

Digging out the low-surface-brightness universe, **BIG** Data, **BIG** responsibility (a practical solution for reproducible science)

Mohammad Akhlaghi

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

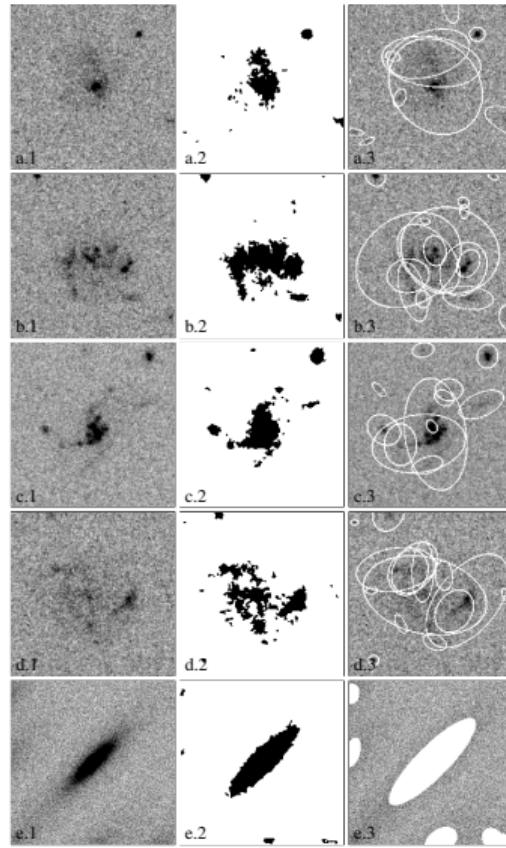


Universidad
de La Laguna

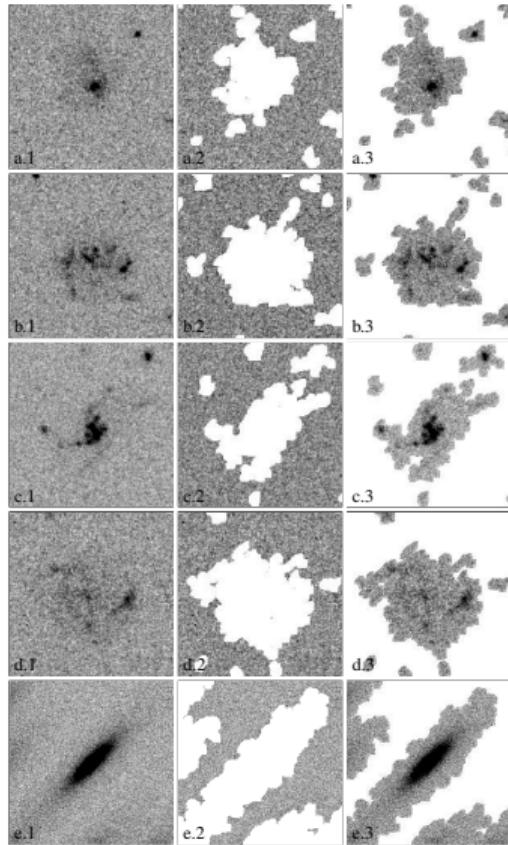
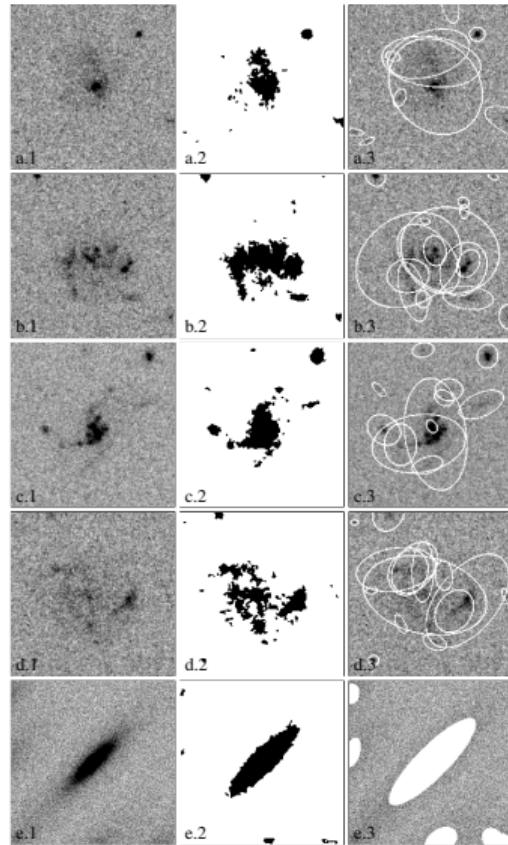


IAU Symposium 355, Session 2, July 8th, 2019
Tenerife, Spain

It all started because I wanted to study *real* galaxies...



It all started because I wanted to study *real* galaxies... So I made NoiseChisel (in Gnuastro)

