#### Maneage: a proof-of-concept for rigorous reproducible research paper criteria

Boud Roukema Institute of Astronomy, Nicolaus Copernicus University

11 March 2024 @IA NCU

Slides' main authors: Akhlaghi + Roukema; pdf built from git commit ba08d20; this pdf: https://cosmo.torun.pl/~boud/Roukema20240311IANCU.pdf

#### Reproducibility crisis in scientific research: astronomy

Snakes on a Spaceship – An Overview of Python in Heliophysics

"...inadequate analysis descriptions and loss of scientific data have made scientific studies difficult or impossible to replicate". From Burrell+2018, (arXiv:1901.00143).

Perspectives on Reproducibility and Sustainability of Open-Source Scientific Software

"It is our interest that NASA adopt an open-code policy because without it, reproducibility in computational science is needlessly hampered". From Oishi+2018, (arXiv:1801.08200).

Schrödinger's code: source code availability and link persistence in astrophysics

"We were unable to find source code online ... for 40.4% of the codes used in the research we looked at". From Allen+2018, (arXiv:1801.02094).

#### and in biology, other sciences, economics

Repeatability of published microarray gene expression analyses

loannidis+2009 tried to repeat the data analyses in 18 articles in Nature Genetics and only were able to (almost) reproduce only 2 DOI:10.1038/ng.295.

Is Economics Research Reproducible? 60 papers from Thirteen Journals Say "Usually Not"

Chang & Li (2015) were are able to reproduce just under half of 67 papers in prestigious journals, *with help* from the authors. DOI:10.17016/FEDS.2015.083

An empirical analysis of journal policy effectiveness for computational reproducibility

Stodden+2018 studied a random sample of 204 scientific papers in *Science* and were able to obtain data or code from 44% and reproduce the findings for 26%. DOI:10.1073/pnas.1708290115

"Reproducibility crisis" in the sciences?

Baker 2016, Nature 533, 452: 70% of researchers couldn't reproduce another scientist's results; half couldn't reproduce their own. nature.com/articles/533452a

## **Definitions & Clarification**

#### Replicability (hardware/statistical)

- Similar data, similar method.
- Involves data collection.
- Inherently includes measurement errors (can never be exactly reproduced).
- Example: Raw telescope image/spectra.
- ▶ NOT DISCUSSED HERE.



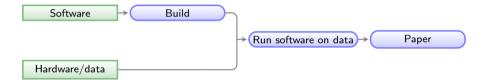
(C) CC BY-SA 2006, R. Feiler

#### Reproducibility (software/deterministic)

- **Same** data, **same** method.
- May include simulations of data.
- Involves data analysis,
- Example: 2 + 2 = 4 (i.e., sum of datasets).
- DISCUSSED HERE.

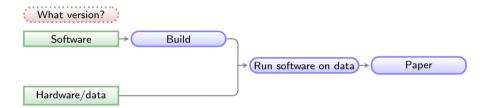


(C) 2008 J. Zawinski



Green boxes with sharp corners: source/input components/files.

Blue boxes with rounded corners: built components.



Green boxes with sharp corners: source/input components/files.

Blue boxes with rounded corners: built components.

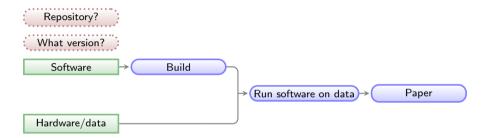
## Different package managers have different versions of software (repology.org, 2021/12/02)

GNU Astrono	omy l	Jtilities (Gnuastro	<b>b</b> )
Packaging statu	s		
Debian 9			
Debian 10			
Debian 11			
Debian 12			
Debian Unstable			
Deepin			
Devuan 2.0			
Devuan 3.0			
Devuan 4.0			
Devuan Unstable			
DPorts			
FreeBSD Ports			
Gentoo			
GNU Guix		PureOS Amber	
Kali Linux Rolling		PureOS landing	
LiGurOS stable		Raspbian Oldstable	0.8
LiGurOS develop		Raspbian Stable	0.14
OpenBSD Ports		Raspbian Testing	0.16.1
openSUSE Leap 15.1		RPM Sphere	0.16.1
openSUSE Leap 15.2		Trisquel 9.0	0.5
openSUSE Leap 15.3		Trisquel 10.0	0.11
openSUSE Tumbleweed		Ubuntu 18.04	0.5
enSUSE Science Tumbleweed		Ubuntu 20.04	0.11
Pardus 17		Ubuntu 20.10	0.12
Pardus 19		Ubuntu 21.04	0.14
Pardus 21		Ubuntu 21.10	
Parrot		Ubuntu 22.04	0.14
PLD Linux	0.15	Ubuntu 22.04 Proposed	0.16.1

# Astropy ekaning status

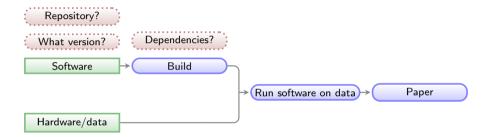
Packaging st	tatus
Debian 10	
Debian 11	
Debian 12	
Debian Unstable	
Debian Experimental	
Deepin	
Devuan 3.0	
Devuan 4.0	
Devuan Unstable	
Kali Linux Rolling	
Pardus 19	
Pardus 21	
Parrot	
PureOS Amber	
PureOS landing	
Raspbian Oldstable	
Raspbian Stable	
Raspbian Testing	
Trisquel 9.0	
Trisquel 10.0	
Ubuntu 18.04	
Ubuntu 20.04	
Ubuntu 20.10	4.0.1+post
Ubuntu 21.04	
Ubuntu 21.10	
Ubuntu 22.04	
Ubuntu 22.04 Proposed	4.3.1

ション 山田 マイビア イビア うくつ



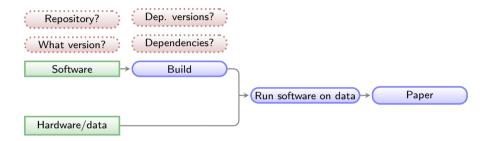
Green boxes with sharp corners: source/input components/files.

Blue boxes with rounded corners: built components.



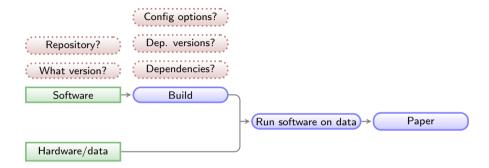
Green boxes with sharp corners: source/input components/files.

Blue boxes with rounded corners: built components.



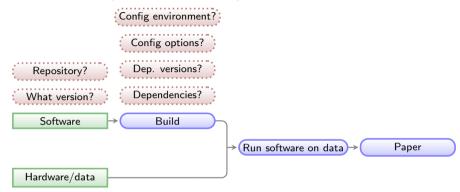
Green boxes with sharp corners: source/input components/files.

Blue boxes with rounded corners: built components.



Green boxes with sharp corners: source/input components/files.

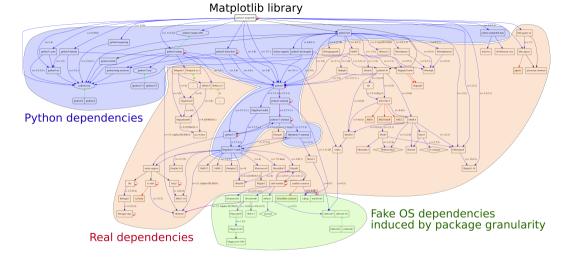
Blue boxes with rounded corners: built components.



▲□▶ ▲□▶ ▲□▶ ▲□▶ □ の00

Green boxes with sharp corners: source/input components/files.

Blue boxes with rounded corners: built components.



## 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.

э

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

Impact of "Dependency hell" on native building in various hardware (CPU architectures), retrieved from Debian on 2021/12/02



#### Tracker - Changelog - Bugs - packages.d.o - Source

Package(s):	astropy	Suite:	sid	~	Go
Compact mode	Co-maintainers				

Architecture	Version	Status	For	Buildd	State	Section	Logs	Actions
🗐 all	5.0.1	Installed	9d 9h 36m	x86-conova-01		misc	old   all (1)	giveback
amd64	5.0-1	Installed	9d 9h 37m	x86-csail-01		misc	old   all (1)	giveback
arm64	5.0-1	Installed	9d 9h 8m	arm-ubc-02		misc	old   all (1)	giveback
🖩 armel	5.0-1	Installed	9d 6h 52m	antheil		misc	old   all (1)	giveback
armhf	5.0-1	Installed	9d 8h 8m	hoiby		misc	old   all (1)	giveback
🖬 i386	5.0.1	Installed	9d 9h 57m	x86-grnet-01		misc	old   all (1)	giveback
🖬 mips64el 🕴	5.0-1	Build-Attempted	8d 18h 46m	mipsel-osuosl-04	out-of-date	misc	old   all (3)	giveback
i mipsel	5.0-1	Installed	9d 9h 37m	mipsel-manda-05		misc	old   all (1)	giveback
m ppc64el	5.0.1	Installed	9d 9h 37m	ppc64el-unicamp-01		misc	old   all (1)	giveback
🖬 s390x	5.0-1	Installed	9d 9h 57m	zandonai		misc	old   all (1)	giveback
🖬 alpha 👍	5.0-1	<b>BD-Uninstallable</b>	9d 10h 22m		out-of-date	misc	old   no log	giveback
Ell hppa 4	5.0-1	Build-Attempted	2d 17h 20m	c8000	out-of-date	misc	old   all (3)	giveback
hurd-i386 4	5.0-1	<b>BD-Uninstallable</b>	9d 10h 22m		uncompiled	misc	old   no log	giveback
🖬 ia64 ↓	5.0.1	<b>BD-Uninstallable</b>	9d 10h 22m		uncompiled	misc	old   no log	giveback
🖼 kfreebsd-amd64 👍	5.0-1	<b>BD-Uninstallable</b>	9d 10h 22m		uncompiled	misc	old   no log	giveback
🖬 kfreebsd-i386 🕴	5.0-1	<b>BD-Uninstallable</b>	9d 10h 22m		uncompiled	misc	old   no log	giveback
🖬 m68k 🕴	5.0-1	<b>BD-Uninstallable</b>	9d 10h 22m		out-of-date	misc	old   no log	giveback
M powerpc 4	5.0-1	<b>BD-Uninstallable</b>	9d 10h 22m		uncompiled	misc	old   no log	giveback
m ppc64	5.0.1	Installed	9d 9h 31m	kapitsa		misc	old   all (1)	giveback
iscv64	5.0-1	Installed	9d 6h 11m	rv-osuosl-02		misc	old   all (1)	giveback
lill sh4 ↓	5.0-1	BD-Uninstallable	9d 10h 21m		out-of-date	misc	old   no log	giveback
🖬 sparc64 🗍	5.0-1	<b>BD-Uninstallable</b>	9d 10h 21m		out-of-date	misc	old   no log	giveback
₩ x32 ↓	5.0-1	<b>BD-Uninstallable</b>	9d 10h 21m		out-of-date	misc	old   no log	giveback

