posit pluralities without necessity".
  - Avoiding the fashionable tool of the day: tomorrow another tool will take its place!
  - Easier learning curve, also doesn't create a generational gap.
  - Is compatible and extensible.
- **Verifable inputs and outputs:** Inputs and Outputs must be automatically verified.
- **Free and open source software:** Free software is essential: non-free software is not configurable, not distributable, and dependent on non-free provider (which may discontinue it in N years).

General outline of a project (after data collection)

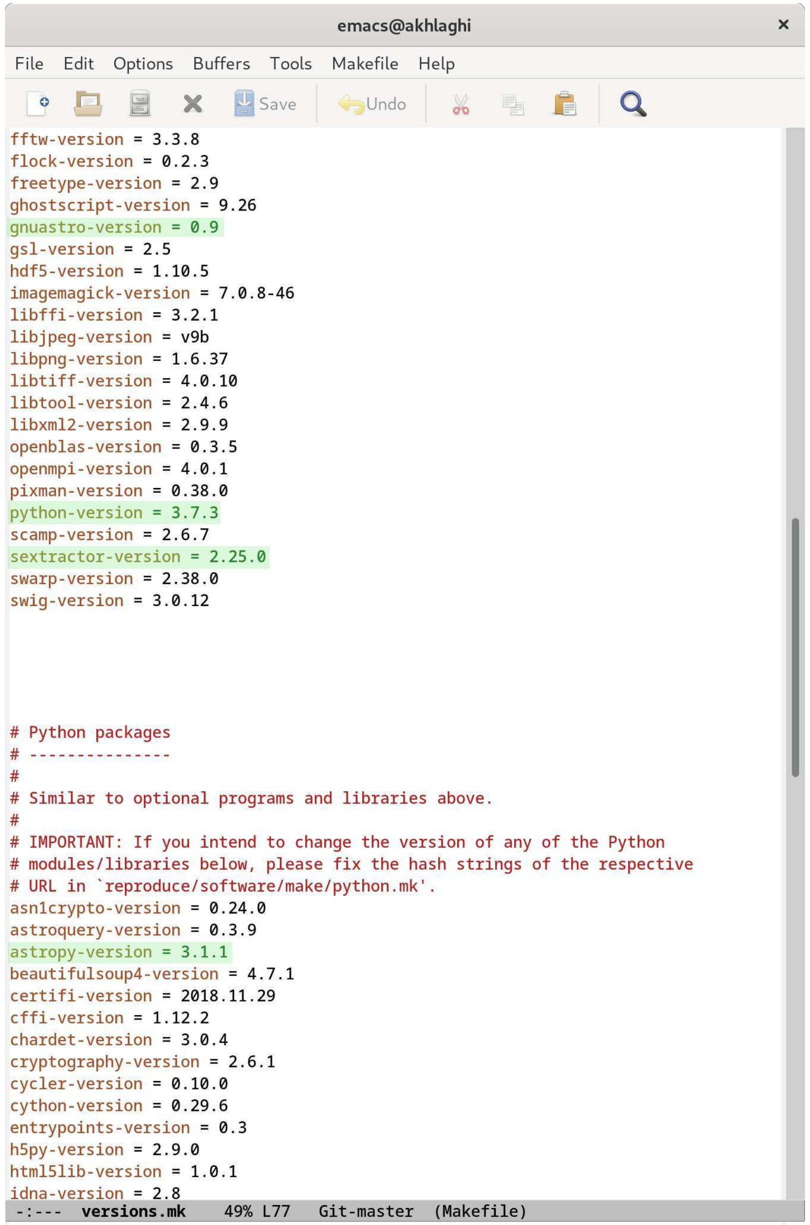


Green boxes with sharp corners: *source*/input components/files.
Blue boxes with rounded corners: *built* components.
Red boxes with dashed borders: questions that must be clarified for each phase.

Predefined/exact software tools

Reproducibility & software

Reproducing the environment (specific software versions, build instructions and dependencies) is also critically important for reproducibility.



- *Containers* or *Virtual Machines* are a binary black box.
- Maneage installs fixed versions of all necessary research software and their dependencies.
- Installs similar environment on GNU/Linux, or macOS systems.
- Works very much like a package manager (e.g., `apt` or `brew`).

Example: Matplotlib (a Python visualization library) build dependencies

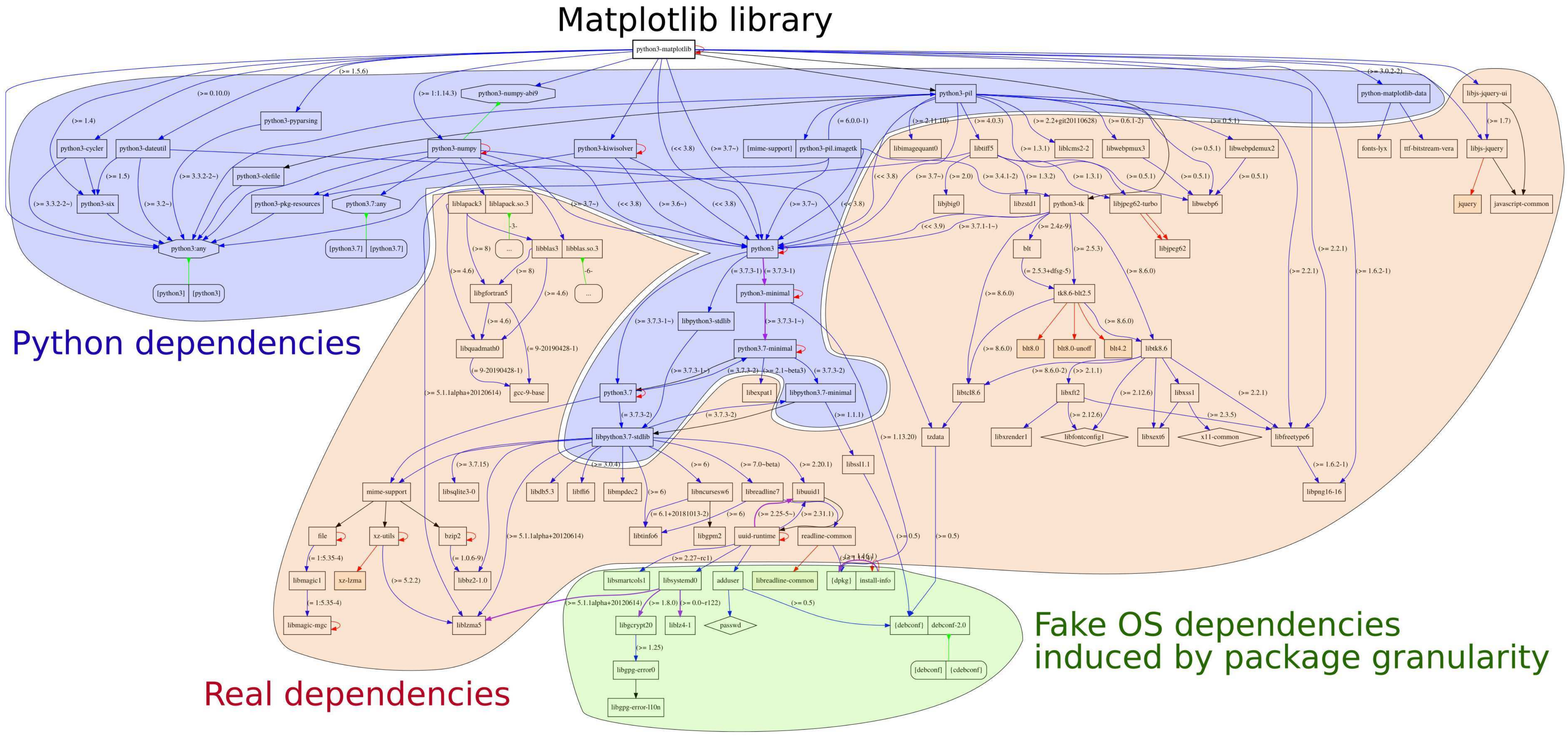


Fig. 1. Transitive dependencies of the software environment required by a simple "`import matplotlib`" command in the Python 3 interpreter.