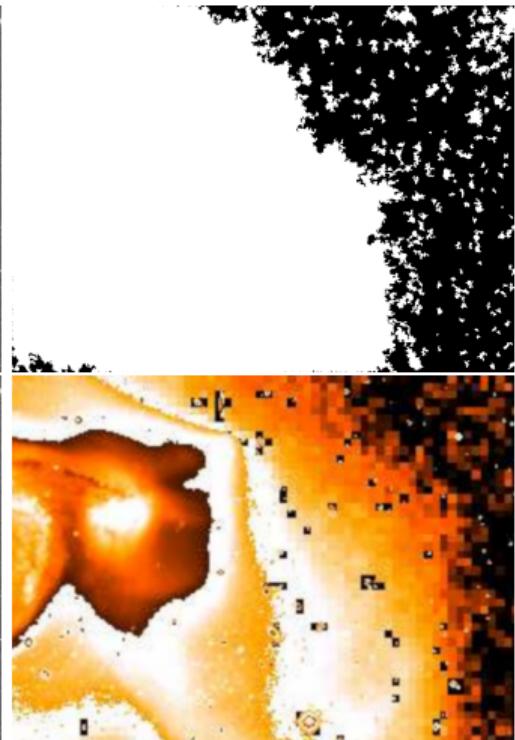
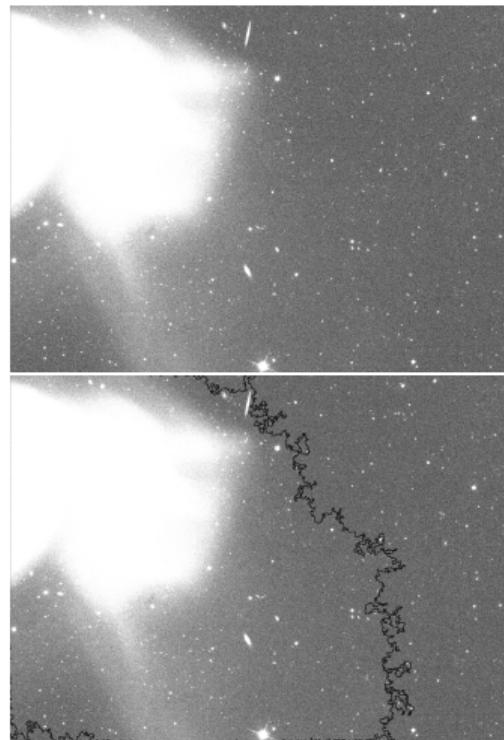


Reproducibility is critically important for low surface brightness astronomy

Example: Outer surface brightness of M51 in a single-exposure SDSS image, using NoiseChisel.

- ▶ Outer wing detected to S/N = 1/20 or ~ 30 mag/arcsec².
- ▶ Complete tutorial in manual fully describes how to derive/reproduce this result:
 - ▶ Run-time options/configuration.
 - ▶ Steps before/after NoiseChisel.
- ▶ Deep/orange image from Watkins+2015 ([arXiv:1501.04599](https://arxiv.org/abs/1501.04599)).
- ▶ Therefore:
 - ▶ Default settings **not enough**.
 - ▶ Result **not just from NoiseChisel**.
- ▶ Low-S/N analysis is hard to reproduce!

Simply reporting in your paper that “we used NoiseChisel” is **not enough** to reproduce, understand, or verify your result.



Reproducibility crisis in the sciences/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](https://arxiv.org/abs/1901.00143)).

Reproducibility crisis in the sciences/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](https://arxiv.org/abs/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](https://arxiv.org/abs/1801.08200)).

Reproducibility crisis in the sciences/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](https://arxiv.org/abs/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](https://arxiv.org/abs/1801.08200)).

Schroedinger’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](https://arxiv.org/abs/1801.02094)).

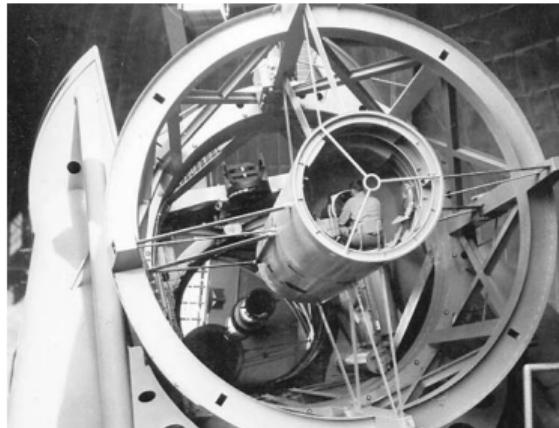


Original image from <https://www.redbubble.com>

Types of reproducibility

Hardware/Statistical reproducibility

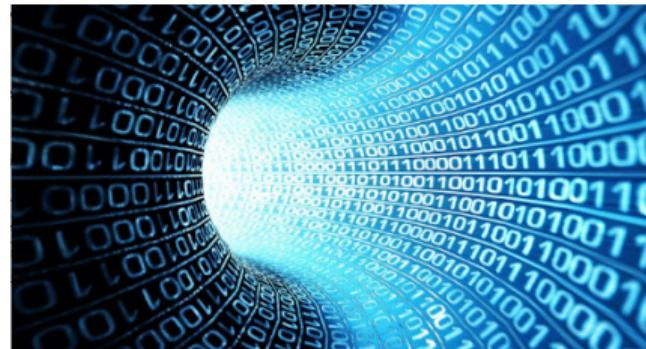
- ▶ Involves data **collection**.
- ▶ Inherently includes **measurements errors** (can never be exactly reproduced).
- ▶ Example: Raw telescope image/spectra.
- ▶ **NOT DISCUSSED HERE (Nacho's talk)**.



<http://slittlefair.staff.shef.ac.uk>

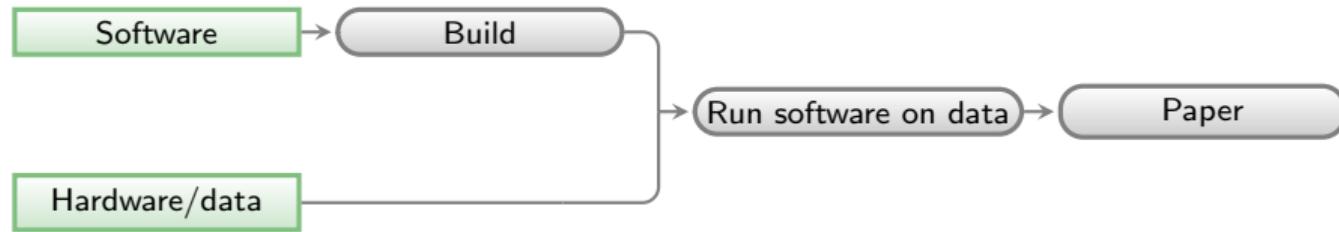
Software/Deterministic reproducibility

- ▶ Involves data **analysis**, or simulations.
- ▶ Starts **after** data is collected/digitized.
- ▶ Example: $2 + 2 = 4$ (i.e., sum of datasets).
- ▶ **DISCUSSED HERE.**