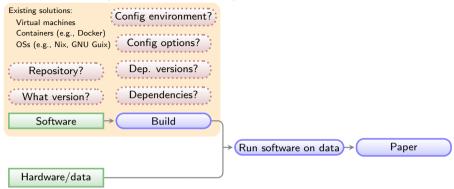
Astropy depends on Matplotlib



#### Debian Package Auto-Building

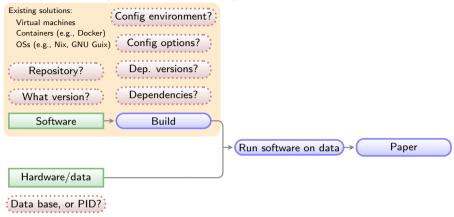
ackage(s): gnuastro		Su	iite: sid	∀ Go				
Compact mode Co-mai	intainers							
Architecture	Version	Status	For	Buildd	State	Section	Logs	Actions
all is not present in	the archite	cture list s	et by the maint	ainer				
amd64	0.16.1-1	Installed	14d 6h 8m	x86-csail-01		misc	old   all (1)	giveback
📾 arm64	0.16.1-1	Installed	14d 5h 56m	arm-ubc-03		misc	old   all (1)	giveback
📾 armel	0.16.1-1	Installed	14d 5h 26m	henze		misc	old   all (1)	giveback
📾 armhf	0.16.1 - 1	Installed	14d 5h 56m	arm-conova-02		misc	old   all (1)	giveback
🖼 i386	0.16.1-1	Installed	14d 5h 56m	x86-ubc-01		misc	old   all (1)	giveback
mips64el	0.16.1-1	Installed	14d 5h 26m	mipsel-aql-03		misc	old   all (1)	giveback
i mipsel	0.16.1-1	Installed	11d 15h 26m	mipsel-osuosl-04		misc	old   all (1)	giveback
ppc64el	0.16.1-1	Installed	14d 6h 8m	ppc64el-unicamp-01		misc	old   all (1)	giveback
s390x	0.16.1-1	Installed	14d 6h 7m	zani		misc	old   all (1)	giveback
🖬 alpha	0.16.1-1	Installed	7d 6h 11m	imago		misc	old   all (2)	giveback
el hppa	0.16.1-1	Installed	14d 5h 31m	c8000b		misc	old   all (1)	giveback
hurd-i386	0.16.1-1	Installed	12d 19h 21m	ironforge		misc	old   all (1)	giveback
ia64	0.16.1-1	Installed	14d 5h 41m	lifshitz2		misc	old   all (1)	giveback
kfreebsd-amd64	0.16.1-1	Installed	13d 22h 31m	kamp		misc	old   all (1)	giveback
kfreebsd-i386	0.16.1-1	Installed	11d 11h 31m	kamp		misc	old   all (1)	giveback
m68k	0.16.1-1	Installed	14d 4h 21m	vs92		misc	old   all (1)	giveback
powerpc	0.16.1-1	Installed	14d 5h 31m	blaauw		misc	old   all (1)	giveback
ppc64	0.16.1-1	Installed	14d 6h	blaauw2		misc	old   all (1)	giveback
riscv64	0.16.1-1	Installed	14d 5h 30m	rv-osuosl-01		misc	old   all (1)	giveback
🔤 sh4	0.16.1-1	Installed	14d 5h	sh4-do-02		misc	old   all (1)	giveback
sparc64	0.16.1-1	Installed	14d 5h 10m	nvg5120b		misc	old   all (1)	giveback
M x32	0.16.1-1	Installed	14d 6h	x32-do-02		misc	old   all (1)	giveback

GNU Astronomy Utilities doesn't.



Green boxes with sharp corners: source/input components/files.

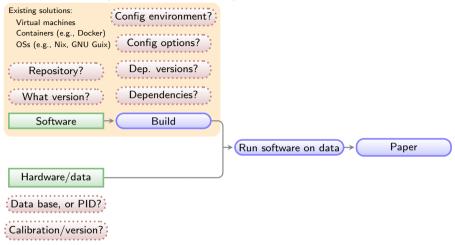
Blue boxes with rounded corners: built components.



▲□▶ ▲□▶ ▲□▶ ▲□▶ □ の00

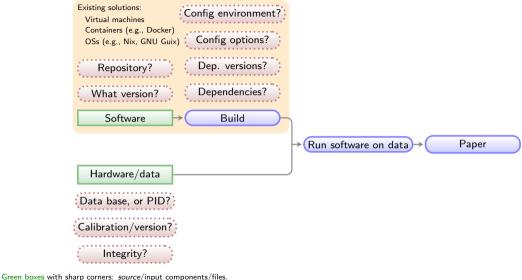
Green boxes with sharp corners: source/input components/files.

Blue boxes with rounded corners: built components.



Green boxes with sharp corners: source/input components/files.

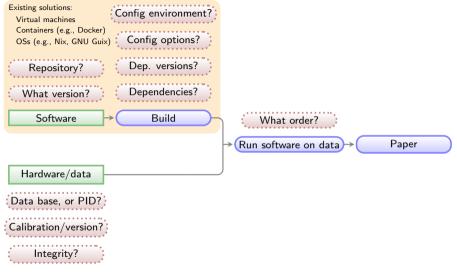
Blue boxes with rounded corners: built components.



イロト 不得 トイヨト イヨト

э.

Blue boxes with rounded corners: *built* components/mes.

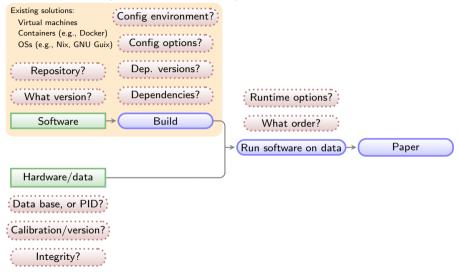


イロト 不得 トイヨト イヨト

э.

Green boxes with sharp corners: *source*/input components/files.

Blue boxes with rounded corners: built components.

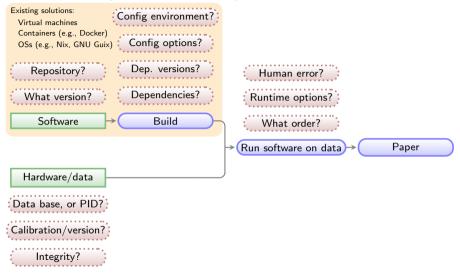


イロト 不得 トイヨト イヨト

э.

Green boxes with sharp corners: *source*/input components/files.

Blue boxes with rounded corners: built components.

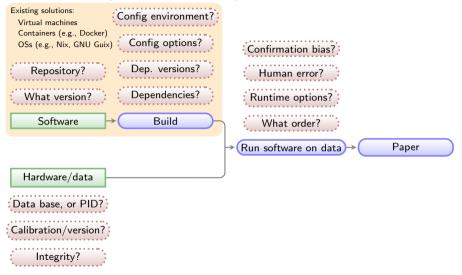


イロト 不得 トイヨト イヨト

э.

Green boxes with sharp corners: *source*/input components/files.

Blue boxes with rounded corners: built components.

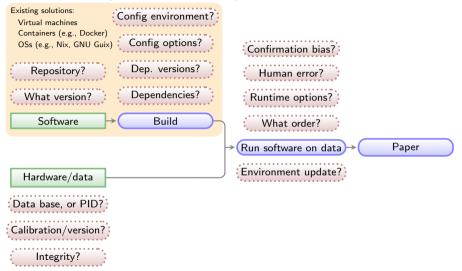


イロト 不得 トイヨト イヨト

э.

Green boxes with sharp corners: *source*/input components/files.

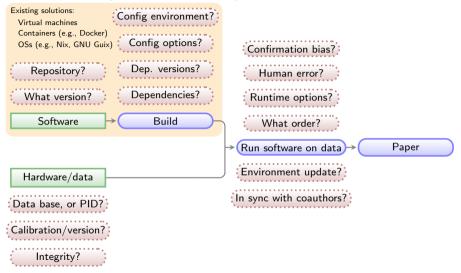
Blue boxes with rounded corners: built components.



イロト 不得 トイヨト イヨト

э.

Green boxes with sharp corners: *source*/input components/files. Blue boxes with rounded corners: *built* components.

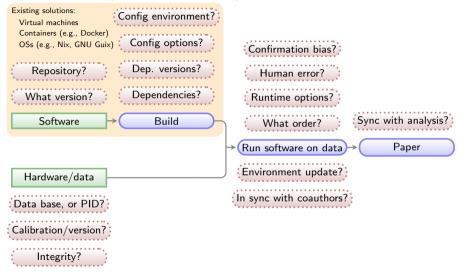


イロト 不得 トイヨト イヨト

э.

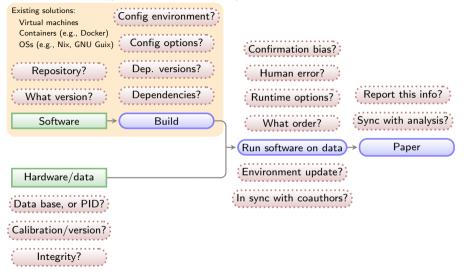
Green boxes with sharp corners: *source*/input components/files.

Blue boxes with rounded corners: built components.



Green boxes with sharp corners: *source*/input components/files.

Blue boxes with rounded corners: built components.

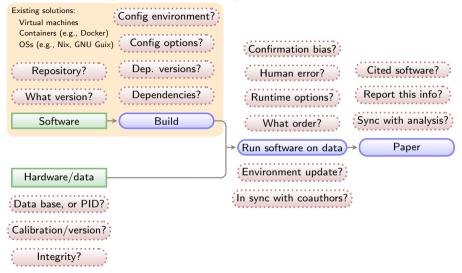


イロト 不得 トイヨト イヨト

э.

Green boxes with sharp corners: *source*/input components/files.

Blue boxes with rounded corners: built components.

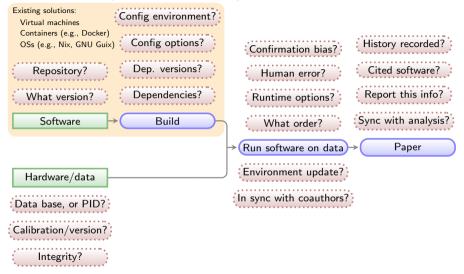


Green boxes with sharp corners: source/input components/files.

Blue boxes with rounded corners: built components.

Di Cosmo & Pellegrini (2019) Encouraging a wider usage of software derived from research

"Software is a hybrid object in the world [of] research as it is equally a driving force (as a tool), a result (as proof of the existence of a solution) and an object of study (as an artefact)".