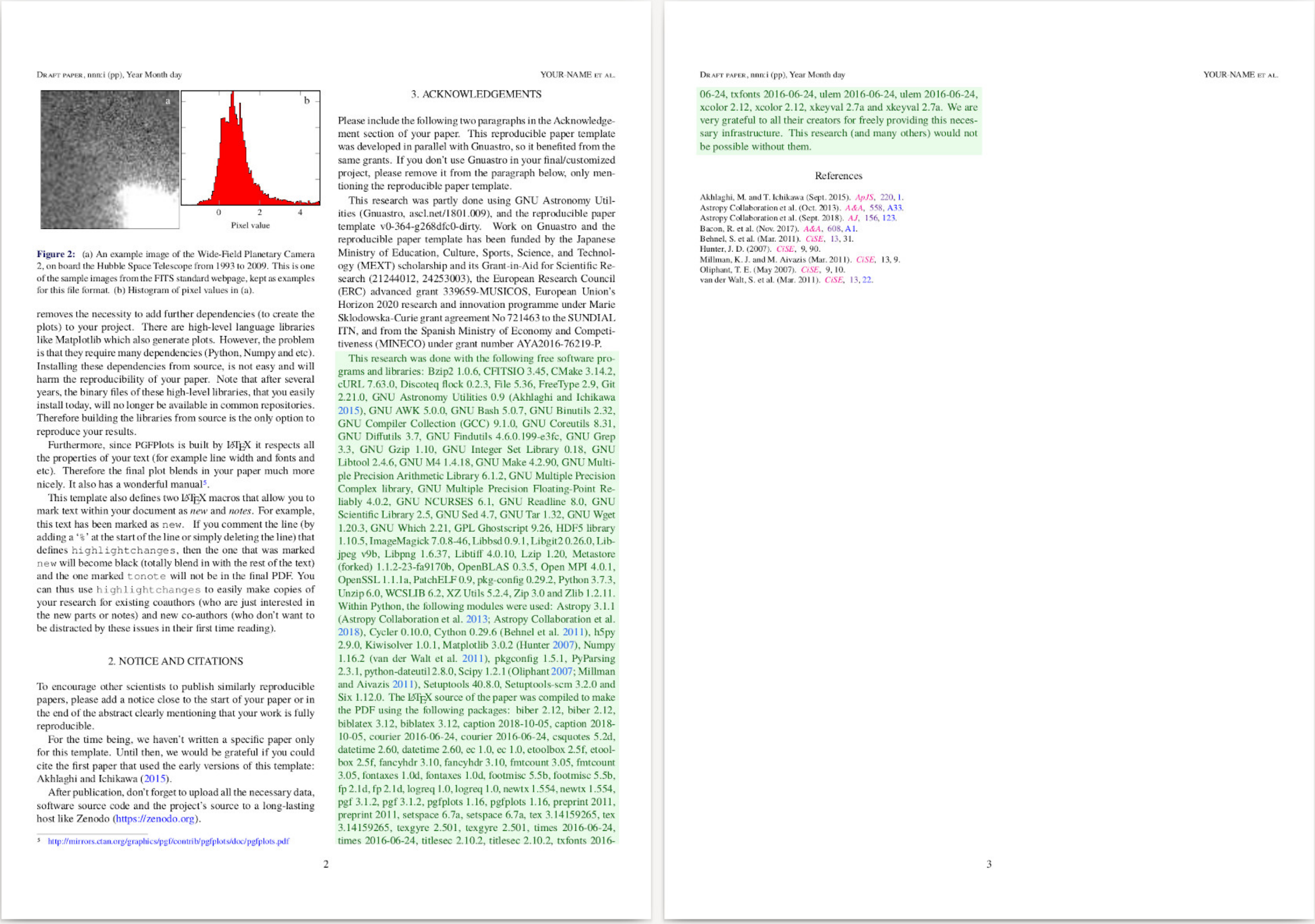
From "Attributing and Referencing (Research) Software: Best Practices and Outlook from Inria" (Alliez et al. 2020, CiSE, DOI:10.1109/MCSE.2019.2949413).

Advantages of this build system

- Project runs in fixed/controlled environment: custom build of Bash, Make, GNU Coreutils (`ls`, `cp`, `mkdir` and etc), AWK, or SED, $\LaTeX$, etc.
- No need for root/administrator permissions (on servers or super computers).
- Whole system is built automatically on any Unix-like operating system (less 2 hours).
- Dependencies of different projects will not conflict.
- Everything in plain text (human & computer readable/archivable).

https://natemowry2.wordpress.com

Software citation automatically generated in paper (including Astropy)



a b

0 2 4

Pixel value

Figure 2: (a) An example image of the Wide-Field Planetary Camera 2, on board the Hubble Space Telescope from 1993 to 2009. This is one of the sample images from the FITS standard webpage, kept as examples for this file format. (b) Histogram of pixel values in (a).

removes the necessity to add further dependencies (to create the plots) to your project. There are high-level language libraries like Matplotlib which also generate plots. However, the problem is that they require many dependencies (Python, Numpy and etc). Installing these dependencies from source, is not easy and will harm the reproducibility of your paper. Note that after several years, the binary files of these high-level libraries, that you easily install today, will no longer be available in common repositories. Therefore building the libraries from source is the only option to reproduce your results.

Furthermore, since PGFPlots is built by LaTeX it respects all the properties of your text (for example line width and fonts and etc). Therefore the final plot blends in your paper much more nicely. It also has a wonderful manual[5].

This template also defines two LaTeX macros that allow you to mark text within your document as *new* and *notes*. For example, this text has been marked as `new`. If you comment the line (by adding a '%' at the start of the line or simply deleting the line) that defines `highlightchanges`, then the one that was marked `new` will become black (totally blend in with the rest of the text) and the one marked `tonote` will not be in the final PDF. You can thus use `highlightchanges` to easily make copies of your research for existing coauthors (who are just interested in the new parts or notes) and new co-authors (who don't want to be distracted by these issues in their first time reading).

2. NOTICE AND CITATIONS

To encourage other scientists to publish similarly reproducible papers, please add a notice close to the start of your paper or in the end of the abstract clearly mentioning that your work is fully reproducible.

For the time being, we haven't written a specific paper only for this template. Until then, we would be grateful if you could cite the first paper that used the early versions of this template: Akhlaghi and Ichikawa (2015).

After publication, don't forget to upload all the necessary data, software source code and the project's source to a long-lasting host like Zenodo (https://zenodo.org).

[5] http://mirrors.ctan.org/graphics/pgf/contrib/pgfplots/doc/pgfplots.pdf

3. ACKNOWLEDGEMENTS

Please include the following two paragraphs in the Acknowledgement section of your paper. This reproducible paper template was developed in parallel with Gnuastro, so it benefited from the same grants. If you don't use Gnuastro in your final/customized project, please remove it from the paragraph below, only mentioning the reproducible paper template.

This research was partly done using GNU Astronomy Utilities (Gnuastro, ascl.net/1801.009), and the reproducible paper template v0-364-g268dfc0-dirty. Work on Gnuastro and the reproducible paper template has been funded by the Japanese Ministry of Education, Culture, Sports, Science, and Technology (MEXT) scholarship and its Grant-in-Aid for Scientific Research (21244012, 24253003), the European Research Council (ERC) advanced grant 339659-MUSICOS, European Union's Horizon 2020 research and innovation programme under Marie Sklodowska-Curie grant agreement No 721463 to the SUNDIAL ITN, and from the Spanish Ministry of Economy and Competitiveness (MINECO) under grant number AYA2016-76219-P.

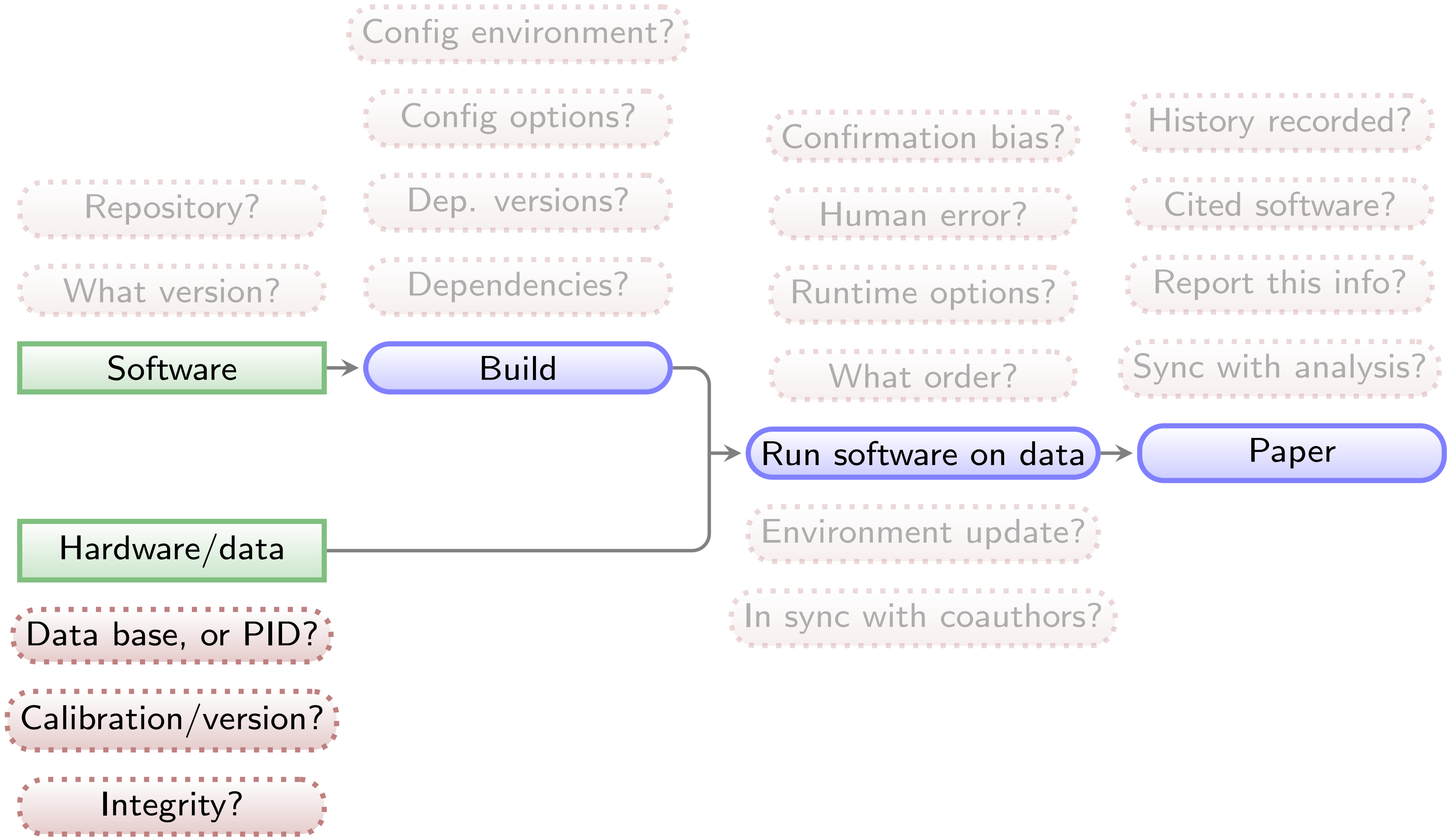
This research was done with the following free software programs and libraries: Bzip2 1.0.6, CFITSIO 3.45, CMake 3.14.2, cURL 7.63.0, Discoteq flock 0.2.3, File 5.36, FreeType 2.9, Git 2.21.0, GNU Astronomy Utilities 0.9 (Akhlaghi and Ichikawa 2015), GNU AWK 5.0.0, GNU Bash 5.0.7, GNU Binutils 2.32, GNU Compiler Collection (GCC) 9.1.0, GNU Coreutils 8.31, GNU Diffutils 3.7, GNU Findutils 4.6.0.199-e3fc, GNU Grep 3.3, GNU Gzip 1.10, GNU Integer Set Library 0.18, GNU Libtool 2.4.6, GNU M4 1.4.18, GNU Make 4.2.90, GNU Multiple Precision Arithmetic Library 6.1.2, GNU Multiple Precision Complex library, GNU Multiple Precision Floating-Point Reliably 4.0.2, GNU NCURSES 6.1, GNU Readline 8.0, GNU Scientific Library 2.5, GNU Sed 4.7, GNU Tar 1.32, GNU Wget 1.20.3, GNU Which 2.21, GPL Ghostscript 9.26, HDF5 library 1.10.5, ImageMagick 7.0.8-46, Libbsd 0.9.1, Libgit2 0.26.0, Libjpeg v9b, Libpng 1.6.37, Libtiff 4.0.10, Lzip 1.20, Metastore (forked) 1.1.2-23-fa9170b, OpenBLAS 0.3.5, Open MPI 4.0.1, OpenSSL 1.1.1a, PatchELF 0.9, pkg-config 0.29.2, Python 3.7.3, Unzip 6.0, WCSLIB 6.2, XZ Utils 5.2.4, Zip 3.0 and Zlib 1.2.11. Within Python, the following modules were used: Astropy 3.1.1 (Astropy Collaboration et al. 2013; Astropy Collaboration et al. 2018), Cycler 0.10.0, Cython 0.29.6 (Behnel et al. 2011), h5py 2.9.0, Kiwisolver 1.0.1, Matplotlib 3.0.2 (Hunter 2007), Numpy 1.16.2 (van der Walt et al. 2011), pkgconfig 1.5.1, PyParsing 2.3.1, python-dateutil 2.8.0, Scipy 1.2.1 (Oliphant 2007; Millman and Aivazis 2011), Setuptools 40.8.0, Setuptools-scm 3.2.0 and Six 1.12.0. The LaTeX source of the paper was compiled to make the PDF using the following packages: biber 2.12, biber 2.12, biblatex 3.12, biblatex 3.12, caption 2018-10-05, caption 2018-10-05, courier 2016-06-24, courier 2016-06-24, csquotes 5.2d, datetime 2.60, datetime 2.60, ec 1.0, ec 1.0, etoolbox 2.5f, etoolbox 2.5f, fancyhdr 3.10, fancyhdr 3.10, fmtcount 3.05, fmtcount 3.05, fontaxes 1.0d, fontaxes 1.0d, footmisc 5.5b, footmisc 5.5b, fp 2.1d, fp 2.1d, logreq 1.0, logreq 1.0, newtx 1.554, newtx 1.554, pgf 3.1.2, pgf 3.1.2, pgfplots 1.16, pgfplots 1.16, preprint 2011, preprint 2011, setspace 6.7a, setspace 6.7a, tex 3.14159265, tex 3.14159265, texgyre 2.501, texgyre 2.501, times 2016-06-24, times 2016-06-24, titlesec 2.10.2, titlesec 2.10.2, txfonts 2016-