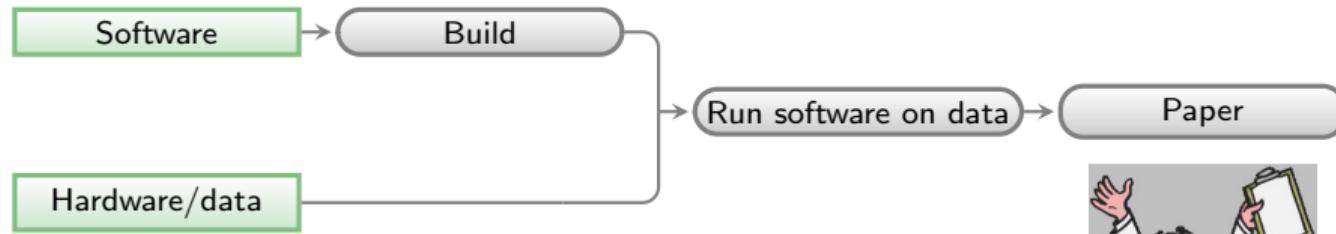


<https://tsongas.com>

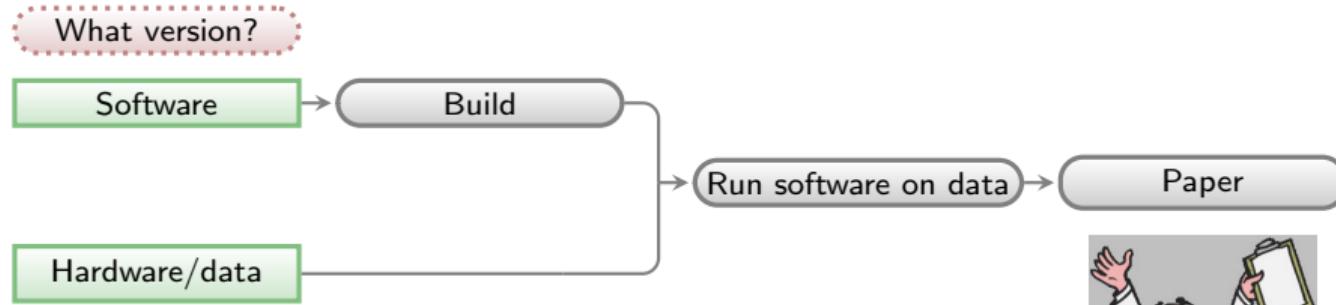
General outline of a project



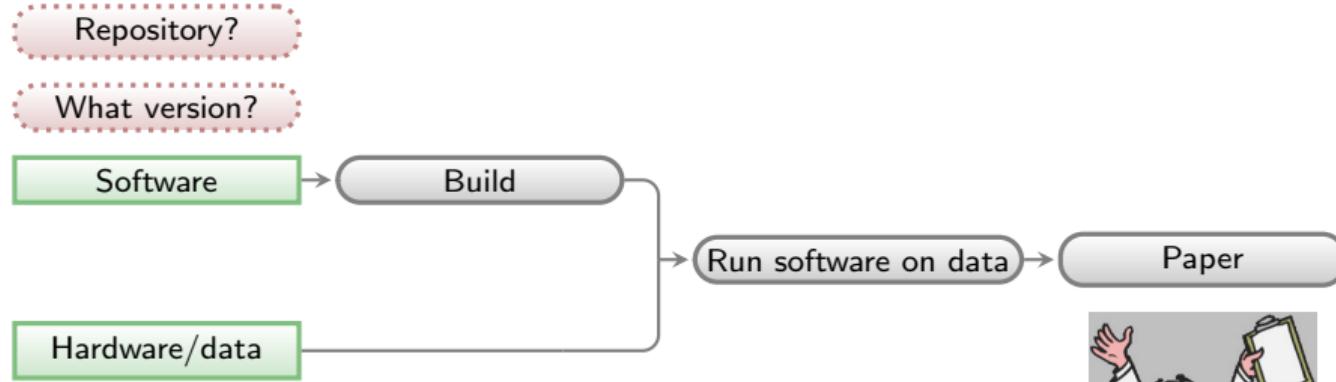