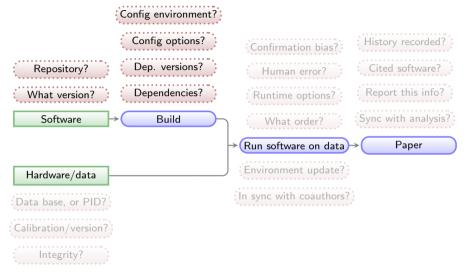


A D > A B > A B > A B >

э

Green boxes with sharp corners: source/input components/files.

Blue boxes with rounded corners: built components.



イロト 不得 トイヨト イヨト

э

Green boxes with sharp corners: *source*/input components/files.

Blue boxes with rounded corners: built components.

#### Criteria for a solution

#### 1. Completeness (self-contained-ness):

- ▶ (1) Only dependency should be POSIX tools (discards Conda or Jupyter which need Python).
- (2) Plain text: Project's source should be in plain-text (binary formats need special software)
- (3) No impact on host OS libraries, programs, env variables.
- (4) Must not require root permissions (discards tools like Docker or Nix/Guix).
- (5) Builds its own controlled software + env variables.
- (6) Should be usable without an internet connection.
- (7) Contains full instructions for inputs, software building, and outputs: analysis + narrative + graphical output (e.g. pdf or html)
- (8) Should be non-interactive or runnable in batch (user interaction is an incompleteness).
- 2. Modularity: Parts of the project should be re-usable in other projects.
- ▶ 3. Minimal complexity: Occam's razor: "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.
- ▶ 4. Scalability: an implementation should easily scale to arbitrarily large, complex projects.

#### Criteria for a solution

- **5. Verifiable inputs and outputs:** Inputs and Outputs must be automatically verified.
- 6. Recorded history: Exploratory research involves modifying methods, reducing over-ambitious goals, serendipity; hypothesis testing should have a fixed, predefined method did the authors modify the method? how? what? which? when?
- 7. Including the narrative that is associated with the analysis: a workflow alone lacks motivations, interpretations.
- 8. Free and open source software (FOSS): Free software is essential: non-free software is not configurable, not distributable, and is dependent on a non-free provider (who may discontinue it after e.g. 5–10 years).

Reminder: Free Software Definition:

freedom to: 2: distribute 3: distribute modified versions

## Our solution: CiSE 23 (3), pp 82-91: DOI:10.1109/MCSE.2021.3072860, arXiv:2006.03018

DITORS: Lorena A. Barba, laberbaggeu edu Sandra Gaalor, anorira malomierd ar

SPECIAL TRACK: REPRODUCIBLE RESEARCH

#### Toward Long-Term and Archivable Reproducibility

Analysis pipelines commonly use high-level technologies that are popular when created. but are unlikely to be readable, executable, or sustainable in the long term. A set of criteria is introduced to address this problem: completeness (no execution requirement beyond a minimal Unix-like operating system, no administrator privileges, no network connection, and storage primarily in plain text); modular design; minimal complexity; scalability; verifiable inputs and outputs: version control: linking analysis with narrative: and free and open-source software. As a proof of concept, we introduce "Maneage" (managing data lineage), enabling cheap archiving, provenance extraction, and peer verification that has been tested in several research publications. We show that longevity is a realistic requirement that does not sacrifice immediate or short-term reproducibility. The caveats (with proposed solutions) are then discussed and we conclude with the benefits for the various stakeholders. This article is itself a Maneage'd project (project commit 313db0b). Appendices-Two comprehensive appendices that review the longevity of existing solutions are available as supplementary "Web extras," which are available in the IEEE Computer Society Digital Library at http://doi.ieeecomputersociety.org/10.1109/ MCSE 2021.3072860. Reproducibility-All products gyallable in zerodo, 4913277, the Git history of this paper's source is at git maneage.org/paper-concept.git, which is also archived in Software Heritage autotyle-23fee87068c3612def011f16156769778750e0df10f Clicking on the SWHIDs in the digital format will provide more "context" for same content.

Reproducible research has been discussed in the sciences for at least 30 years.<sup>12</sup> Many "solutions") have been proposed, which mostly rely on the common technology of the day, starting

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://oreativecommore.org/ficenses/by/4.0/ Digital Object Identifier 10.1103/MCSE.2021.3072860 Date of publication 13 April 2021; date of ourrent vension 15 June 2021. with Make and Matlab libraries in the 1990s, Java in the 2000s, and mostly shifting to Python during the past decade.

However, these technologies develop fast, e.g., code written in Python 2 (which is no longer officially maintained) often cannot run with Python 3. The could of staying up to date within this rapidly exciving landscape is high. Scientific projects, in particular, suffer the most Scientists have to focus on their own research domain, but to some degree, they need to understand the technology of their tools because it determines their results.

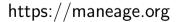
Comp

82

Published by the IEEE Computer Society N

In M	1anea	ge N	Jana	ging	dat× +							×
$\rightarrow$	С	6	0	۵	https://maneage.org	☆ Q Search	坌	${\times}$	$\overline{\mathbf{A}}$	hiv	»	=
	Hon	ne			About	Git			Tutor	ial		
			$\leq$	>								
			Ч		~ ~							- 1
			9	T	ΨΨ	Maneage						- 1
			٢	>		Managing Data Lin	ea	ae				- 1
			U		$\Psi \Psi$	Data Elli	- 4	90				

Maneage is a framework for having full control over a project's data lineage (thus producing a reproducible result), Maneage is a recipient of the RDA Europe Adoption grant and was featured in a Nature Astronomy "News and Views" article (Kuttel 2021, free-to-read link). To learn more about its founding criteria and a basic introduction, see Akhitaghi et al. (2021), published in CISE (Gold Open Access), also available in arXiv:2006.03018 (with extended appendix in one PDF). You can also watch the short talk linked below or see this published FDA Adoption story (a short PDF).



## 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 its dependencies.
- Installs similar environment on GNU/Linux, or Darwin/Xnu (macOS, iOS) systems.
- Works very much like a package manager (e.g., apt or brew).

File Edit	Options Buffers Tools Conf Help
P 🖪	🗮 🗶 🖳 Save 🥱 Undo 🕌 🖷 龍 🔍
161	openssh-version = 8.0p1
162	patch-version = 2.7.6
163	pcre-version = 8.44
164	pixman-version = 0.40.0
165	python-version = 3.10.0
166	r-cran-version = 4.1.2
167	ramses-scalav-version = 0.0-482f90f
168	ramses-scalav-commit-id = 482f90f59542
169	ramses-use-mpif08-version = 3.0-a317f07
170	ramses-use-mpif08-commit-id = a317f0761dcc
171	revolver-version = 0.0-43cd72f
172	revolver-commit-id = 3b153351e3bb67c50010c0dc2 *
	ersions.conf 43% L167 Git-gevcurvtest [(Conf[Unix]
348	#
349	# CLASS:R-CRAN-START (important identifier for 'a
	•wk'; don't modify this line)
350	r-cran-cli-version = 2.5.0
351	r-cran-colorspace-version = 2.0-1
352	r-cran-cowplot-version = 1.1.1
353	r-cran-crayon-version = 1.4.1
354	r-cran-digest-version = 0.6.27
355	r-cran-ellipsis-version = 0.3.2
356	r-cran-fansi-version = 0.5.0
357	r-cran-farver-version = 2.1.0
337	

(日)

#### Controlled environment and build instructions

emacs@akhlaghi File Edit Options Buffers Tools Makefile Help 0 Save 🥎 Undo 🐰 include reproduce/software/config/installation/texlive.mk include reproduce/software/config/installation/versions.mk lockdir = \$(BDIR)/locks tdir = \$(BDIR)/software/tarballs ddir = \$(BDIR)/software/build-tmp idir = \$(BDTR)/software/installed ibdir = \$(BDTR)/software/installed/bin 41 - 5 ildir = \$(BDIR)/software/installed/lib dtexdir = \$(shell pwd)/reproduce/software/bibtex itidir = \$(BDIR)/software/installed/version-info/tex ictdir = \$(RDIR)/software/installed/version-info/cite ipydir = \$(BDIR)/software/installed/version-info/python ibidir = \$(BDIR)/software/installed/version-info/proglib # Set the top-level software to build. all: \$(foreach p, \$(top-level-programs), \$(ibidir)/\$(p)) \ \$(foreach p. \$(top-level-python), \$(ipydir)/\$(p)) \ \$(itidir)/texlive # Other basic environment settings: We are only including the host # operating system's PATH environment variable (after our own!) for the Ø compiler and linker. For the library binaries and headers, we are only # using our internally built libraries. # To investigate: 1) Set SHELL to `\$(ibdir)/env - NAME=VALUE \$(ibdir)/bash' and set all the parameters defined bellow as 'NAME=VALUE' statements before calling Bash. This will enable us to completely ignore the user's native environment 2) Add '--noprofile --norc' to '.SHELLFLAGS' so doesn't load the uper's environment and if ONECHELLY CHELLEL ACC. := --noprofile --norc -ec export CCACHE DISABLE := 1 export PATH := \$(ibdir) export SHELL := \$(ibdir)/bash export CRRELAGS := .I\$/idir)/include export PKG CONFIG PATH := \$(ildir)/pkgconfig export PKG CONFIG LIBDIR := \$(ildir)/pkgconfig export LD RUN PATH := \$(ildir):\$(il64dir) export LD LIBRARY PATH := \$(ildir):\$(il64dir) export IDELAGS i= \$(reath\_compand) =(\$(i)dir) # We want the download to happen on a single thread. So we need to define a lock, and call a special script we have written for this job. These are Uters high-level mk 4% 181 Gittmaster (Makefile)

emacs@akhlaghi File Edit Options Buffers Tools Makefile Help 🛄 Save 👆 Undo 📈 p Q # not 'LIBS' # On Nac systems, the build complains about 'clang' specific # features, so we can't use our own GCC build here. if [ x\$(on mac os) = xves 1: then \ export CC=clang: \ export (XX=clang++: ) cd \$(ddir) \ && rn -rf cnake-\$(cnake-version) \ && tar xf \$< \ && cd\_cmake-\$(cmake-version) \ && ./bootstrap --prefix=\$(idir) --system-curl --system-zlib \ --system-bzip2 --system-liblzma --no-qt-gui \ && make -1\$(numthreads) LIBS="\$\$LIBS -1ssl -lcrvpto -1z" VERBOSE=1 \ && make install \ && cd ... \ && rm -rf cmake-\$(cmake-version) \ A& echo "(Make Sicmake-version)" > 50 \$(ibidir)/ghostscript: \$(tdir)/ghostscript-\$(ghostscript-version).tar.gz \$(call obuild, \$<, obostscript-\$(obostscript-version)) \</pre> && echo "GPL Ghostscript \$(ghostscript-version)" > \$0 G(ibidir)/gnuastro: \$(tdir)/gnuastro-\$(gnuastro-version).tar.lz \ \$(ibidir)/ghostscript \ \$(ibidir)(libinen) \$(ibidir)/libtiff ) \$(ibidir)/libnit2 ) \$(ibidir)/wcslib \ \$(ibidir)/asl ifen (\$(static build) ves) staticonts="--enable.staticsves --enable.shared=no": \$(call gbuild, \$<, gnuastro-\$(gnuastro-version), static, \</pre> \$\$staticopts. -i\$(punthreads). \ make check -i\$(numthreads)) \ A& on \$(dtexdir)/onuastro\_tex\_\$(ictdir)/ ) && echo "GNU Astronomy Utilities \$(gnuastro-version) \citep(gnuastro}" > \* \$(ibidir)/imagemagick: \$(tdir)/imagemagick-\$(imagemagick-version).tar.xz \ \$(ibidir)/libipeg \ \$(ibidir)/libtiff \ \$(ibidir)/zlib \$(call gbuild, \$<, ImageMagick-\$(imagemagick-version), static, \</pre> --without-x --disable-openmp, V=1) \ && echo "ImageMagick \$(imagemagick-version)" > \$0