06-24, txfonts 2016-06-24, ulem 2016-06-24, ulem 2016-06-24, xcolor 2.12, xcolor 2.12, xkeyval 2.7a and xkeyval 2.7a. We are very grateful to all their creators for freely providing this necessary infrastructure. This research (and many others) would not be possible without them.

References

Akhlaghi, M. and T. Ichikawa (Sept. 2015). *ApJS*, 220, 1.
Astropy Collaboration et al. (Oct. 2013). *A&A*, 558, A33.
Astropy Collaboration et al. (Sept. 2018). *AJ*, 156, 123.
Bacon, R. et al. (Nov. 2017). *A&A*, 608, A1.
Behnel, S. et al. (Mar. 2011). *CiSE*, 13, 31.
Hunter, J. D. (2007). *CiSE*, 9, 90.
Millman, K. J. and M. Aivazis (Mar. 2011). *CiSE*, 13, 9.
Oliphant, T. E. (May 2007). *CiSE*, 9, 10.
van der Walt, S. et al. (Mar. 2011). *CiSE*, 13, 22.

General outline of a project (after data collection)



Green boxes with sharp corners: *source*/input components/files.
Blue boxes with rounded corners: *built* components.
Red boxes with dashed borders: questions that must be clarified for each phase.

Input data source and integrity is documented and checked

Stored information about each input file:

- PID (where available).
- Download URL.
- MD5-sum to check integrity.

All inputs are downloaded from the given PID/URL when necessary (during the analysis).

MD5-sums are checked to make sure the download was done properly or the file is the same (hasn't changed on the server/source).

Example from the reproducible paper arXiv:1909.11230.
This paper needs three input files (two images, one catalog).

emacs@akhlaghi

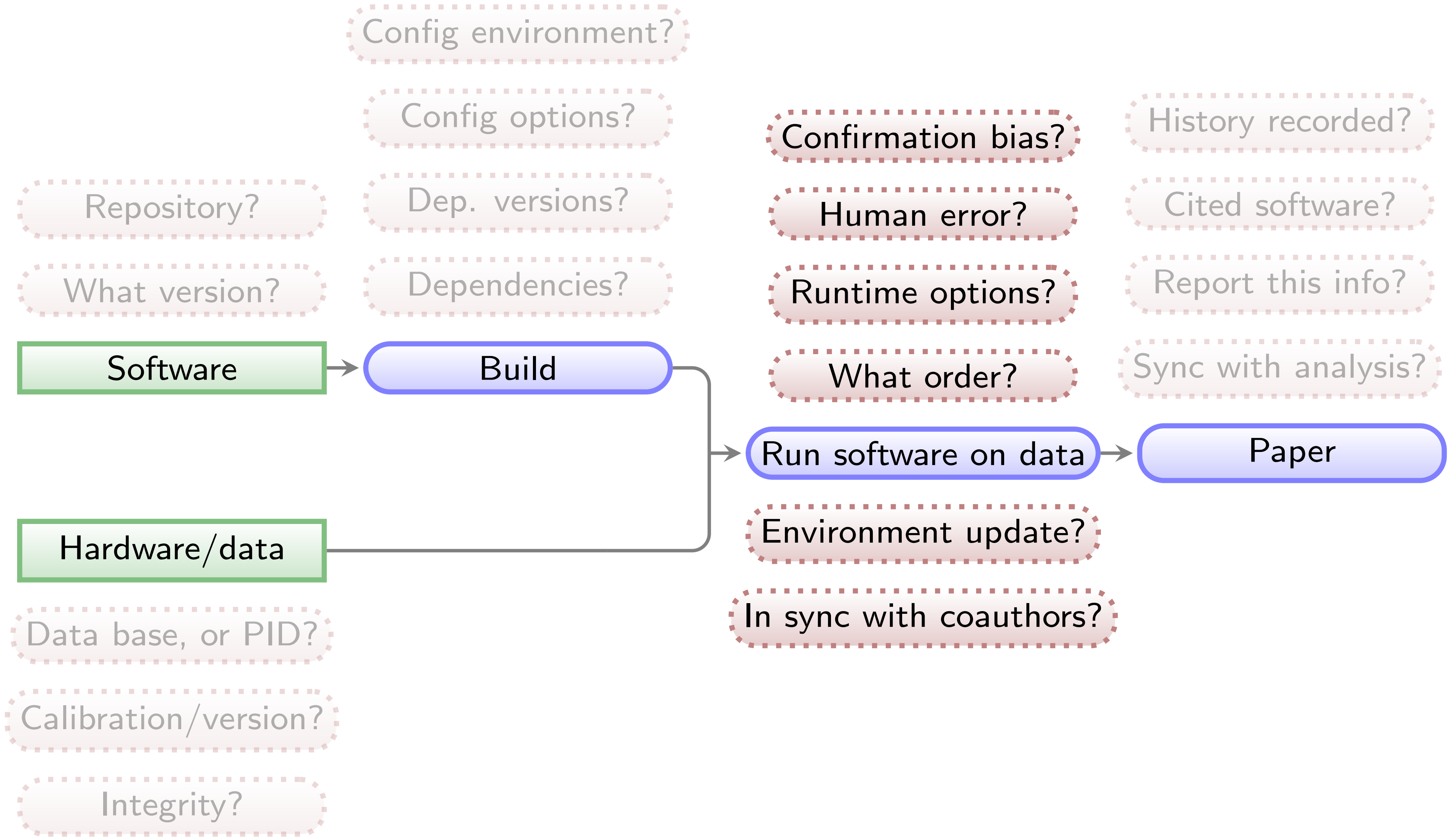
File Edit Options Buffers Tools Makefile Help

```
# Input files necessary for this project.
#
# This file is read by the configure script and running Makefiles.
#
# Copyright (C) 2018-2019 Mohammad Akhlaghi <mohammad@akhlaghi.org>
#
# Copying and distribution of this file, with or without modification, are
# permitted in any medium without royalty provided the copyright notice and
# this notice are preserved.  This file is offered as-is, without any
# warranty.


M51SDSSRURL   = https://dr12.sdss.org/sas/dr12/boss/photoObj/frames/301/3716/6
M51SDSSRIMAGE = frame-r-003716-6-0117.fits.bz2
M51SDSSRMD5   = 965da8bd861e94a9701521a11b2d80aa
M51SDSSRSIZE  = 2.8M

XDFF755WURL   = http://archive.stsci.edu/pub/hlsp/xdf
XDFF755WIMAGE = hlsp_xdf_hst_acswfc-60mas_hudf_f775w_v1_sci.fits
XDFF755WMD5   = 81408ed0949bd3a93c4bfe7e229472e6
XDFF775WSIZE  = 106M

UVUDFSEGURL   = https://asd.gsfc.nasa.gov/UVUDF
UVUDFSEGIMAGE = segmentation_map_rafelski_2015.fits.gz
UVUDFSEGMD5   = 29d5b3e5311b77512baf27db6ad0e11b
UVUDFSEGSIZE  = 1.3M
```

-:--- INPUTS.mk   All L1   Git-master (GNUmakefile)

For information about GNU Emacs and the GNU system, type C-h C-a.