General outline of a project



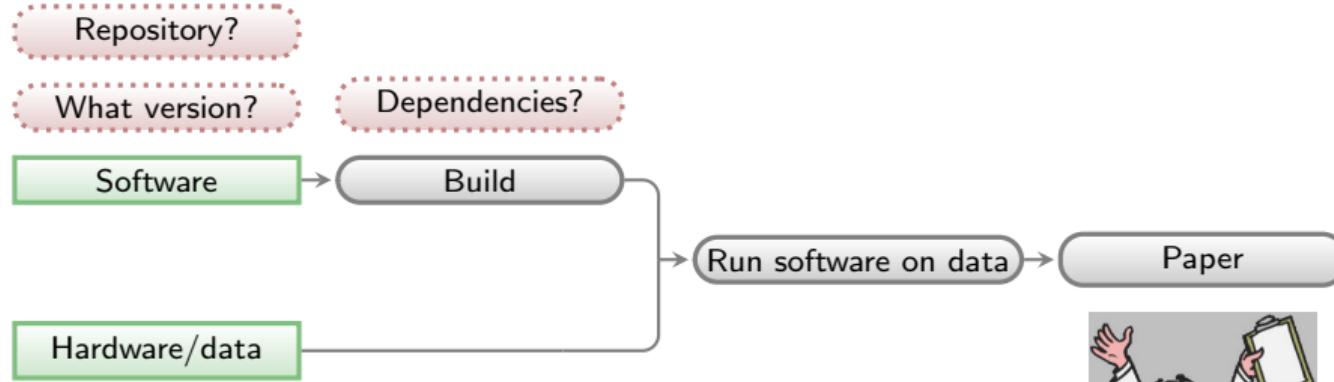
General outline of a project



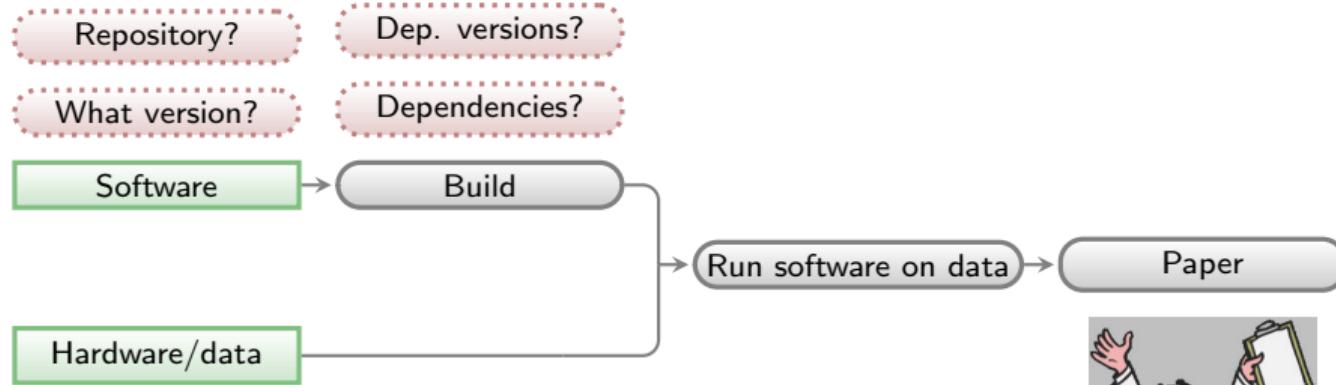
General outline of a project



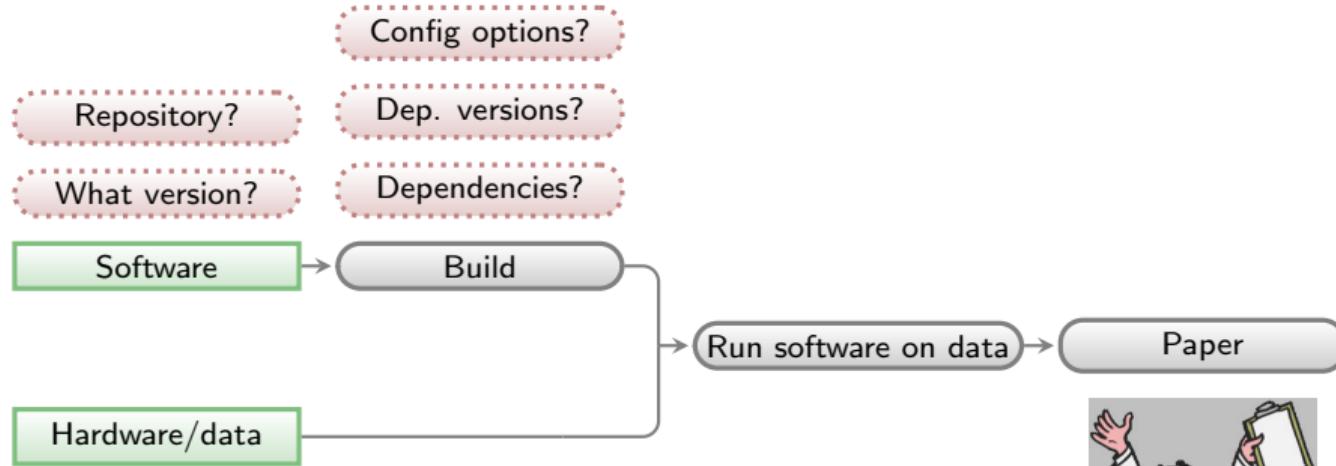