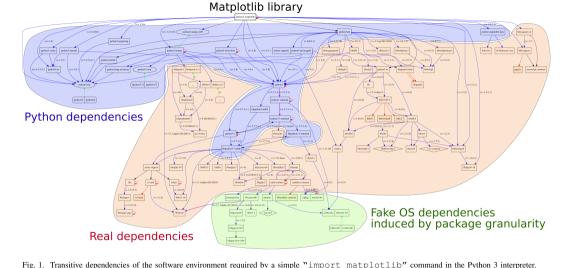
U:--- high-level.mk 67% L584 Git:master (Makefile)

#### Controlled environment and build instructions

emacs@akhlaghi File Edit Options Buffers Tools Makefile Help 0 Save 🥎 Undo 🐰 include reproduce/software/config/installation/texlive.mk # not 'LIBS' include reproduce/software/config/installation/versions.mk lockdir = \$(BDIR)/locks tdir = \$(BDIR)/software/tarballs ddir = \$(BDIR)/software/build-tmp export CC=clang: \ idir = \$(BDTR)/software/installed export (XX=clang++: ) ibdir = \$(BDTR)/software/installed/bin 41 - 5 ildir = \$(BDIR)/software/installed/lib cd \$(ddir) \ dtexdir = \$(shell pwd)/reproduce/software/bibtex itidir = \$(BDIR)/software/installed/version-info/tex && tar xf \$< \ ictdir = \$(RDIR)/software/installed/version-info/cite ipydir = \$(BDIR)/software/installed/version-info/python ibidir = \$(BDIR)/software/installed/version-info/proglib # Set the top-level software to build. && make install \ all: \$(foreach p, \$(top-level-programs), \$(ibidir)/\$(p)) \ && cd ... \ \$(foreach p. \$(top-level-python), \$(ipydir)/\$(p)) \ \$(itidir)/texlive # Other basic environment settings: We are only including the host # operating system's PATH environment variable (after our own!) for the Ø compiler and linker. For the library binaries and headers, we are only # using our internally built libraries. # To investigate: 1) Set SHELL to `\$(ibdir)/env - NAME=VALUE \$(ibdir)/bash' and set all the parameters defined bellow as 'NAME=VALUE' statements before calling Bash. This will enable us to completely ignore the user's native environment ifen (\$(static build) ves) 2) Add '--noprofile --norc' to '.SHELLFLAGS' so doesn't load the uper's environment and if ONECHELLY CHELLEL ACC. := --noprofile --norc -ec export CCACHE DISABLE := 1 export PATH := \$(ibdir) export SHELL := \$(ibdir)/bash export CPPELAGS -- .I\$(idir)(include export PKG CONFIG PATH := \$(ildir)/pkgconfig export PKG CONFIG LIBDIR := \$(ildir)/pkgconfig export LD RUN PATH := \$(ildir):\$(il64dir) export ID LIBRARY PATH := \$(ildir):\$(il64dir) export IDELAGS := \$(rmath command) -1\$(ildir) # We want the download to happen on a single thread. So we need to define a lock, and call a special script we have written for this job. These are Uters high-level mk 4% 181 Gittmaster (Makefile)

emacs@akhlaghi File Edit Options Buffers Tools Makefile Help 🛄 Save 👆 Undo 📈 p Q # On Nac systems, the build complains about 'clang' specific # features, so we can't use our own GCC build here. if [ x\$(on mac os) = xves 1: then \ && rn -rf cnake-\$(cnake-version) \ && cd\_cmake-\$(cmake-version) \ && ./bootstrap --prefix=\$(idir) --system-curl --system-zlib \ --system-bzip2 --system-liblzma --no-qt-gui \ && make -1\$(numthreads) LIBS="\$\$LIBS -1ssl -lcrvpto -1z" VERBOSE=1 \ && rm -rf cmake-\$(cmake-version) \ A& echo "(Make Sicmake-version)" > 50 \$(ibidir)/ghostscript: \$(tdir)/ghostscript-\$(ghostscript-version).tar.gz \$(call obuild, \$<, obostscript-\$(obostscript-version)) \</pre> && echo "GPL Ghostscript \$(ghostscript-version)" > \$0 [(ibidir)/gnuastro: \$(tdir)/gnuastro-\$(gnuastro-version).tar.lz \ \$(ibidir)/ghostscript \ \$(ibidir)(libinen) \$(ibidir)/libtiff ) \$(ibidir)/libnit2 ) \$(ibidir)/wcslib \ \$(ibidir)/asl staticonts="--enable.staticsves --enable.shared=no": \$(call gbuild, \$<, gnuastro-\$(gnuastro-version), static, \</pre> \$\$staticopts. -i\$(punthreads). \ make check -i\$(numthreads)) \ A& on \$(dtexdir)/onuastro\_tex\_\$(ictdir)/ ) && echo "GNU Astronomy Utilities \$(gnuastro-version) \citep(gnuastro}" > \* \$(ibidir)/imagemagick: \$(tdir)/imagemagick-\$(imagemagick-version).tar.xz \ \$(ibidir)/libipeg \ \$(ibidir)/libtiff \ \$(ibidir)/zlib \$(call gbuild, \$<, ImageMagick-\$(imagemagick-version), static, \</pre> --without-x --disable-openmp, V=1) \ && echo "ImageMagick \$(imagemagick-version)" > \$0

Utres high-level mk 67% 1584 Gittmaster (Makefile)



# Example: Matplotlib (a Python visualization library) build dependencies

From "Attributing and Referencing (Research) Software: Best Practices and Outlook from Inria" (Alliez et al. 2019, hal-02135891)

▲□▶▲圖▶▲≣▶▲≣▶ ■ のへ⊙

All high-level dependencies are under control (e.g., NoiseChisel's dependencies)

#### **GNU/Linux** distribution

#### \$ ldd .local/bin/astnoisechisel

libgnuastro.so.7 => /PROJECT/libgnuastro.so.7 (0x00007f6745f39000) libgit2.so.26 => /PROJECT/libgit2.so.26 (0x00007f6745df1000) libtiff.so.5 => /PROJECT/libtiff.so.5 (0x00007f6745d77000) liblzma so 5 => /PROJECT/liblzma so 5 (0x00007f6745d4f000) libipeg.so.9 => /PROJECT/libipeg.so.9 (0x00007f6745d12000) libwcs.so.6 => /PROJECT/libwcs.so.6 (0x00007f6745ba8000) libcfitsio.so.8 => /PR0JECT/libcfitsio.so.8 (0x00007f674588b000) libcurl.so.4 => /PR0JECT/libcurl.so.4 (0x00007f6745811000)  $libssl.so.1.1 \Rightarrow /PR0JECT/libssl.so.1.1 (0x00007f6745777000)$ libcrypto.so.1.1 => /PR0JECT/libcrypto.so.1.1 (0x00007f6745491000) libz.so.1 => /PR0JECT/libz.so.1 (0x00007f6745474000) libgsl.so.23 => /PROJECT/libgsl.so.23 (0x00007f67451e3000) libgslcblas.so.0 => /PROJECT/libgslcblas.so.0 (0x00007f67451a1000) linux-vdso.so.1 (0x00007fffdcbf7000) libpthread.so.0 => /usr/lib/libpthread.so.0 (0x00007f6745006000) libm.so.6 => /usr/lib/libm.so.6 (0x00007f6745027000) libc.so.6 => /usr/lib/libc.so.6 (0x00007f6744e43000) libdl.so.2 => /usr/lib/libdl.so.2 (0x00007f6744e1e000)  $/lib64/ld - linux - x86 - 64 \cdot so 2 => /usr/lib64/ld - linux - x86 - 64 \cdot so 2$ 

#### Darwin/Xnu

\$ otool -L .local/bin/astnoisechisel

/PROJECT/libgnuastro.7.dylib (comp ver 8.0.0, cur ver 8.0.0) /PROJECT/libgit2.26.dylib (comp ver 26.0.0, cur ver 0.26.0) /PROJECT/libtiff.5.dylib (comp ver 10.0.0, cur ver 10.0.0) /PROJECT/liblpg.9.dylib (comp ver 8.0.0, cur ver 8.4.0) /PROJECT/libbrgs.9.dylib (comp ver 6.0.0, cur ver 12.0.0) /PROJECT/libcifisio.8.dylib (comp ver 8.0.0, cur ver 6.2.0) /PROJECT/libcifisio.8.dylib (comp ver 10.0.0, cur ver 10.0.0) /PROJECT/libcifisio.8.dylib (comp ver 11.0, cur ver 11.0) /PROJECT/libcifisio.8.dylib (comp ver 11.0, cur ver 11.0) /PROJECT/libcifisio.8.dylib (comp ver 1.1.0, cur ver 11.0) /PROJECT/libcifisio.8.dylib (comp ver 1.1.0, cur ver 11.0) /PROJECT/libcifisio.8.dylib (comp ver 1.0.0, cur ver 12.11) /PROJECT/libcifisio.8.dylib (comp ver 1.0.0, cur ver 12.0.0) /PROJECT/libgil.23.dylib (comp ver 10.0, cur ver 12.0.0) /usr/lib/jibSystem.B.dylib (comp ver 1.0.0, cur ver 12.0.0)

Project libraries: High-level libraries built from source for each project (note the same version in both OSs). GNU C Library: Project specific build is in progress (http://savannah.nongnu.org/task/?15390). Closed operating system files: We have no control on low-level non-free operating systems components.

#### Advantages of this build system

- Project runs in fixed/controlled environment: custom build of bash, make, GNU Coreutils (1s, 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 (~ 2–3 hours; Sep 2022).
- Dependencies of different projects will not conflict.
- Everything in plain text (human & computer readable/archivable).