General outline of a project (after data collection)



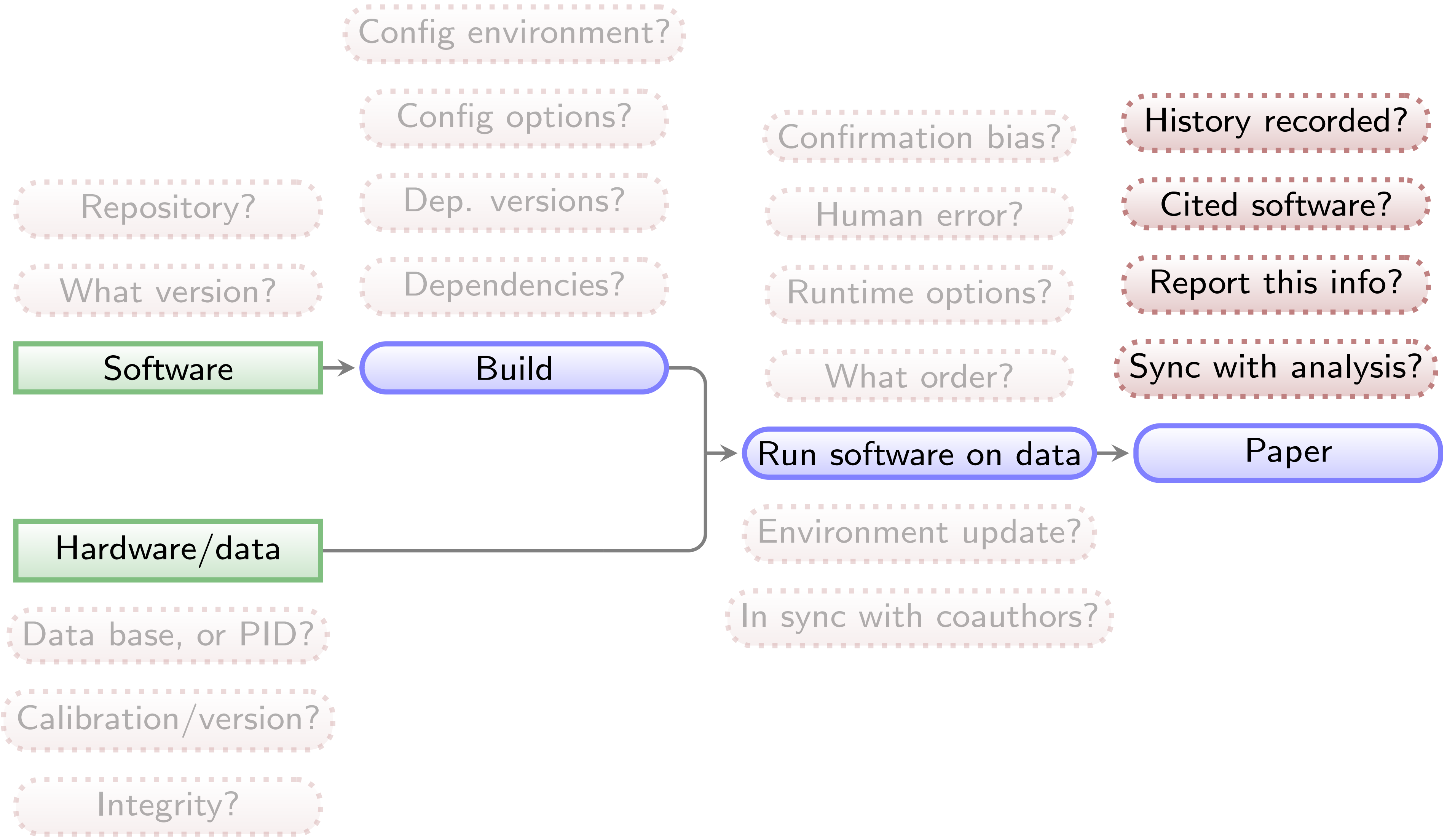
Green boxes with sharp corners: *source*/input components/files.
Blue boxes with rounded corners: *built* components.
Red boxes with dashed borders: questions that must be clarified for each phase.

Reproducible science: Maneage is managed through a Makefile

All steps (downloading and analysis) are managed by Makefiles (example from zenodo.1164774):

- Unlike a script which always starts from the top, a Makefile starts from the end and steps that don't change will be left untouched (not remade).
- A single *rule* can manage any number of files.
- Make can identify independent steps internally and do them in parallel.
- Make was designed for complex projects with thousands of files (all major Unix-like components), so it is highly evolved and efficient.
- Make is a very simple and small language, thus easy to learn with great and free documentation (for example GNU Make's manual).

General outline of a project (after data collection)



Green boxes with sharp corners: *source*/input components/files.
Blue boxes with rounded corners: *built* components.
Red boxes with dashed borders: questions that must be clarified for each phase.

Values in final report/paper

All analysis results (numbers, plots, tables) written in paper's PDF as LaTeX macros. They are thus updated automatically on any change.
Shown here is a portion of the NoiseChisel paper and its LaTeX source (arXiv:1505.01664).

```latex
\begin{equation}
  \label{tSNeq}
  \mathrm{S/N}_T=\frac{NF-NS_a}{\sqrt{NF+N\sigma_s^2}}
  =\frac{\sqrt{N}(F-S_a)}{\sqrt{F+\sigma_s^2}}.
\end{equation}

\noindent
See Section \ref{SNeqmodif} for the modifications required when
the input image is not in units of counts or has already been
Sky subtracted. The distribution of {\small S/N}$_T$ from the
objects in $R_s$ for the three examples in Figure \ref{dettf}
can be seen in column 5 (top) of that figure. Image processing
effects, mainly due to shifting, rotating, and re-sampling the
images for co-adding, on the real data further increase the size
and count, and hence, the {\small S/N} of false detections in
real, reduced/co-added images. A comparison of scales on the
{\small S/N} histograms between the mock ((a.5.1) and (b.5.1))
and real (c.5.1) examples in Figure \ref{dettf} shows the effect
quantitatively. In the histograms of Figure \ref{dettf}, the bin
with the largest number of false pseudo-detections respectively
has an {\small S/N} of $\onelargedettfmax$,
$\sensitivitycdettfmax$, and $\fourdettfmax$.
```

smaller than `--detsnminarea` are removed from the analysis in both R_s and R_d. In the examples in this section, it is set to `15`. Note that since a threshold approximately equal to the Sky value is used, this is a very weak constraint. For each pseudo-detection, $\mathrm{S/N}_T$ can be written as,

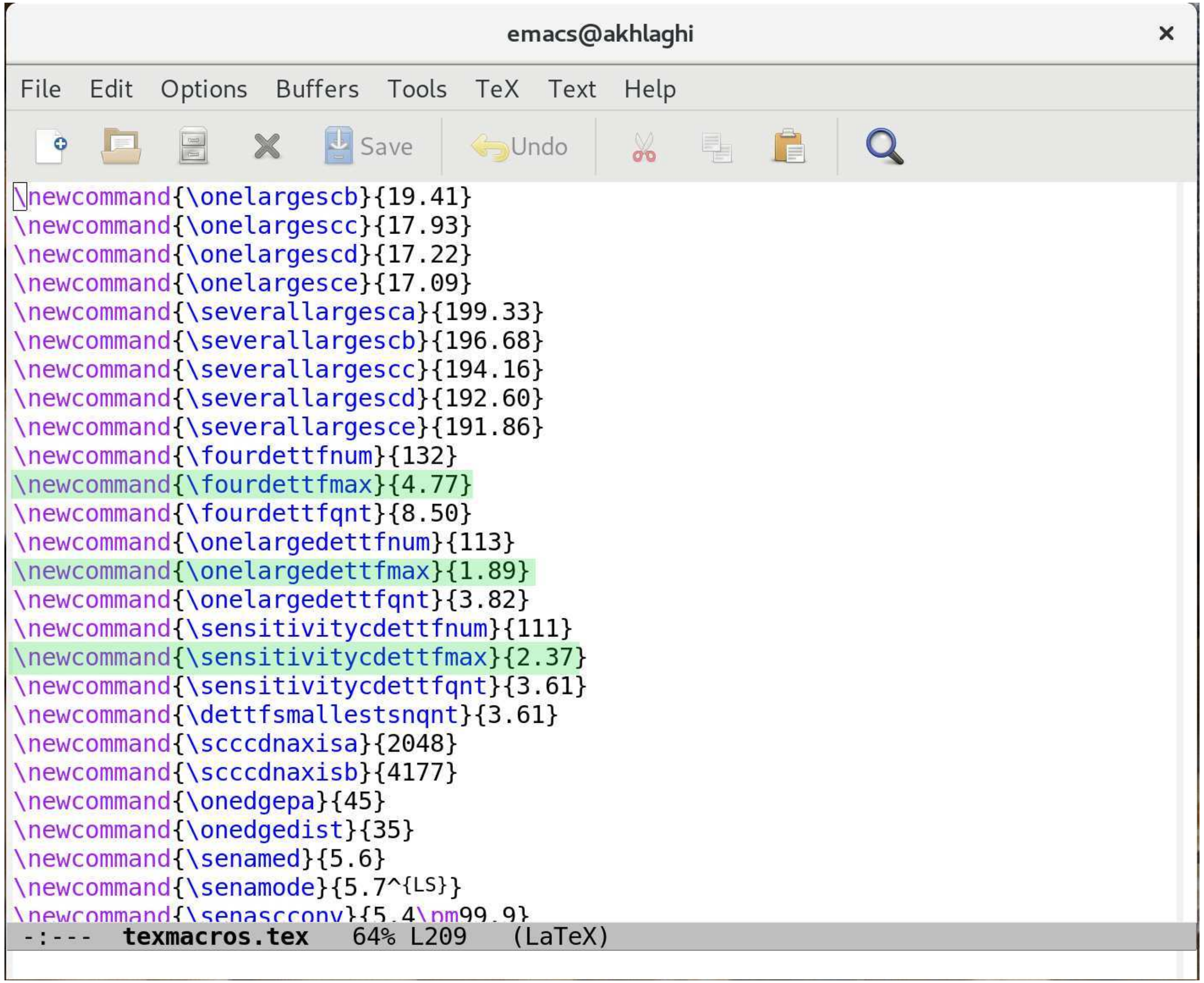
$$\mathrm{S/N}_T = \frac{NF - NS_a}{\sqrt{NF + N\sigma_S^2}} = \frac{\sqrt{N}(F - S_a)}{\sqrt{F + \sigma_S^2}}. \qquad (3)$$