General outline of a project



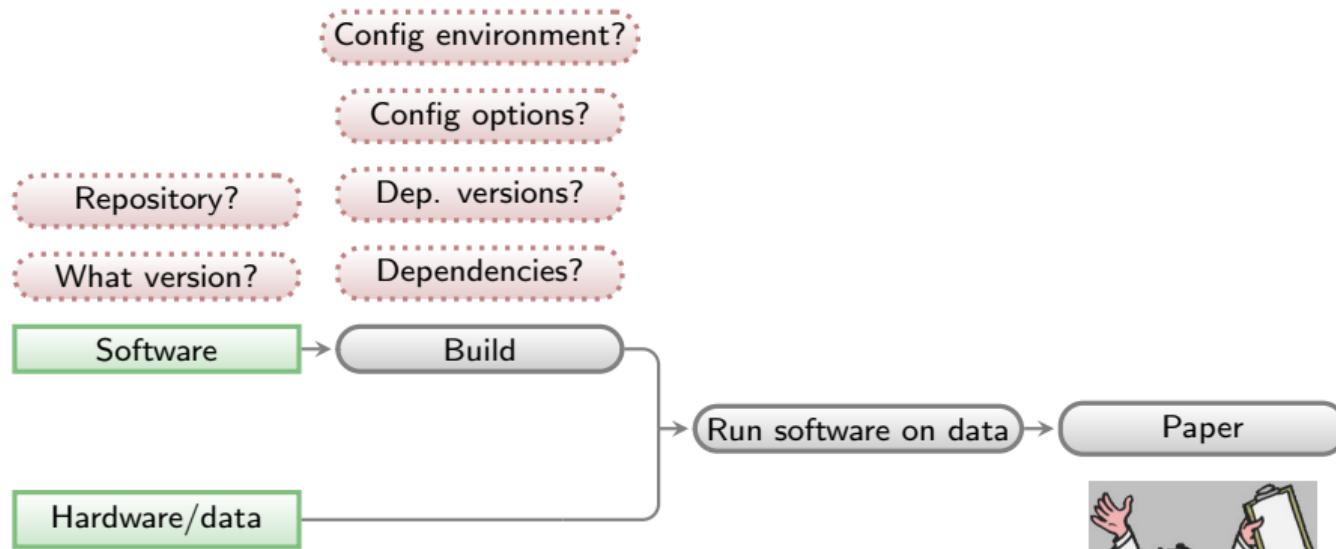
General outline of a project



General outline of a project



General outline of a project