### Software citation automatically generated in paper

DRAFT PAPER, DER 1 (pp), Year Month day



Figure 21 (a) An example image of the Wide-Field Planetary Carners 2, or beaut the Habble Space Telescope from 1903 to 2009. This is one of the sample images freenthe FITS standard webpage, kept as examples for this (the format, (b) Habsgams of pilet solves in (a).

removes the necessity to add further dependencies (to create the physics) to your projects. Threas are high-read language libraries like Maphedito which also possenze plant. However, the problem like Maphedito which also possenze plant. The server, the problem like Mapheditory and the server plant and the server like Mapheditory and the server plant and the server like Mapheditory and the server like server plant with harm the precolschilting of your paper. Note that after serveral work, the binary the obstance is not enough the server plant and therefore to tailing the libraries from source is the only option to therefore to tailing the libraries from source is the only option to

Furthermore, since PGPPlots is built by MI<sub>4</sub>X is 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'.

This template also defines two BIJJS masses that allow you mouth test within you demonstrat as new an alway. For example, this test has been matched as news. If you comment the line of the sides of wight a site of the line of wight defining the line in the diffusion highlight the hanges, then the cuse that was marked as new will become block traditions by allow the line in the Pi-Yies of the site of the line of wight a site of the line of the line of the line of wight a site of the line of the line of wight a site of the line of the li

2. NOTICE AND CITATIONS

To encourage other scientists to publish similarly reproducible papers, plasse add a notice close to the start of your paper or in the end of the abstract clearly meationing 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: Adultadia and Ichikowa (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).

InterPenirrors.ctaicorp/graphics/pg/Feomritr/pg/plotsidee/pg/plots.pdf

#### YOUR NAME OF AL. 3. ACKNOWLEDGEMENTS

Piceose include the following two paragraphs in the Acknowledgement societion of your paper. This reproducible paper template was developed in parallel with Guasarto, so it benefited from the same grants. If you don't use Grassitro in your final/customized project, focuse remove it from the paragraph below, only mentioning the provolucible name template.

This meanth was partly done using GNL Autonomy Unities Ginauron, and control 2010. By an in the prophositive paper kampiate of 36-4e 2364dc3-64m; Work on Ginauton and the approximation paper on the strength on the particular by part erg DISTS solutioning main in Gener. In Add for Streneris Reserved C124-001, 232(2030), the Burgenman inder Maine Ottorion 2020 research and instruction papers Research Council (JRC) advanced grant 336695 MUSICOS, Brazynez Uhion's Distribution 2020 research and instruction papers Research Council (JRC) advanced grant 336695 MUSICOS, Brazynez Uhion's Distribution 2020 research and instruction papers Research Council (JRC) advanced grant 336695 MUSICOS, Brazynez Uhion's Distribution and the structure of the paper of the structure of the distribution of the structure of the structure of the structure distribution of the structure of the structure of the structure distribution of the structure of the structure of the structure distribution of the structure of the structure of the structure of the distribution of the structure of the structure of the structure of the distribution of the structure of the structure of the structure of the distribution of the structure of the structure of the structure of the distructure of the structure of the structur

This research was done with the following free software progroups and libraries: Brin? 10.6 CETTSIO 3.45 CMake 3.14.2 cHPL 7.63.0. Discotes flock 0.2.3. Eds 5.36. Erer/Type 2.9. Git 2.21.0 (2011) Astronomy Utilities 0.9 (Akhlashi and Ehikora) 2015), GNU AWK 5.0.0, GNU Bash 5.0.7, GNU Binutik 2.32. GNU Compiler Collection (GCC) 9.1.9. GNU Consults 8.31. GNU Dafatile 3.7. GNU Findutile 4.6.0 199-e36: GNU Green 1.1. GNU Gain 1.10. GNU Integer Set Library 0.18. GNU Libroid 2.4.6 ONU MA LA LE ONU Mile 4.2.90 ONU Multiple Precision Arithmetic Library 6.1.2. GNU Multiple Precision Consolex library, GNU Multiple Precision Floating Processon Jubby 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 West 120.1 GNU Which 2.21 GPL Glossforder 9.26 HDES about 1.10.5. ImageMagick 7.0.8-46. Libbed 0.9.1. Libeit 2.0.26.0. Libince vib. Library 1.6.37. Libriff 4.0.10. Lain 1.20. Metastore (forked) 1.1.2.23.649170b. OpenBLAS 0.3.5. Open MPI 4.0.1. OpenSSL1111a, PatchELE0.9, nke-config 0.29.2, Pethon 3.7.3, Davia 6.0 WCSUB 6.2 X7 Units 5.2.4 Zio 3.0 and Ziih 1.2.11 (Astrony Collaboration et al. 2013: Astrony Collaboration et al. 2018). Cycler 0 10.0. Cythen 0 29.6 (Rebuel et al. 2011). h5m/ 2.9.0. Kiwisolver 1.0.1. Matriotlib 3.0.2 (Humer 2007). Numry 1.16.2 (use der Welt et al. 2011), eksemplie 1.5.1. Politeiner 2.3.1, python-dateutil 2.8.0, Scipy 1.2.1 (Oliphant 2007: Millman and Alexade 2011). Schusterle 40.8.0 Schusterle com 2.2.0 and Six 1.12.0. The lifteX source of the paper was compiled to make the PDF using the following packages: hiber 2.12, hiber 2.12. hibiter 3.12 hibbiter 3.12 cartion 2018-10-05 cartion 2018-10.05 courier 2016.05.24 courier 2016.05.24 counter 5.24 datetime 2.60, datetime 2.60, ec 1.0, ec 1.0, etoolbox 2.5f, etool-3.05. funtaxes 1.0d. fontaxes 1.0d. footmise 5.5h. footmise 5.5h. fn 2 lid fn 2 lid Journey I () Journey I () measts I 554 measts I 554 pef 3.1.2, pef 3.1.2, pefplots 1.16, pefplots 1.16, preprint 2011, prominit 2011 setsnace 6.7a setsnace 6.7a tex 3.14159765 tex 3.14159265, texavre 2.501, texavre 2.501, times 2016-06-24. times 2016-06-24, titlesec 2 10.2, titlesec 2 10.2, tyfonts 2016DRAFT PAPER, much (pp), Year Month day

06-24, txfoats 2016-06-24, ulem 2016-06-24, ulem 2016-06-24, xcolor 2.12, xcolor 2.12, xkayau 2.7a and skayval 2.7a. Wo are very grateful to all their creates for freely providing this necesisary infrastructure. This research (and many others) would not be nossible without them.

#### References

Akhighi, M. and T. Ichikawa (Sept. 2015). Apr7, 220, 1. Anray Collaboration et al. (Oct. 2015). Add., 55 (A33). Anray Collaboration et al. (Sept. 2016). Add., 56 (A33). Biana, B. et al. (New 2017). Add., 605, AJ, 156 (123). Biana, S. et al. (New 2017). Add., 605, AJ, 156 (123). Biana, S. et al. (New 2017). Add., 605, AJ, 166 (123). Biana, S. J. and N. 2011). COSE, 11, 33. Biana, S. J. and M. Avratis, (Mar. 2011). COSE, 13, 9. Okpose, T. B. (Mey 2005). COSE, 9, 40. Milliana, K. J. and (Mar. 2011). COSE, 113, 22. YOUR NAME IT AL.

### Software citation automatically generated in paper

DRAFT PAPER, DERI (pp), Year Month day



Figure 21 (a) An example image of the Wide-Field Planetary Carners 2, or beaut the Habble Space Telescope from 1903 to 2009. This is one of the sample images freenthe FITS standard webpage, kept as examples for this (the format, (b) Habsgams of pilet solves in (a).

removes the necessity to add further dependencies (to create the physics) to your projects. Threas are high-read language libraries like Maphedito which also possenze plant. However, the problem like Maphedito which also possenze plant. The server, the problem like Mapheditory and the server plant and the server like Mapheditory and the server plant and the server like Mapheditory and the server like server plant with harm the precolschilting of your paper. Note that after serveral work, the binary the obstance is not enough the server plant and therefore to tailing the libraries from source is the only option to therefore to tailing the libraries from source is the only option to

Furthermore, since PGPPlots is built by MIJAX it respects all the properties of your text (for example line within and fonts and etc). Therefore the final plot blends in your paper much more mixely. It sho has a wonderful marnal<sup>1</sup>.

This template also defines two BIJK masses that also you to much test within you document as new anisative. For example, this test has been matched as new. If you comment the list of you document that has a set of the list of why defining the list of distars highlight changes, then the sure that was marked as well become black totally been in unit the ener of the test and the one marked Lensory will not be in the list of PMP. You comments and the list of the list of the list of the PMP. You comments and the list of the list of the list of the PMP. You comments and the list of the new parts or notice) and new co-ambient (who dot's start to be discussed) by these issues in the first first in reading).

2. NOTICE AND CITATIONS

To encourage other scientists to publish similarly reproducible papers, plasse add a notice close to the start of your paper or in the end of the abstract clearly meationing 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: Adultadia and Ichikowa (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).

Intp://www.com/org/puplics/pgfcom///pgfplots/ko/pgfplots.pdf

#### YOUR NAME OF AL. 3. ACKNOWLEDGEMENTS

Please include the following two paragraphs in the Acknowledgement section of your paper. This reproducible paper template was developed in parallel with Gausatos, so it benefited from the series areau. (Found dorf) the Gausator is your final/contention

project, forease remove 16 mm the prograph bolies, with meastiming the reproducible paper tamplate. This research was parily done using GNU Astronomy Unities (Ginasaro, and calefold 1009, and the prophositile paper tamplate (JOS-4g-28/dall-6-dag). Work on Ganation and the Ministry of Hinzaro, Callure, Sprox, Science, and Tchroniogy (MSTV) solubindip unit in Garac-in-Aid for Simmirk Research (214-001, 242/2003), the European Research Coursel (ERC) advanced grant 33/668/MISRCOR, Farepare Likely Michalorshele Callure matterments (NG-1164) to the SUBSMIT