See Section 3.3 for the modifications required when the input image is not in units of counts or has already been Sky subtracted. The distribution of $\mathrm{S/N}_T$ from the objects in R_s for the three examples in Figure 7 can be seen in column 5 (top) of that figure. Image processing effects, mainly due to shifting, rotating, and re-sampling the images for co-adding, on the real data further increase the size and count, and hence, the S/N of false detections in real, reduced/co-added images. A comparison of scales on the S/N histograms between the mock ((a.5.1) and (b.5.1)) and real (c.5.1) examples in Figure 7 shows the effect quantitatively. In the histograms of Figure 7, the bin with the largest number of false pseudo-detections respectively has an S/N of 1.89, 2.37, and 4.77.

The $\mathrm{S/N}_T$ distribution of detections in R_s provides a very ro-

Analysis step results/values concatenated into a single file.

All LaTeX macros come from a single file.



Analysis results stored as LaTeX macros

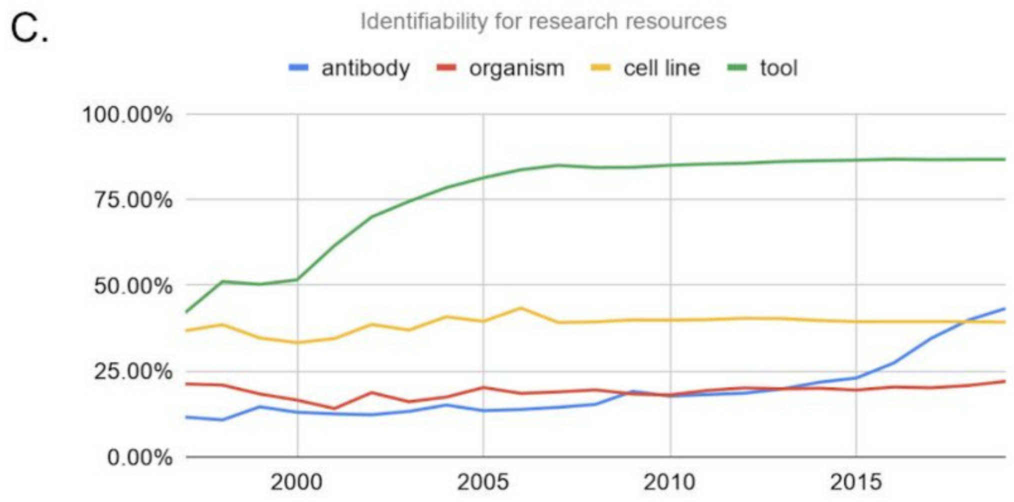
The analysis scripts write/update the LaTeX macro values automatically.

```
# Numbers for dettf.tex:
sqnt=9999999
function dettfhist
{
    # Set the file name.
    if [ $2 == 4 ]; then                    obase=four;
    elif [ $2 = sensitivity3 ]; then        obase=sensitivityc;
    else                                    obase=$2;
    fi
    if [ $2 == onelarge ]; then ind="_7"; else ind="_12"; fi
    name=$1$2$ind"_detsn"$txt

    dettfnum=$(awk '/points binned in/{print $4; exit(0)}' $name)
    dettfqnt=$(awk '/quantile has a value of/{
                       printf("%.2f", $9); exit(0);}' $name)
    dettfmax=$(awk 'BEGIN { max=-999999 }
                    !/^#/ { if($2>max){max=$2; mv=$1} }
                    END { printf("%.2f", mv) }' $name)
    addtexmacro $obase"dettfnum" $dettfnum
    addtexmacro $obase"dettfmax" $dettfmax
    addtexmacro $obase"dettfqnt" $dettfqnt

    # Find the smallest S/N quantile:
    sqnt=$(echo " " | awk '{if('$dettfqnt'<'$sqnt') print '$dettfqnt'}')
}
for base in 4 onelarge sensitivity3
do dettfhist $texdir/dettf/ $base;  done
addtexmacro dettfsmallestsnqnt $sqnt
```

Let's look at the data lineage to replicate Figure 1C (green/tool) of Menke+2020 (DOI:10.1101/2020.01.15.908111), as done in arXiv:2006.03018 for a demo.

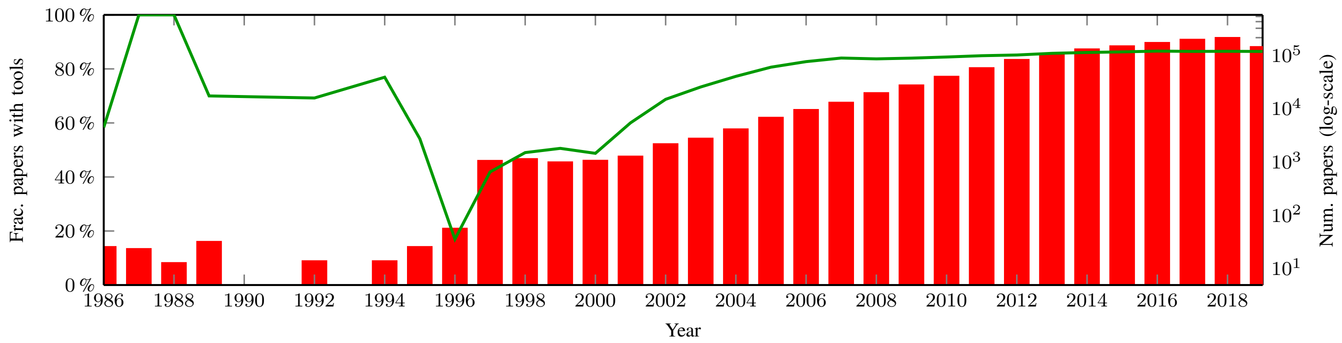
ORIGINAL PLOT

The Green plot shows the fraction of papers mentioning software tools from 1997 to 2019.