Example: Matplotlib (a Python visualization library) build dependencies

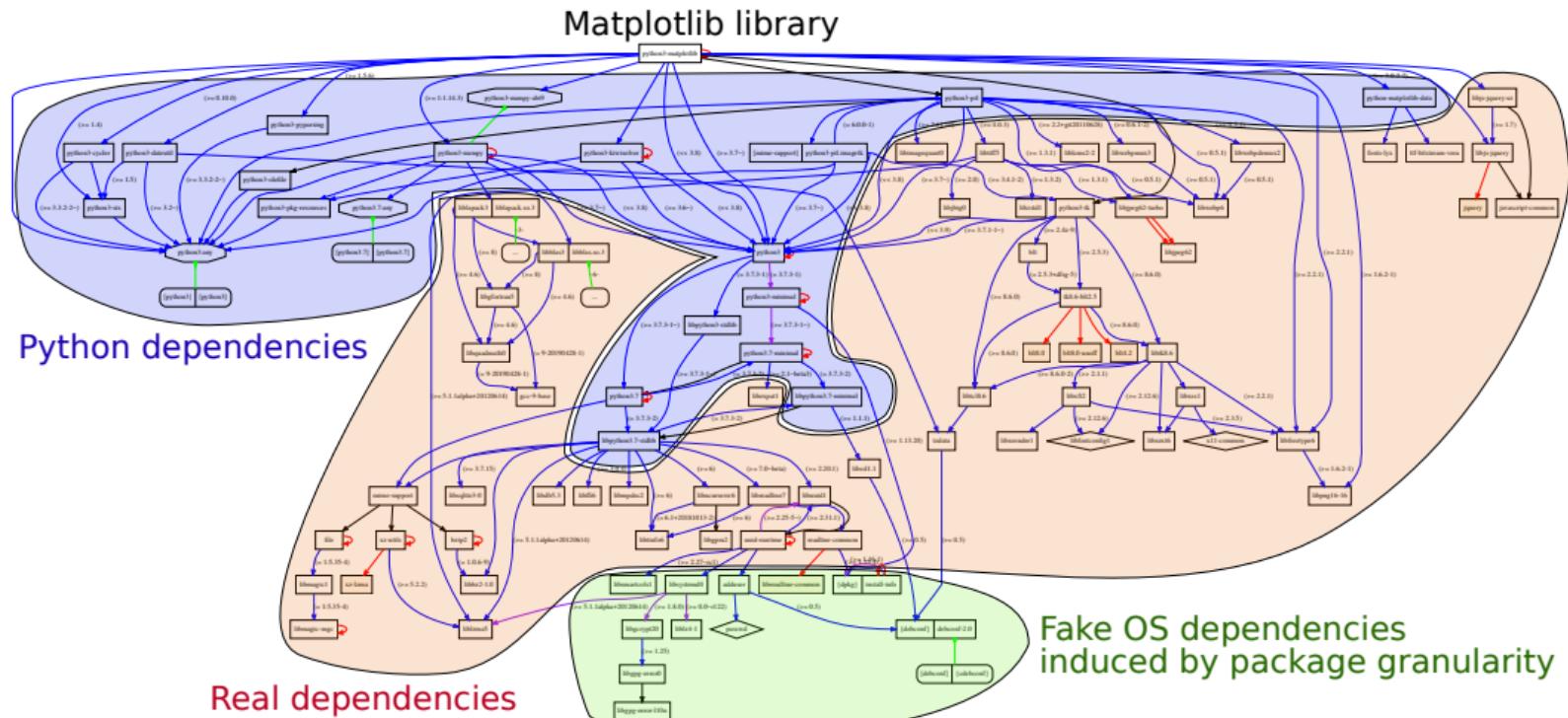
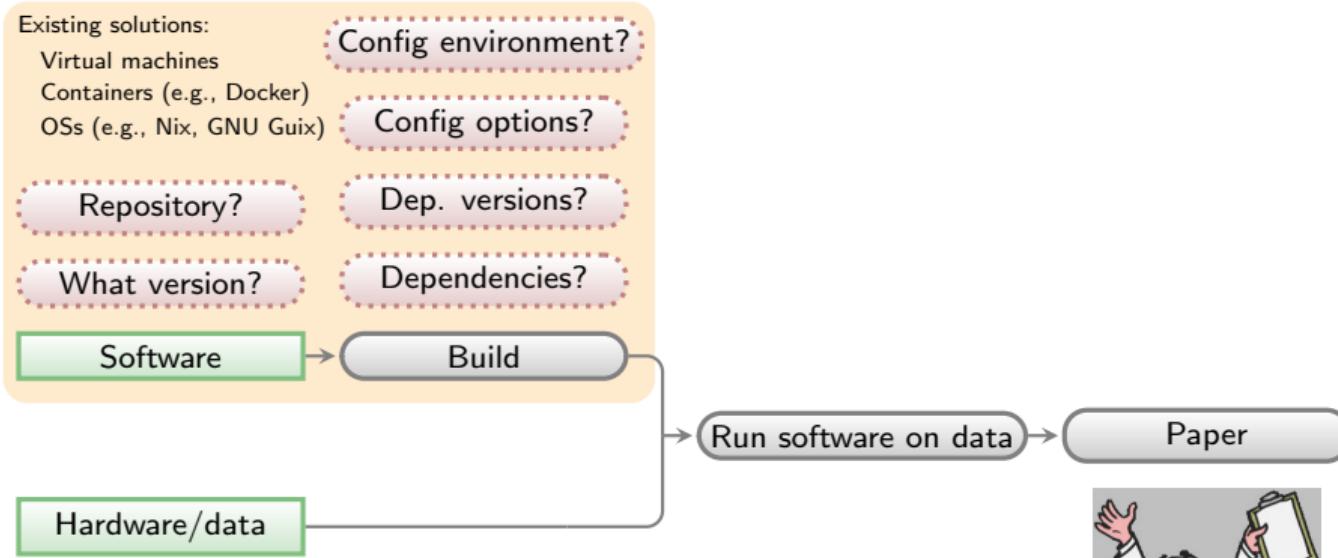
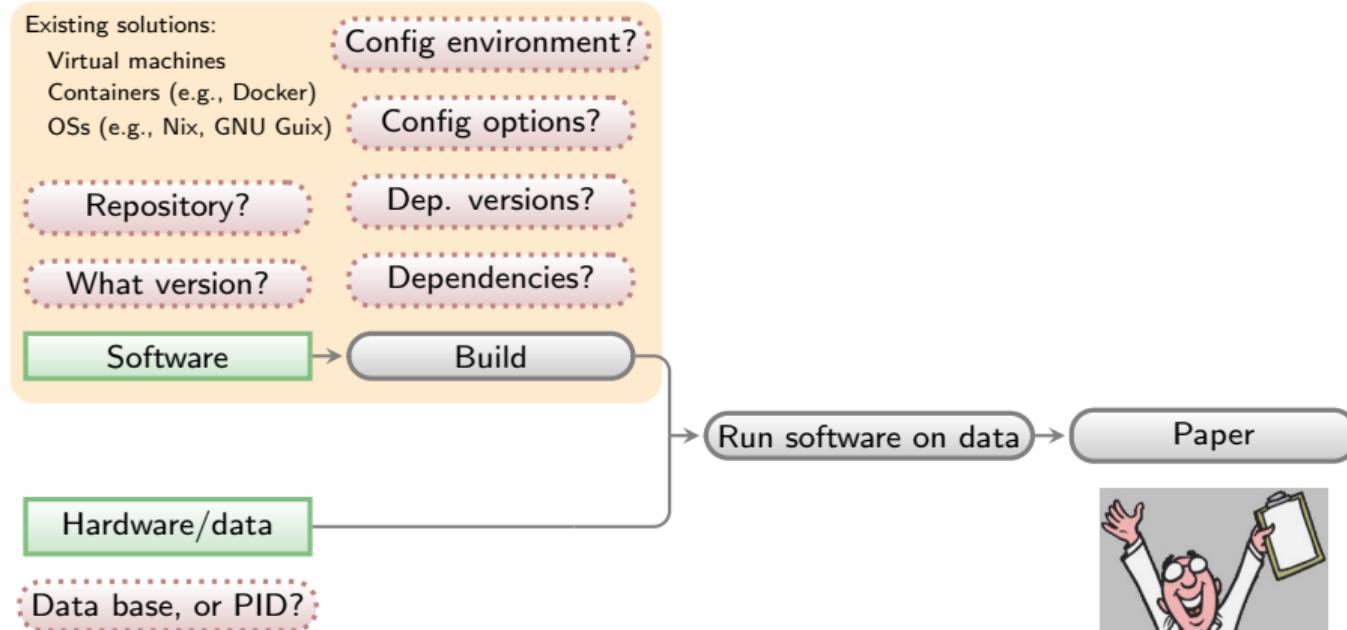