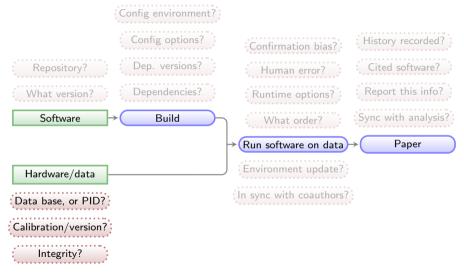
ITN, and from the Spanish Ministry of Economy and Competi-Electron (MINECO) under smat number AVA2016-26219.P This research was done with the following free software prorroms and libraries: Brin2 1.0.6 CETTSIO 3.45 CMake 3.14.2 cURL 7.63.0, Discoteg flock 0.2.3, File 5.36, FreeTyne 2.9, Git 2015), GNU AWK 5.0.0, GNU Bash 5.0.7, GNU Binutik 2.32. GNU Consiler Collection (GCC) 9.1.0. GNU Consults 8.31. GNU Dafatile 3.7. GNU Findutile 4.6.0 199-e36: GNU Green 13. GNU Gain 1.10. GNU Integer Set Library 0.18. GNU Librord 2.4.6, GNU M4 1.4.18, GNU Make 4.2.90, GNU Multiple Precision Arithmetic Library 6.1.2, GNU Multiple Precision Jubby 4.0.2. GNU NCURSES 6.1. GNU Readline 8.0. GNU 120.1 ONU Which 2.21 GPL Chastocaist 9.26 HDES Sherey izee vib. Librae 1.6.37. Librief 4.0.10. Lain 1.20. Metastore (forked) 1.1.2.23-fr9170b. OnenBLAS 0.3.5. Onen MPI 4.0.1. OrenSSL111a PatchELE0.9, nke-config.0.29.2, Pethon 3.7.3, 2018). Oxder 0 10.0. Cython 0 29.6 (Rehnel et al. 2011). h5my 2.9.0. Kiwisolver 1.0.1, Matpiothb 3.0.2 (Hunter 2007), Numpy 1.16.2 (use der Welt et al. 2011), eksemplie 1.5.1. Polleminer 2.3.1, python-dateutil 2.8.0, Scipy 1.2.1 (Oliphant 2007; Millman the PDF using the following packages: hiber 2.12, hiber 2.12. hibiter 3.12 hibbiter 3.12 cartion 2018-10-05 cartion 2018datetime 2.60, datetime 2.60, ec 1.0, ec 1.0, etcolbox 2.5f, etcol-3.05. fontaxes 1.0d. fontaxes 1.0d. footmise 5.5h. footmise 5.5h. fn 2.1d. fn 2.1d. Journey 1.0. Journey 1.0. means 1.554 means 1.554 pef 3.1.2, pef 3.1.2, pefplots 1.16, pefplots 1.16, preprint 2011, 3.14159265, texavre 2.501, texavre 2.501, times 2016-06-24, times 2016-06-24, titlesec 2 10.2, titlesec 2 10.2, txfonts 2016DRAFT PAPER, nonci (pp), Year Month day

#### 06-24, txiouts 2016-06-24, ularn 2016-06-24, ularn 2016-06-24, xoolor 2.12, xoolor 2.12, xkeynal 2.7n and skeyval 2.7n. Wo are very grateful to all their creatence for freely providing this necessary infrastructure. This research (and many others) would not be notsible workfour them.

#### References

Akhighi, M. and T. Johowa (Sept. 2015). ApJ, 5 (20), 1, Annyo Collaboration et al. (OC) 2015). AdA, 5 (9), A33, Annyo Collaboration et al. (Sep. 2015). AdA, 5 (9), A33, Annyo Collaboration et al. (Sep. 2016). AdA, 5 (9), A13, Binens, K. et al. (New 2017). AdA, 6 (0), A1, 16 Binens, K. et al. (New 2017). AdA, 6 (0), A1, Binens, K. J. and N. 2011). COSE, 10, 30, Binens, K. J. and M. Arratis, Obse. 2011). COSE, 13, 9 Objews, T. B. (Mry 2007). COSE, 9, 49. Vonder Wells, See and Obse. 2011). COSE, 13, 22. YOUR NAME IT AL.

# 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 (Persistent IDentifier if available)
- Download URL (if available).
- checksum (e.g. md5sum, sha512sum) to check integrity (calculate with standard tools)

All inputs are downloaded using the given  $\mathsf{PID}/\mathsf{URL}$  when necessary (during the analysis).

Checksums 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).



▲日▼▲□▼▲ヨ▼▲ヨ▼ ヨークタペ

#### Input data source and integrity is documented and checked

#### Stored information about each input file:

- PID (Persistent IDentifier if available)
- Download URL (if available).
- checksum (e.g. md5sum, sha512sum) to check integrity (calculate with standard tools)

All inputs are downloaded using the given  $\mathsf{PID}/\mathsf{URL}$  when necessary (during the analysis).

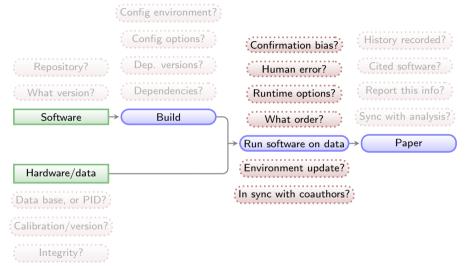
Checksums are checked to make sure the download was done properly or the file is the same (hasn't changed on the server/source).

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



▲日▼▲□▼▲ヨ▼▲ヨ▼ ヨークタル

# 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 well-checked, free-licensed documentation (for example GNU Make's manual).

File Edit Options Buffers Tools Makefile Help							
💽 🛅 📓 🗶 🔛 Save 🥠 Undo 🕌 🖷 🎁 🔍							
\$(ibidir)/gevolution-\$(gevolution-version): \$(ibidir)/gsl-\$(gsl-v							
ersion) \							
\$(ibidir)/fftw-\$(fftw-version) \							
\$(ibidir)/hdf5-\$(hdf5-version) \							
\$(ibidir)/patch-\$(patch-version) \							
\$(ibidir)/cosmdist-\$(cosmdist-version)							
\$(idir)/share ## Autoreconf is needed when inhomog is somewhat old,							
## e.g. built for debian/oldstable, while maneage uses							
## much more recent versions of the autotools.							
tarball=gevolution-\$(gevolution-version).tar.gz							
U: high-level.mk 26% L594 Git-gevcurvtest [(GNUmakefile)							
rm -rf cmake-\$(cmake-version)							
echo "CMake \$(cmake-version)" > \$@							
\$(ibidir)/cosmdist-\$(cosmdist-version): \$(ibidir)/qsl-\$(qsl-versi							
on)							
tarball=cosmdist-\$(cosmdist-version).tar.gz							
\$(call import-source, \$(cosmdist-url), \$(cosmdist-checksu -							
em), \							
v\$(cosmdist-version).tar.qz) U: high-level.mk 55% L1226 Git-gevcurvtest [(GNUmakefile)] 19							
\$(call gbuild, gperf-\$(gperf-version), static)							
echo "GNU gperf \$(gperf-version)" > \$@							
\$(ibidir)/gsl-\$(gsl-version):							
tarball=gsl-\$(gsl-version).tar.lz							
<pre>\$(call import-source, \$(gsl-url), \$(gsl-checksum))</pre>							
\$(call gbuild, gsl-\$(gsl-version), static)							
U: high-level.mk 30% L675 Git-gevcurvtest [(GNUmakefile)] 19							
Mark set							

# 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 well-checked, free-licensed documentation (for example GNU Make's manual).



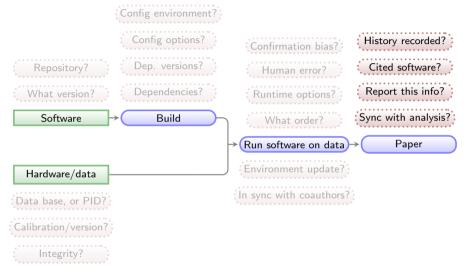
# Reproducible science: Maneage is managed through a Makefile

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 well-checked, free-licensed 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.

Here is a portion of the Borkowska & Roukema (2022) abstract and its LATEX source (arXiv:2112.14174).

#### File Edit Options Buffers Tools TeX Text Help

💽 🛅 🗮 🗶 🔛 Save 🥠 Undo 🕌 🖷 🏥 🔍

{We find that {\inhomogname} allows emergent volu we evolution correctly at first order through to the cur wrent epoch.

For initial conditions with a resolution of \$N=\\croot InhomogHTwEightvalue^3\$ particles and an initial nonvzero extrinsic curvature invariant \$\initial{\inVI} = \Init PerturbInhomogPlusvalue\$, {\inhomogname} matches an exact Friedmannian solution to \postrefereechange s{\$\InhomogAccuracyPercentEdSzdzzouHTwEightvalue \$\% (Einstein-de-Sitter, EdS) or \$\InhomogAccuracyPer' ccentLCDMzdzzouHTwEightvalue\$\%) (\$\Lambda\$CDM), %The Poisson-gauge formalism of {\gevolutionname}}

allows both growing and decaying modes for emergen + -:--- paper.tex 4% L85 Git-gevcurvtest [(LaTeX)] 21:20 equations that convert from the perturbed reference solution to the effective solution. We find that INNOMOG allows emergent volume evolution correctly at first order through to the current epoch. For initial conditions with a resolution of  $N=128^3$  particles and an initial non-zero extrinsic curvature invariant  $I_i=0.001$ , INHOMOG matches an exact Friedmannian solution to -0.0058% (Einstein-de Sitter, EdS) or -0.0033% (ACDM). We find that GEVOLUTION models the decaying mode to fair accuracy, and excludes the growing mode by construction. For  $N=128^3$  and an initial scalar potential  $\Phi=0.001$ , GEVOLUTION is accurate for the decaying mode to 0.012% (EdS) or 0.013% (ACDM). We conclude that this special case of an exact non-linear solution for a perturbed Friedmannian model provides a robust calibration for relativistic cosmological simulations.

Page 1 7 m 2 ? of 50 125%

Ouit

# Values in final report/paper

All analysis results (numbers, plots, tables) written in paper's PDF as LTFX macros. They are thus updated automatically on any change.

Here is a portion of the Borkowska & Roukema (2022) abstract and its LATEX source (arXiv:2112.14174).

#### File Edit Options Buffers Tools TeX Text Help

Mark set

📭 🛄 💥 🔛 Save 🥠 Undo 🕌 🖷 🏥 🔾

{We find that {\inhomogname} allows emergent volu me evolution correctly at first order through to the cur rent epoch.

For initial conditions with a resolution of \$N=\Ncroot InhomogHTwEightvalue^3\$ particles and an initial nonzero extrinsic curvature invariant \$\initial{\invI} = \Init PerturbInhomogPlusvalue\$, {\inhomogname} matches an exact Friedmannian solution to \postrefereechange s/\$\InhomogAccuracyPercentEdSzdzzouHTwEightvalue \$\% (Einstein--de~Sitter, EdS) or \$\InhomogAccuracyPer centLCDMzdzzouHTwEightvalue\$\%} (\$\Lambda\$CDM). %The Poisson-gauge formalism of {\gevolutionname} allows both growing and decaving modes for emergen -:--- paper.tex 4% L85 Git-gevcurvtest [(LaTeX)] 21:20

equations that convert from the perturbed reference solution to the effective solution. We find that INHOMOG allows emergent volume evolution correctly at first order through to the current epoch. For initial conditions with a resolution of  $N = 128^3$ particles and an initial non-zero extrinsic curvature invariant  $I_{4} = 0.001$ , INHOMOG matches an exact Friedmannian solution to -0.0058% (Einstein-de Sitter, EdS) or -0.0033% (ACDM). We find that GEVOLUTION models the decaying mode to fair accuracy, and excludes the growing mode by construction. For  $N = 128^3$  and an initial scalar potential  $\Phi = 0.001$ , GEVOLUTION is accurate for the decaying mode to 0.012% (EdS) or 0.013% (ACDM). We conclude that this special case of an exact non-linear solution for a perturbed Friedmannian model provides a robust calibration for relativistic cosmological simulations. Page 1 7 m 2 ? of 50 125%

#### 

Ouit

# Analysis step results/values stored in a single build directory.

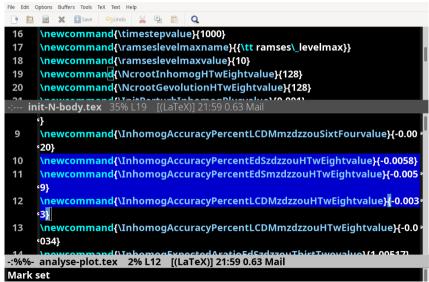
All LATEX macros are output in a single directory.