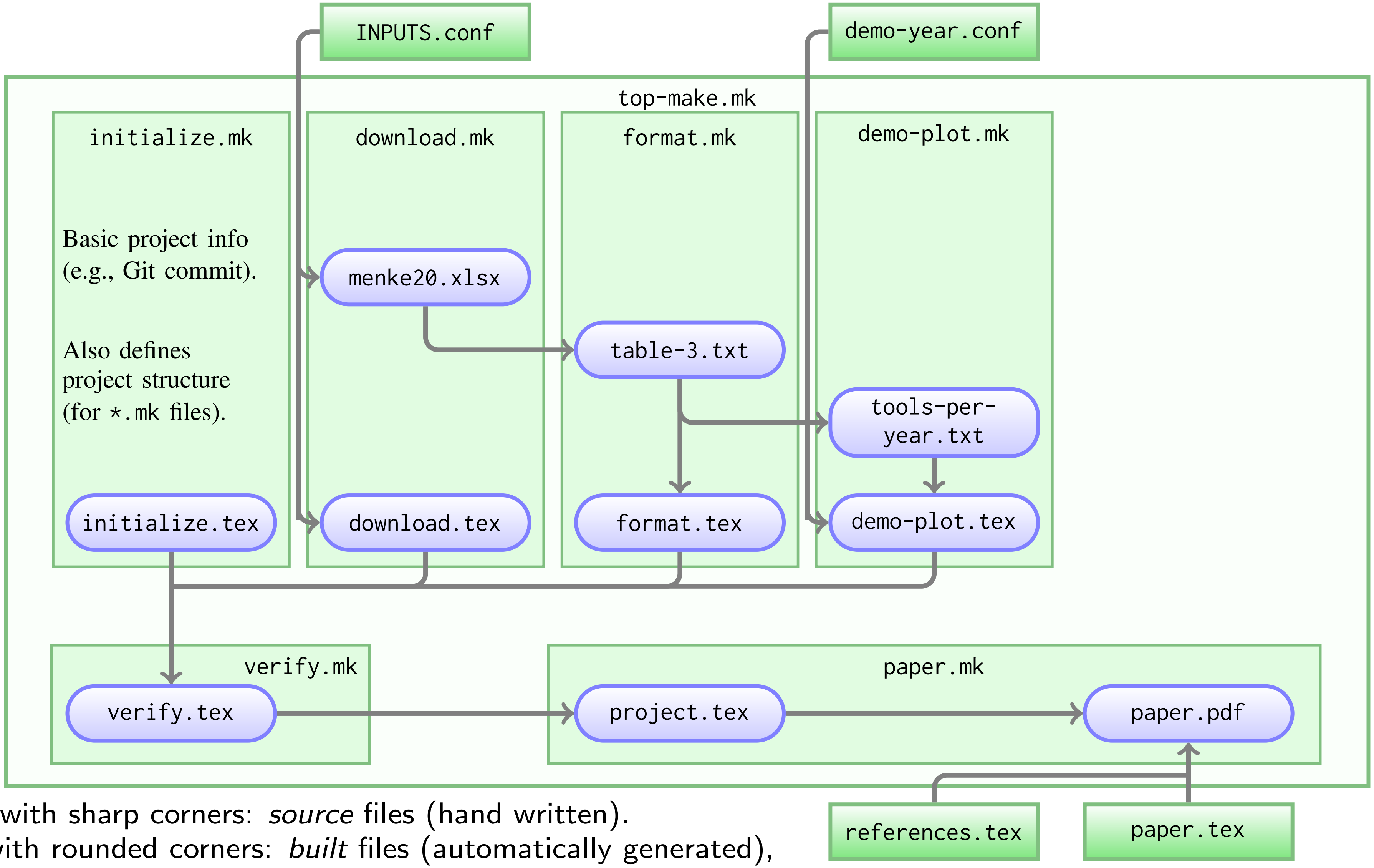


OUR enhanced REPLICATION

The green line is same as above but over their full historical range.
Red histogram is the number of papers studied in each year

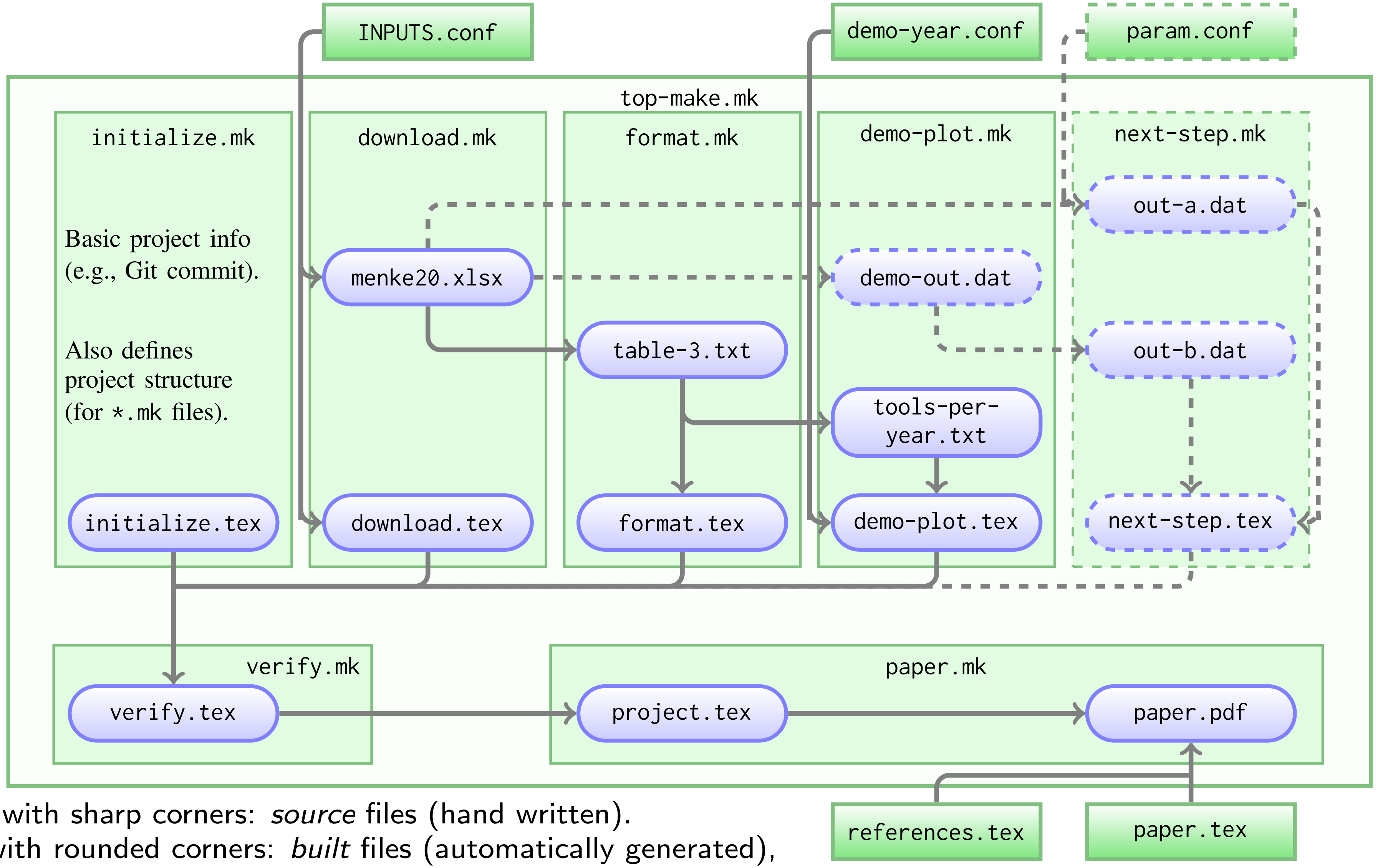


All analysis steps cascade down to paper.pdf (URL and checksum of input in `INPUTS.conf`).



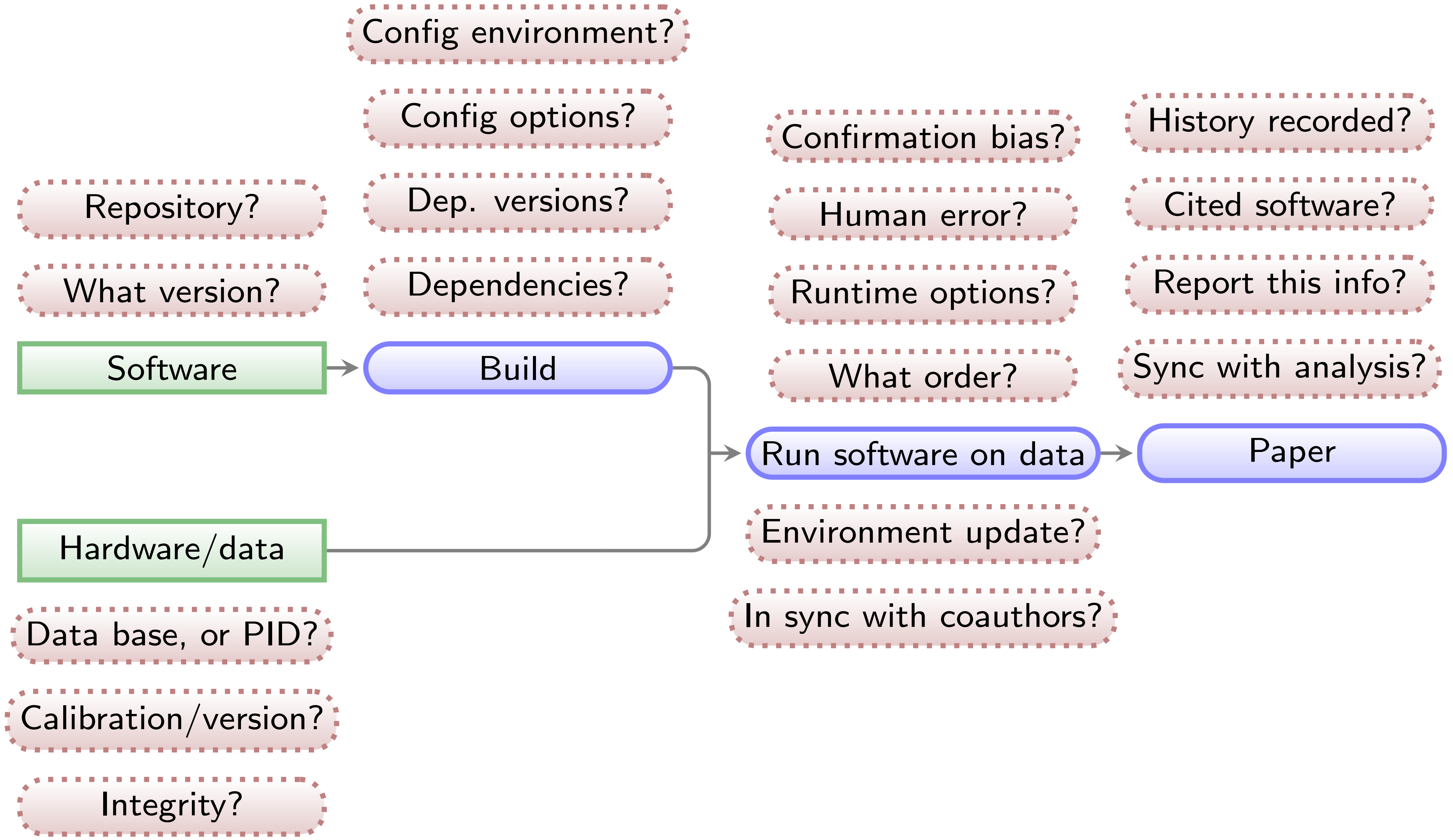
Green boxes with sharp corners: *source* files (hand written).
Blue boxes with rounded corners: *built* files (automatically generated),
built files are shown in the Makefile that contains their build instructions.