Fig. 1. Transitive dependencies of the software environment required by a simple “import matplotlib” command in the Python 3 interpreter.

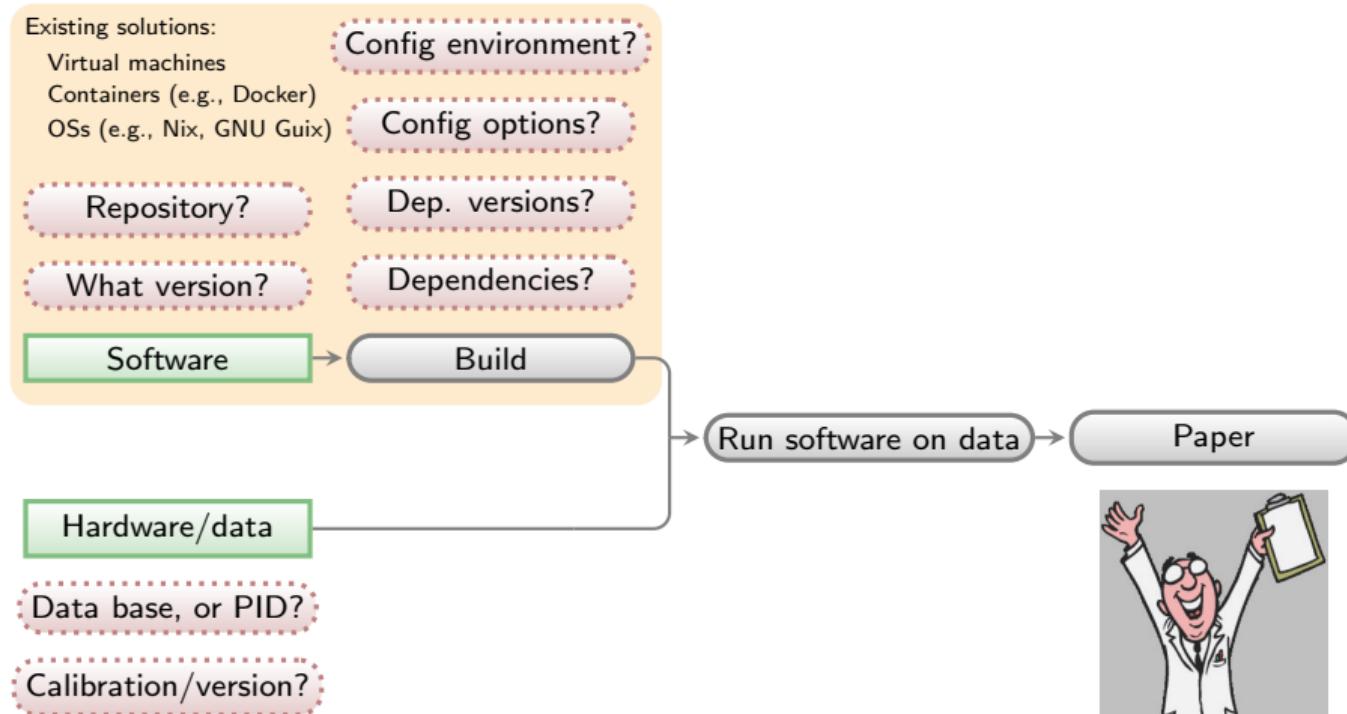
General outline of a project