File Edit Options Buffers Tools TeX Text Help	
🕐 🛄 📓 🕱 🖳 Save 🥱 Undo 👷 🖏 🎼 🔍	
16 \newcommand{\timestepvalue}{1000}	
17 \newcommand{\ramseslevelmaxname}{{\tt ramses\_levelmax}}	
18 \newcommand{\ramseslevelmaxvalue}{10}	
19 \ <mark>newcommand</mark> {\NcrootInhomogHTwEightvalue}{128}	
20 \newcommand{\NcrootGevolutionHTwEightvalue}{128}	
-: init-N-body.tex 35% L19 [(LaTeX)] 22:00 0.64 Mail	
8 }	
9 \newcommand{\InhomogAccuracyPercentLCDMmzdzzouSixtFourvalue}	{-0.00
°20}	
10 \newcommand{\InhomogAccuracyPercentEdSzdzzouHTwEightvalue}{-0	.0058}
11 \newcommand{\InhomogAccuracyPercentEdSmzdzzouHTwEightvalue}{	-0.005 °
·9}	
12 \newcommand{\InhomogAccuracyPercentLCDMzdzzouHTwEightvalue}	-0.003°
·3}	
13 \InhomogAccuracyPercentLCDMmzdzzouHTwEightvalue	}{-0.0 °
*034 <b>}</b>	
14 Inswcommand() Inhomog Expected Aratio EdSzdzzou ThirtTwovalue) (1.00	5477
-:%%- analyse-plot.tex 2% L11 [(LaTeX)] 22:00 0.64 Mail	

▲□▶ ▲圖▶ ▲필▶ ▲필▶ - ヨー のへで

# Analysis step results/values stored in a single build directory.

All LATEX macros are output in a single directory.



#### Analysis results stored as LATEX macros

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

```
# Numbers for dettf.tex:
sant=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)}' $name)
    dettfgnt=$(awk '/guantile 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 sobase"dettfnum" sdettfnum
    addtexmacro $obase"dettfmax" $dettfmax
    addtexmacro $obase"dettfont" $dettfont
    # Find the smallest S/N quantile:
    sqnt=$(echo " " | awk '{if('$dettfqnt'<'$sqnt') print '$dettfqnt'}')</pre>
for base in 4 onelarge sensitivity3
do dettfhist stexdir/dettf/ sbase: done
addtexmacro dettfsmallestsngnt $sgnt
```

▲□▶ ▲□▶ ▲□▶ ▲□▶ ▲□ ● ● ●

#### Analysis results stored as LATEX macros

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

```
# Numbers for dettf.tex:
sant=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)}' $name)
    dettfgnt=$(awk '/guantile 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"dettfont" $dettfont
    # Find the smallest S/N quantile:
    sqnt=$(echo " " | awk '{if('$dettfqnt'<'$sqnt') print '$dettfqnt'}')</pre>
for base in 4 onelarge sensitivity3
do dettfhist stexdir/dettf/ sbase: done
addtexmacro dettfsmallestsngnt $sgnt
```

# Makefiles (.mk) keep contextually separate parts of the project, all imported into top-make.mk

top-make.mk							
initialize.mk	download.mk	format.mk	demo-plot.mk				
verif	ŷ.mk	paper.mk					

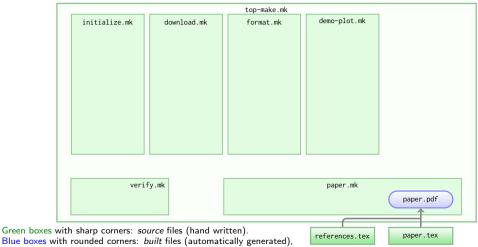
Green boxes with sharp corners: *source* files (hand written). Blue boxes with rounded corners: *built* files (automatically generated), built files are shown in the Makefile that contains their build instructions.

# The ultimate purpose of the project is to produce a paper/report (in PDF).

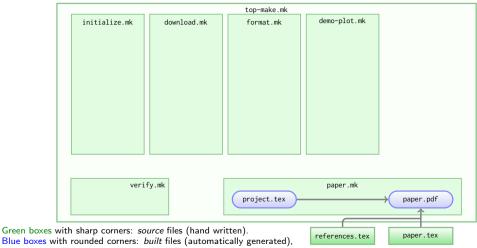
top-make.mk							
initialize.mk	download.mk	format.mk	demo-plot.mk				
veri	fy.mk		paper.mk	paper.pdf			

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.

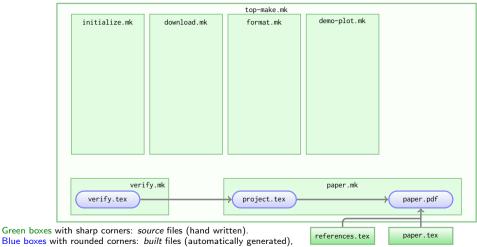
## The narrative description, typography and references are in paper.tex & references.tex.



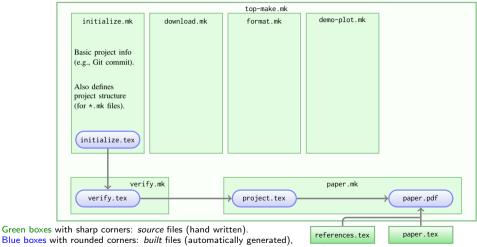
# Analysis outputs (blended into the PDF as LATEX macros) come from project.tex.



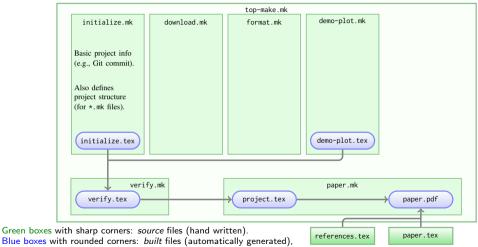
# But analysis outputs must first be verified (with checksums) before entering the report/paper.



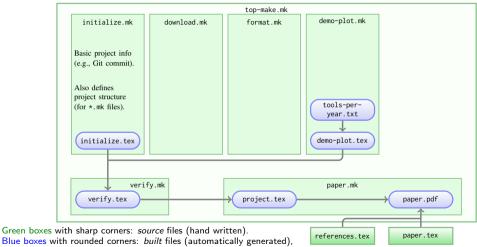
# Basic project info comes from initialize.tex.



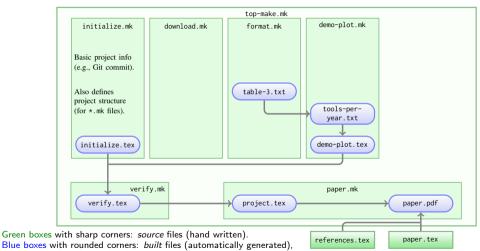
# The paper includes some information about the plot.



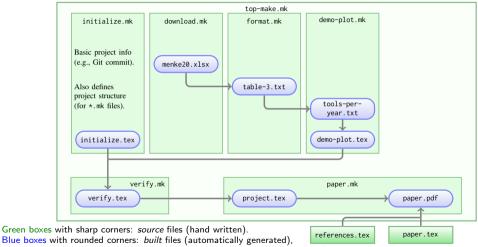
### The final plotted data are calculated and stored in tools-per-year.txt.



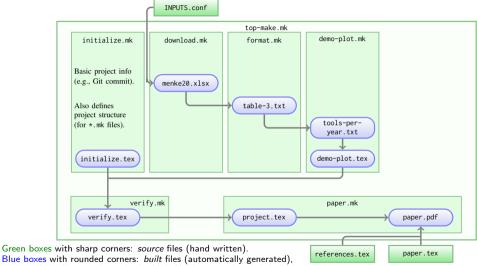
# The plot's calculation is done on a formatted sub-set of the raw input data.



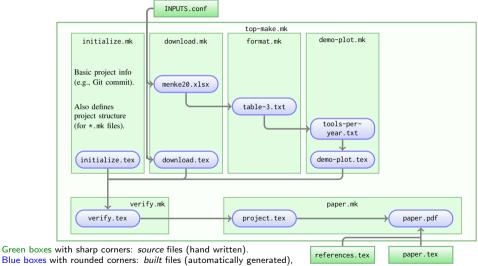
### The raw data that were downloaded are stored in XLSX format.



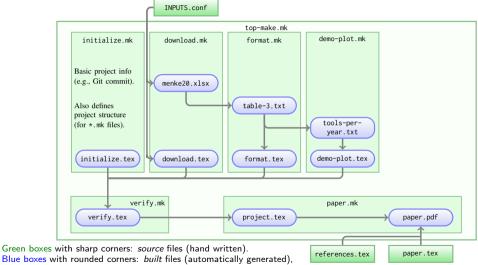
### The download URL and a checksum to validate the raw inputs, are stored in INPUTS.conf.



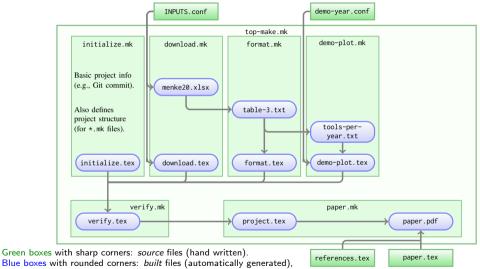
#### We also need to report the URL in the paper...



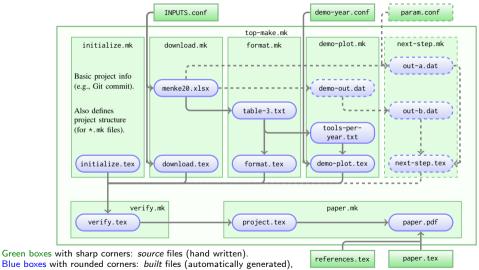
## Some general info about the full dataset may also be reported.



We report the number of papers studied in a special year, desired year is stored in .conf file.



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



The whole project is a directed graph (codifying the data's lineage).