It is very easy to expand the project and add new analysis steps (this solution is scalable)



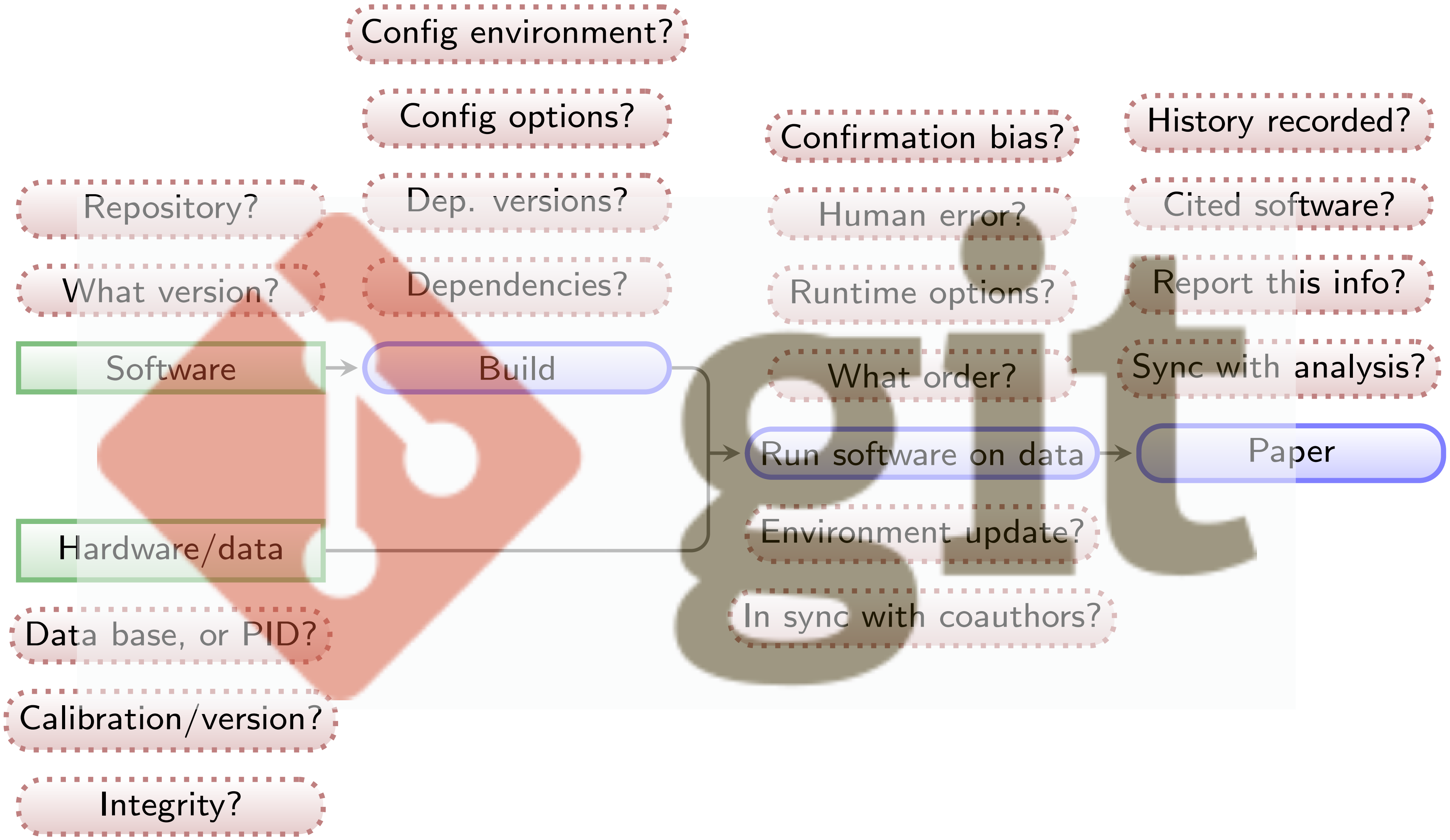
Green boxes with sharp corners: *source* files (hand written).
Blue boxes with rounded corners: *built* files (automatically generated),
built files are shown in the Makefile that contains their build instructions.

All questions have an answer now (in plain text: human & computer readable/archivable).



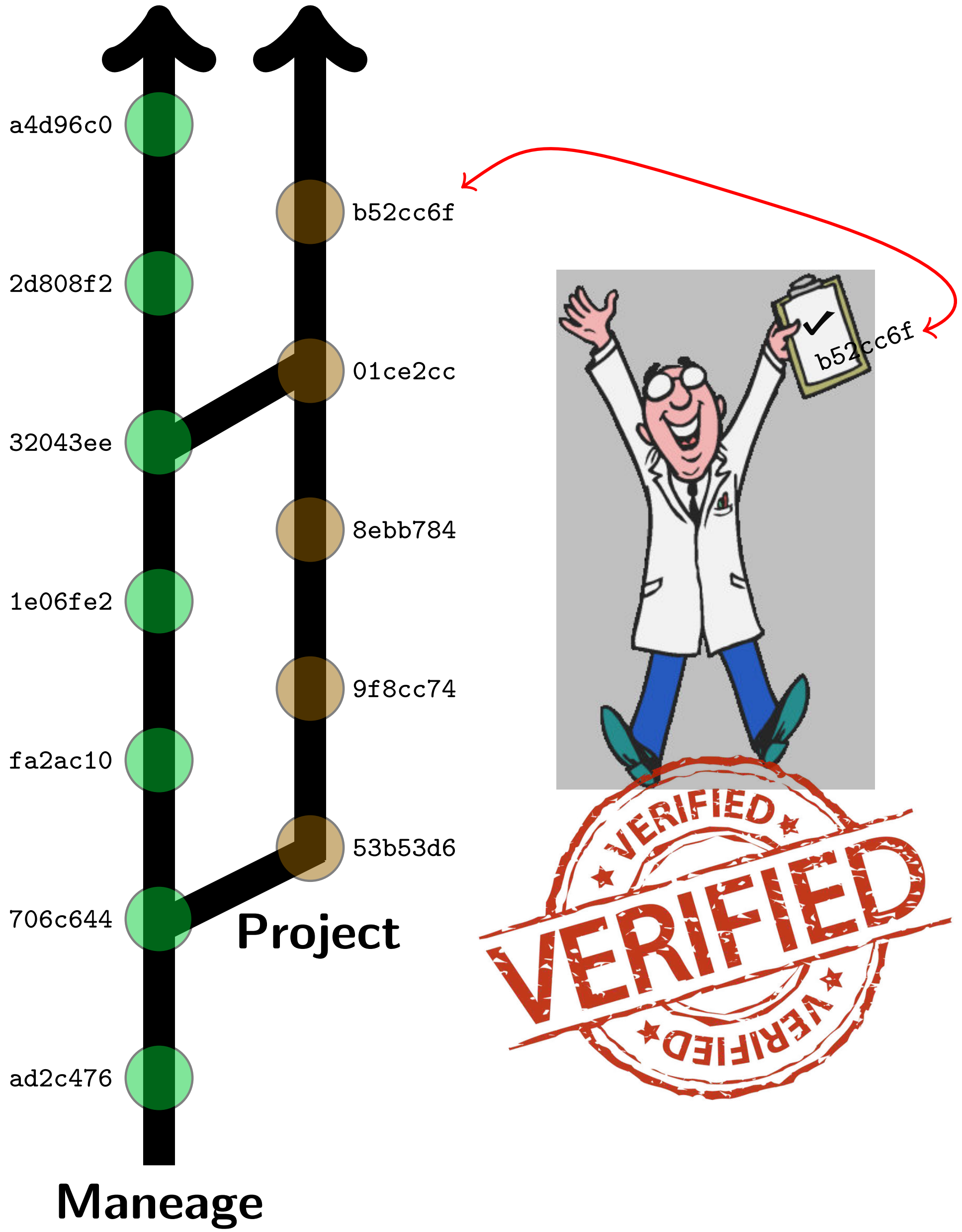
Green boxes with sharp corners: *source*/input components/files.
Blue boxes with rounded corners: *built* components.
Red boxes with dashed borders: questions that must be clarified for each phase.

All questions have an answer now (in plain text: so we can use Git to keep its history).



Green boxes with sharp corners: *source*/input components/files.
Blue boxes with rounded corners: *built* components.
Red boxes with dashed borders: questions that must be clarified for each phase.

New projects branch from Maneage



- Template's history is recorded in Git.
- New project: a branch from the template.
  Recall that every commit contains the following:
  - Instructions to download, verify and build software.
  - Instructions to download and verify input data.
  - Instructions to run software on data (do the analysis).
  - Narrative description of project's purpose/context.
- Research progresses in the project branch.
- Template will evolve (improved infrastructure).
- Template can be imported/merged back into project.
- The template and project will evolve.
- During research this encourages creative tests (previous research states can easily be retrieved).
- Coauthors can work on same project in parallel (separate project branches).
- Upon publication, the Git checksum is enough to verify the integrity of the result.

"Verified" image from vectorstock.com

Two recent examples (publishing Git checksum in abstract)

The Realm of the Low-Surface-Brightness Universe
Proceedings IAU Symposium No. 355, 2019
D. Valls-Gabaud, I. Trujillo & S. Okamoto, eds.

DOI: 00.0000/X000000000000000X

Carving out the low surface brightness universe with NoiseChisel

Mohammad Akhlaghi[1,2]

[1]Instituto de Astrofísica de Canarias, C/ Vía Láctea, 38200 La Laguna, Tenerife, Spain.
email: mohammad@akhlaghi.org

[2]Facultad de Física, Universidad de La Laguna, Avda. Astrofísico Fco. Sánchez s/n, 38200 La Laguna, Tenerife, Spain.