Every file (source or built) is a node in the graph (connected to others). (The links/connections/dependencies between the nodes, defined by the Makefiles: \*.mk)

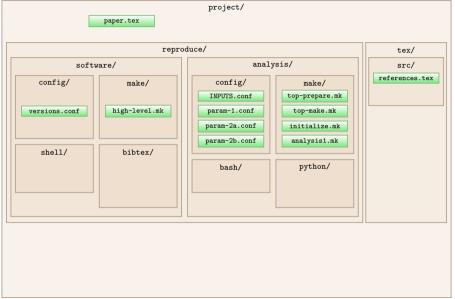
- There are two types of nodes/files:
  - Source nodes (\*.conf and paper.tex) only have an outward link.
  - Built files always have inward and (except paper.pdf) outward link(s).

All built files ultimately originate from a \*.conf file, ... and ultimately conclude in paper.pdf.

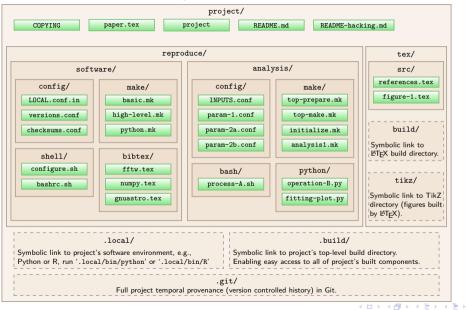
# Benefits of using Make

- Make can parallelize the analysis: Make knows which steps are independent and will run them at the same time.
- Make can automatically detect a change and will re-do only the affected steps. (for example to change the multiple of sigma in a configuration file to see its effect)
- Easily backtrace any step (without needing to remember!). (very useful to find problems/improvements)
- ▶ The above will speed up your work, and encourage experimentation on methods.
- Make is available on any system: many people are already familiar with it.
- And again: its all in plain text! (doesn't take much space, easy to read, distribute, parse automatically, or archive)
- Recall that the project's software installation was also managed in Make.

# Files organized in directories by context (here are some of the files discussed before)

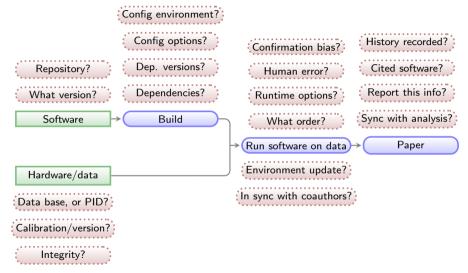


# Files organized in directories by context (now with other project files and symbolic links)



ъ

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



A D > A B > A B > A B >

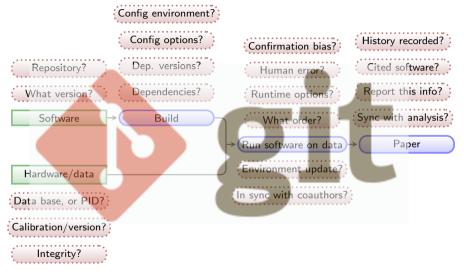
э.

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).



イロト 不得 トイヨト イヨト

э.

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.

► The project (answers to questions above) will evolve.

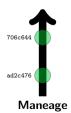


► The project (answers to questions above) will evolve.



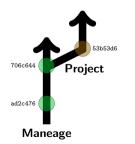
Today

Each point of project's history is recorded with Git.



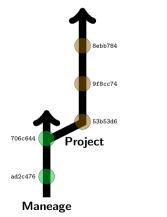
- Each point of project's history is recorded with Git.
- New project: a branch from the template. Every commit contains the following:
  - Instructions to download, verify and build software.
  - Instructions to download and verify input data.
  - Instructions to run software on data (do the analysis).

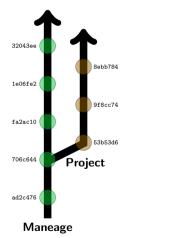
Narrative description of project's purpose/context.



- Each point of project's history is recorded with Git.
- New project: a branch from the template. 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).

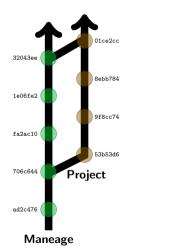
- Narrative description of project's purpose/context.
- Research progresses in the project branch.





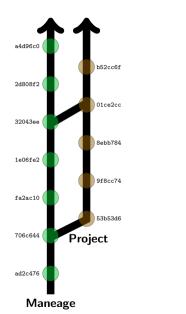
- Each point of project's history is recorded with Git.
- New project: a branch from the template. Every commit contains the following:
  - Instructions to download, verify and build software.
  - Instructions to download and verify input data.
  - Instructions to run software on data (do the analysis).

- Narrative description of project's purpose/context.
- Research progresses in the project branch.
- Template will evolve (improved infrastructure).

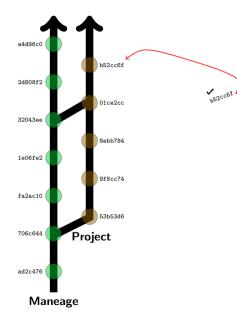


- Each point of project's history is recorded with Git.
- New project: a branch from the template.
   Every commit contains the following:
  - Instructions to download, verify and build software.
  - Instructions to download and verify input data.
  - Instructions to run software on data (do the analysis).
  - 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.

▲□▶ ▲□▶ ▲□▶ ▲□▶ □ の00

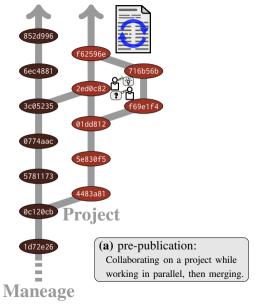


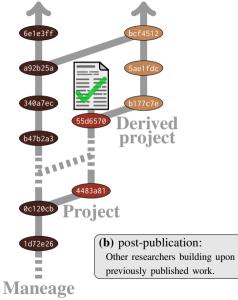
- Each point of project's history is recorded with Git.
- New project: a branch from the template. Every commit contains the following:
  - Instructions to download, verify and build software.
  - Instructions to download and verify input data.
  - Instructions to run software on data (do the analysis).
  - 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).



- Each point of project's history is recorded with Git.
- New project: a branch from the template. 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).

Any Git-based workflow is possible.





# 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 tiny:  $\lesssim 1$  Mb. Borkowska & Roukema (2022) 550 kb + Peper, Roukema & Bolejko (2023) 740 kb would fit on a floppy disk

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:2112.14174).
- Zenodo: Along with all the input datasets (many gigabytes) and software (for example zenodo.6794222) and given a unique DOI.
  - ... and put links to data in paper! See the caption of Fig. 3 in Borkowska & Roukema (2022).
- Software Heritage: to archive the full version-controlled history of the project. (for example swh:1:dir:33fea87068c1612daf011f161b97787b9a0df39fk)

► ... and put links to exact parts of the code! See caption of Listing 1 in the Maneage paper.

Programs [here: Scientific projects] must be written for people to read...

...and only *incidentally* for machines to *execute*.

Harold Abelson, Structure and Interpretation of Computer Programs

General outline of using this system (e.g. arXiv:2112.14174)

\$ git clone https://codeberg.org/boud/gevcurvtest # Import the project. \$ cd gevcurvtest # Enter the directory. \$ ./project --help # RTFM. Skip if brave enough.

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

#### ◆□ → ◆□ → ◆三 → ◆三 → ● ● ● ● ●

### Practical experience: peer-reviewed cosmology papers

- Peper & Roukema (2021), MNRAS, 505, 1223, "The role of the elaphrocentre in void galaxy formation"
  - https://arXiv.org/abs/2010.03742
  - frozen record: https://zenodo.org/record/4699702
  - live git: https://codeberg.org/boud/elaphrocentre
  - archived git: swh:1:rev:a029edd32d5cd41dbdac145189d9b1a08421114e
- Borkowska & Roukema (2022) CQG, 39, 215007 "Does relativistic cosmology software handle emergent volume evolution?"
  - https://arXiv.org/abs/2112.14174
  - frozen record: https://zenodo.org/record/5806027
  - live git: https://codeberg.org/boud/gevcurvtest
  - archived git swh:1:rev:d9b47736f81aff9bb8f2f359d9f0331aa923f38d
- Peper, Roukema & Bolejko (2023) MNRAS, 525, 91 "Detecting cosmic voids via maps of geometric-optics parameters"
  - https://arXiv.org/abs/2304.00591
  - frozen record: https://zenodo.org/record/8103985
  - b live git: https://codeberg.org/mpeper/lensing
  - archived git swh:1:rev:b5dff23ab8ba8c758112d5fd3f737fb6f44cd6fe
- Was it a lot of work? Yes.
- Did we feed our variations and fixes upstream to Maneage? Many, yes.
- Overall feeling: was it worth it? Yes.

#### How can you try this out?

Try to reproduce Borkowska & Roukema (2022), e.g. from SWH: swh:1:rev:d9b47736f81aff9bb8f2f359d9f0331aa923f38d (0.5 Mb; fits on a floppy disk)

Did it fully configure, run and verify? If not, then post an issue at https://codeberg.org/boud/gevcurvtest or, if relevant, post it upstream as a task or a bug:

Tasks: https://savannah.nongnu.org/tasks/?group=reproduce, e.g.

- #15739 debian-verified sources (stable || testing)
- ▶ #15363 file dates after git checkout
- #15997 safe-rm
- #15390 glibc within Maneage
- Bugs: https://savannah.nongnu.org/bugs/?group=reproduce, e.g.
  - #62879 Maneage handling of /dev/shm and required RAM
- Core Maneage: https://git.maneage.org/project.git
- Merge requests: any git server of your choice
- Description of implementation, how-to guide, recommendations: https://codeberg.org/boud/maneage\_dev/src/branch/maneage/README-hacking.md

Interactive discussion: #maneage\_community:matrix.org (e.g. https://matrix.to/#/#maneage\_community:matrix.org)

## Conclusion

Maneage (Akhlaghi+2021, CiSE 23, 82 arXiv:2006.03018) is a customisable template that does the following — all in 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.
- Only those components that need to be re-done are re-done.
- ▶ The whole project is under version control (Git) encouraging tests and experimentation.
- > The Git commit hash of the project source is printed in the paper and on output data products.

Published cosmology papers:

- Peper & Roukema (2021) MNRAS, 505, 1223, "The role of the elaphrocentre in void galaxy formation", https://doi.org/10.5281/zenodo.4699702
- Borkowska & Roukema (2022) CQG, 39, 215007 "Does relativistic cosmology software handle emergent volume evolution?", https://doi.org/10.5281/zenodo.5806027
- Peper, Roukema & Bolejko (2023) MNRAS, 525, 91 "Detecting cosmic voids via maps of geometric-optics parameters", https://doi.org/10.5281/zenodo.8103985

this pdf - full of clickable links: https://cosmo.torun.pl/~boud/Roukema20240311IANCU.pdf

ъ