Abstract. NoiseChisel is a program to detect very low signal-to-noise ratio (S/N) features with minimal assumptions on their morphology. It was introduced in 2015 and released within a collection of data analysis programs and libraries known as GNU Astronomy Utilities (Gnuastro). Over the last ten stable releases of Gnuastro, NoiseChisel has significantly improved: detecting even fainter signal, enabling better user control over its inner workings, and many bug fixes. The most important change may be that NoiseChisel's segmentation features have been moved into a new program called Segment. Another major change is the final growth strategy of its true detections, for example NoiseChisel is able to detect the outer wings of M51 down to S/N of 0.25, or 28.27 mag/arcsec^2 on a single-exposure SDSS image (r-band). Segment is also able to detect the localized HII regions as "clumps" much more successfully. Finally, to orchestrate a controlled analysis, the concept of a "reproducible paper" is discussed: this paper itself is exactly reproducible (snapshot v4-0-g8505cfd).

VERIFIED

Keywords. galaxies: halos, galaxies: photometry, galaxies: structure, methods: data analysis, methods: reproducible, techniques: image processing, techniques: photometric

1. Introduction

Signal from the low surface brightness universe is buried deep in the datasets noise and thus requires accurate detection methods. In Akhlaghi and Ichikawa (2015) (henceforth AI15) a new method was introduced to detect such very low signal-to-noise ratio (S/N) signal from the images in a non-parametric manner. It allows accurate detection of the diffuse outer features of galaxies (that often have a different morphology from the centers). The software implementation of this method (NoiseChisel) is released as part of a larger collection of data analysis software known as GNU Astronomy Utilities† (Gnuastro). It was the first professional astronomical software to be independently refereed by an independent panel (GNU Evaluation committee) and fully conforms with the GNU Coding Standards‡.

Since its release, NoiseChisel has been used in many studies. For example Bacon et al. (2017) used it to identify objects that were missed by Rafelski et al. (2015) (henceforth R15), who used a combination of six SExtractor (Bertin and Arnouts 1996) runs with different configurations to avoid deblending problems, but still missed many sources with significant signal, see Figure 1. Borlaff et al. (2019), Mller et al. (2019), and Trujillo et al. (2019) used it for accurate flat field and Sky subtraction to create deeper co-added images in galaxy fields for optimal detection of the low surface brightness features. Calvi et al. (2019) used it to find Lyman-α emitters in spectra. For future studies, Laine et al.

† https://www.gnu.org/s/gnuastro
‡ https://www.gnu.org/prep/standards

Monthly Notices of the ROYAL ASTRONOMICAL SOCIETY

MNRAS **491,** 5317–5329 (2020)
Advance Access publication 2019 November 14

doi:10.1093/mnras/stz3111

The Sloan Digital Sky Survey extended point spread functions

Raúl Infante-Sainz[1,2★], Ignacio Trujillo[1,2] and Javier Román[1,2,3]

[1]Instituto de Astrofísica de Canarias, c/ Vía Láctea s/n, E-38205 La Laguna, Tenerife, Spain
[2]Departamento de Astrofísica, Universidad de La Laguna, E-38205 La Laguna, Tenerife, Spain
[3]Instituto de Astrofísica de Andalucía (CSIC), Glorieta de la Astronomía, E-18008 Granada, Spain

Accepted 2019 October 30. Received 2019 October 29; in original form 2019 September 10

ABSTRACT

A robust and extended characterization of the point spread function (PSF) is crucial to extract the photometric information produced by deep imaging surveys. Here, we present the extended PSFs of the Sloan Digital Sky Survey (SDSS), one of the most productive astronomical surveys of all time. By stacking ~1000 images of individual stars with different brightness, we obtain the bidimensional SDSS PSFs extending over 8 arcmin in radius for all the SDSS filters (u, g, r, i, z). This new characterization of the SDSS PSFs is near a factor of 10 larger in extension than previous PSFs characterizations of the same survey. We found asymmetries in the shape of the PSFs caused by the drift scanning observing mode. The flux of the PSFs is larger along the drift scanning direction. Finally, we illustrate with an example how the PSF models can be used to remove the scattered light field produced by the brightest stars in the central region of the Coma cluster field. This particular example shows the huge importance of PSFs in the study of the low-surface brightness Universe, especially with the upcoming of ultradeep surveys, such as the Large Synoptic Survey Telescope (LSST). Following a reproducible science philosophy, we make all the PSF models and the scripts used to do the analysis of this paper publicly available (snapshot v0.4-0-gd966ad0).

Key words: instrumentation: detectors – methods: data analysis – techniques: image processing – techniques: photometric – galaxies: haloes.

VERIFIED

1 INTRODUCTION

The point spread function (PSF) describes the response of an imaging system to the light produced by a point source. Real PSFs have complex structures as their shapes depend on the optical path that light takes as it travels through the atmosphere and multiple optical elements, mirrors, lenses, detectors, etc. For the vast majority of astronomical works, only a tiny portion of the PSF (i.e. normally a few inner arcseconds; see e.g. Trujillo et al. 2001a, b) is characterized. In practice, however, the light of both point and extended sources are spread over the entire detector due to the effect of the PSF at large radii. Therefore, it is necessary to have a good understanding of its structure along the entire detector (typically extending over arcminutes or more).

Extended PSFs have become a vital tool to obtain precise photometric information in modern astronomical surveys. For instance, Slater, Harding & Mihos (2009) modelled the extended PSF and the internal reflections produced by the stars of the Burrell Schmidt telescope and showed that virtually all the pixels of the image are dominated by the scattered light by both stars and galaxies at 29.5 mag arcsec^{-2} (V-band). Trujillo & Fliri (2016) also characterized and used the extended PSF of the 10.4 m Gran Telescopio Canarias (GTC) telescope to model and remove the scattered light in ultradeep observations of the UGC 00180 galaxy. Even more troublesome for low-surface brightness studies is the finding (see e.g. Trujillo & Bakos 2013; Sandin 2014, 2015) that the outer regions of astronomical objects are severely affected by their own scattered light produced by the convolution with the PSF. In order to correct this effect, Karabal et al. (2017) generated the PSF and models of the internal reflections from images of the Canada–France–Hawaii Telescope (CFHT) to de-convolve a sample of three galaxies and correct them from instrumental scattered light. More recently, Román, Trujillo & Montes (2019) characterized the PSFs of the Stripe 82 survey and used them to model and correct the scattered light field produced by stars to study the optical properties of the Galactic cirri. All the above works have shown that having an extended PSF is crucial when accurate photometric and structure properties of astronomical objects at low-surface brightness levels are required.

One of the most commonly used surveys for measuring photometric properties of astronomical objects is the Sloan Sky Digital Survey (SDSS; York et al. 2000), covering 14 555 deg^2 on the sky (just over 35 per cent of the full sky) in five photometric bands (u, g, r, i, and z). Although SDSS is a relatively shallow survey compared

★ E-mail: infantesainz@gmail.com

Publication of the project

A reproducible project using Maneage will have the following (plain text) components:

- Makefiles.
- LaTeX source files.
- Configuration files for software used in analysis.
- Scripts/programming files (e.g., Python, Shell, AWK, C).

The volume of the project's source will thus be negligible compared to a single figure in a paper (usually ~ 100 kilo-bytes).

The project's pipeline (customized Maneage) can be published in

- arXiv: uploaded with the LaTeX source to always stay with the paper (for example arXiv:1505.01664 or arXiv:2006.03018).
- Zenodo: Along with all the input datasets (many Gigabytes) and software (for example zenodo.3872248) and given a unique DOI.

General outline of using Maneage (for example arXiv:2006.03018)

```
$ git clone https://gitlab.com/makhlaghi/maneage-paper    # Import the project.

$ ./project configure          # You will specify the build directory on your system,
                               # and it will build all software (about 1.5 hours).

$ ./project make               # Does all the analysis and makes final PDF.
```

Future prospects...

Adoption of reproducibility by many researchers will enable the following:

- A repository for education/training (PhD students, or researchers in other fields).
- Easy verification/understanding of other research projects (when necessary).
- Trivially test different steps of others' work (different configurations, software and etc).
- Science can progress incrementally (shorter papers actually building on each other!).
- Extract meta-data after the publication of a dataset (for future ontologies or vocabularies).
- Applying machine learning on reproducible research projects will allow us to solve some Big Data Challenges:
  - *Extract the relevant parameters automatically.*
  - *Translate the science to enormous samples.*
  - *Believe the results when no one will have time to reproduce.*
  - *Have confidence in results derived using machine learning or AI.*

Summary:

Maneage and its principles are described in arXiv:2006.03018. It is a customizable template that will do the following steps/instructions (all in simple plain text files).

- Automatically downloads the necessary *software* and *data*.
- Builds the software in a closed environment.
- Runs the software on data to generate the final research results.
- Modification of part of the analysis will only result in re-doing that part, not the whole project.
- Using LaTeX macros, paper's figures, tables and numbers will be Automatically updated after a change in analysis. Allowing the scientist to focus on the scientific interpretation.
- The whole project is under version control (Git) to allow easy reversion to a previous state. This encourages tests/experimentation in the analysis.
- The Git commit hash of the project source, is printed in the published paper and saved on output data products. Ensuring the integrity/reproducibility of the result.
- These slides are available at `https://maneage.org/pdf/slides-intro-short.pdf`.
- Longer slides are available at `https://maneage.org/pdf/slides-intro.pdf`.

For a technical description of Maneage's implementation, as well as a checklist to customize it, and tips on good practices, please see this page:
`https://gitlab.com/maneage/project/-/blob/maneage/README-hacking.md`