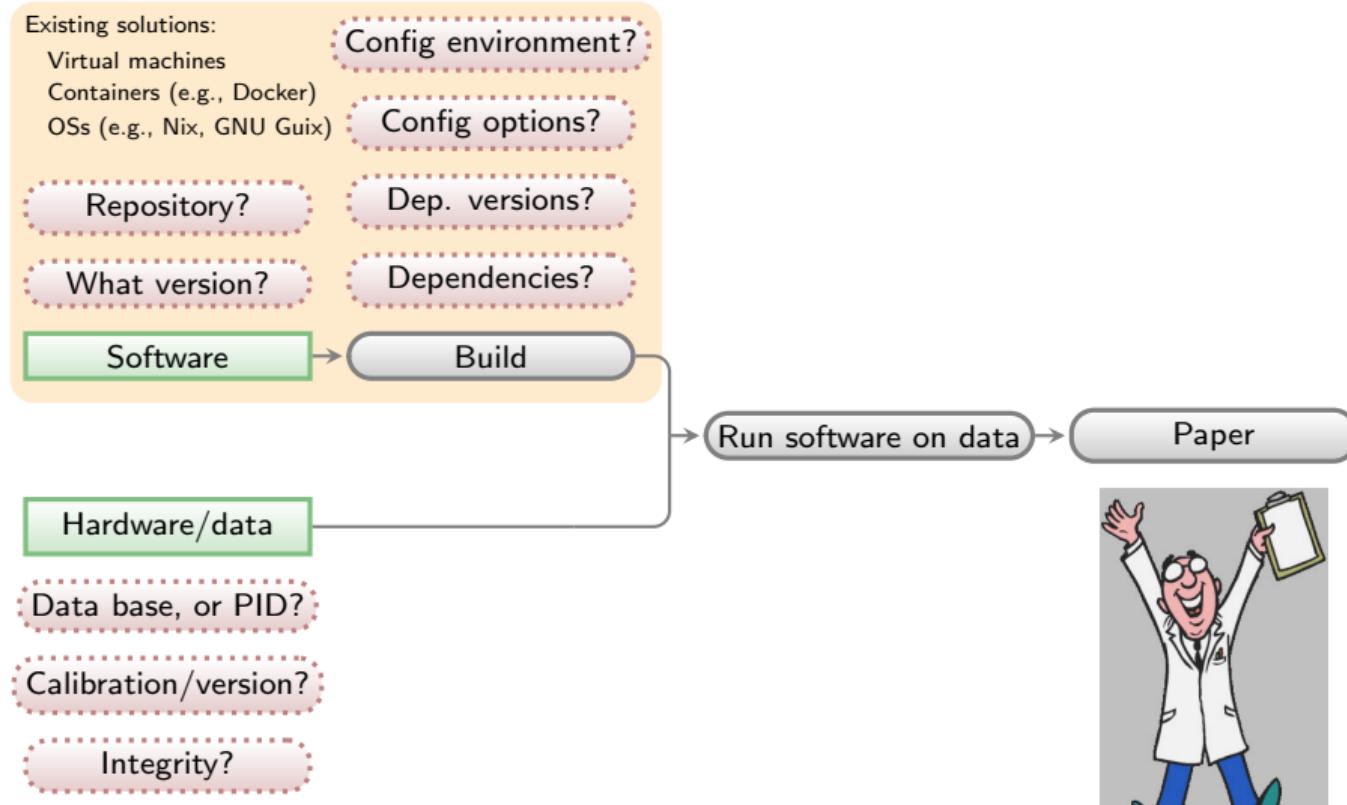
General outline of a project



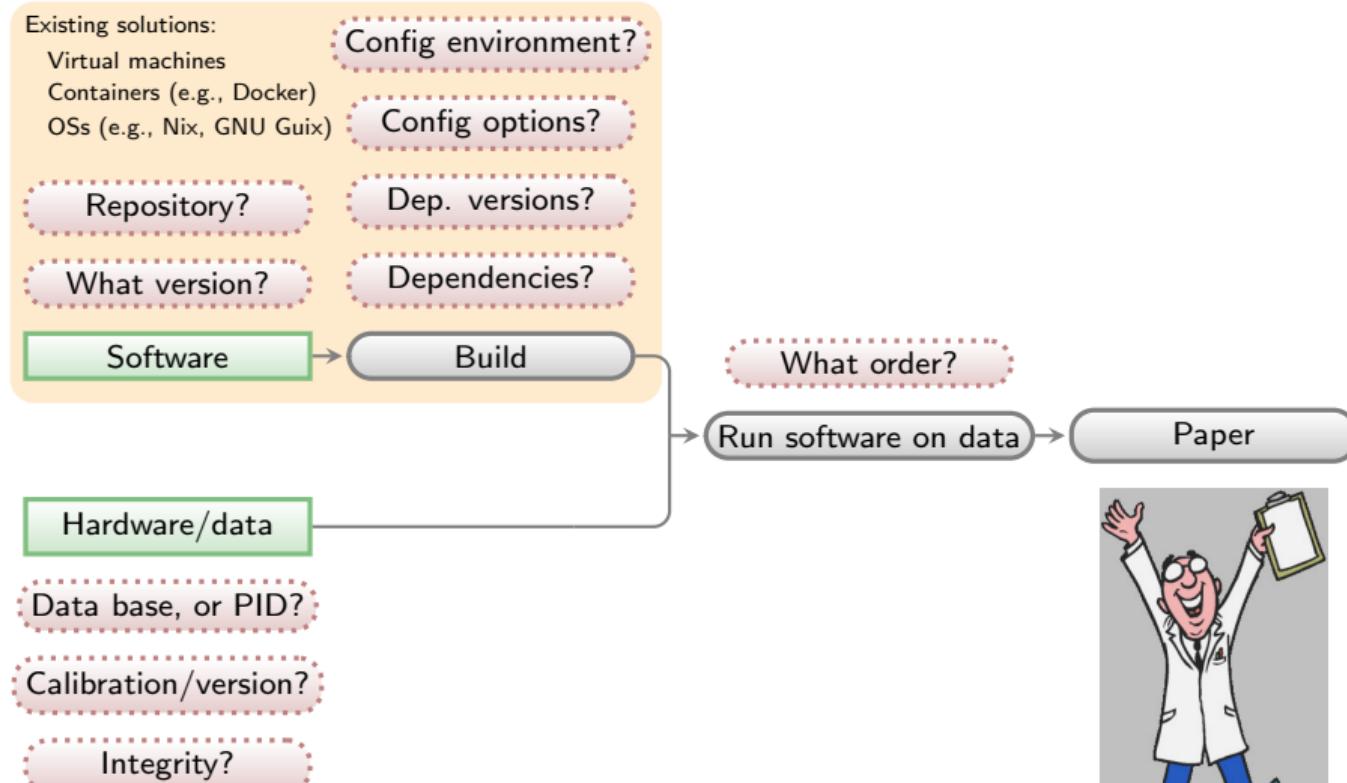
General outline of a project



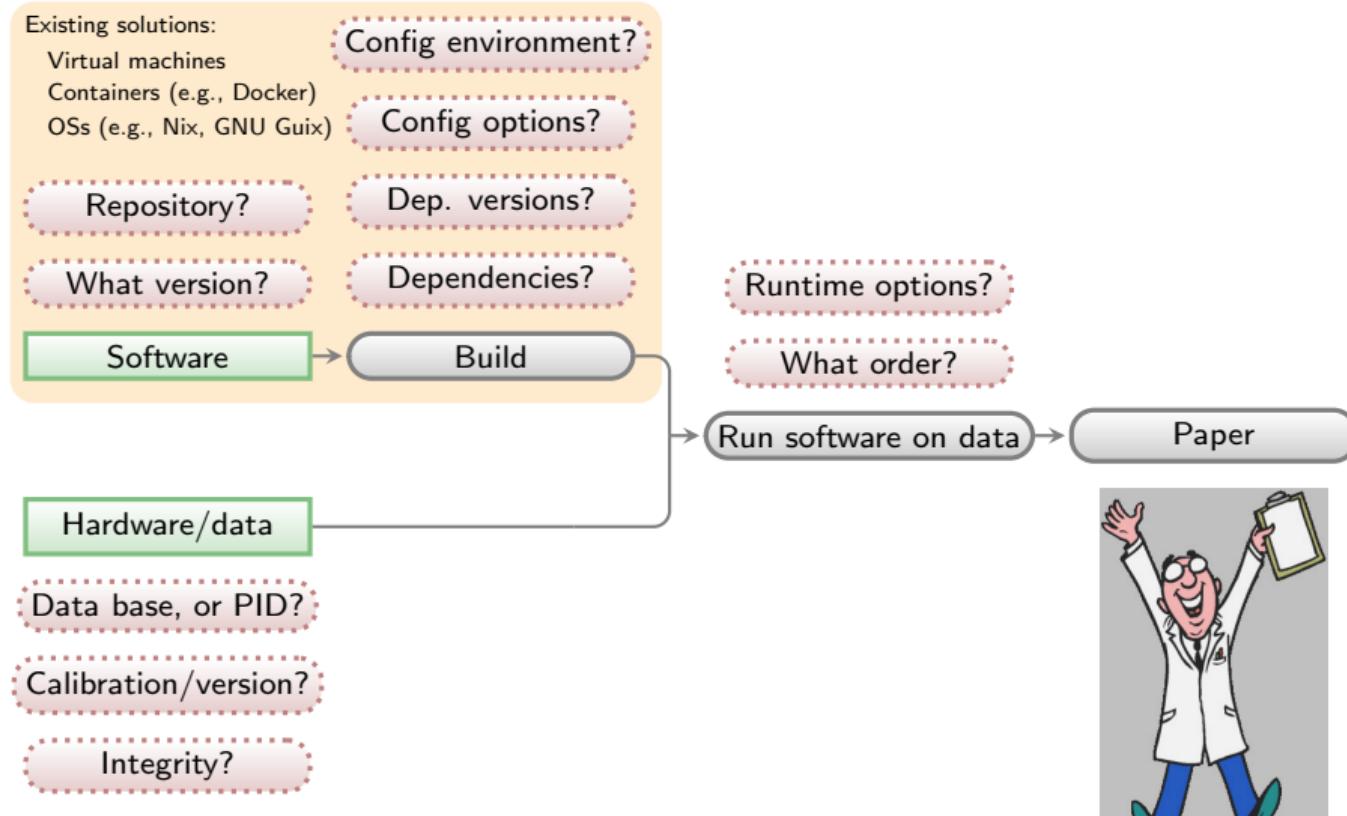
General outline of a project



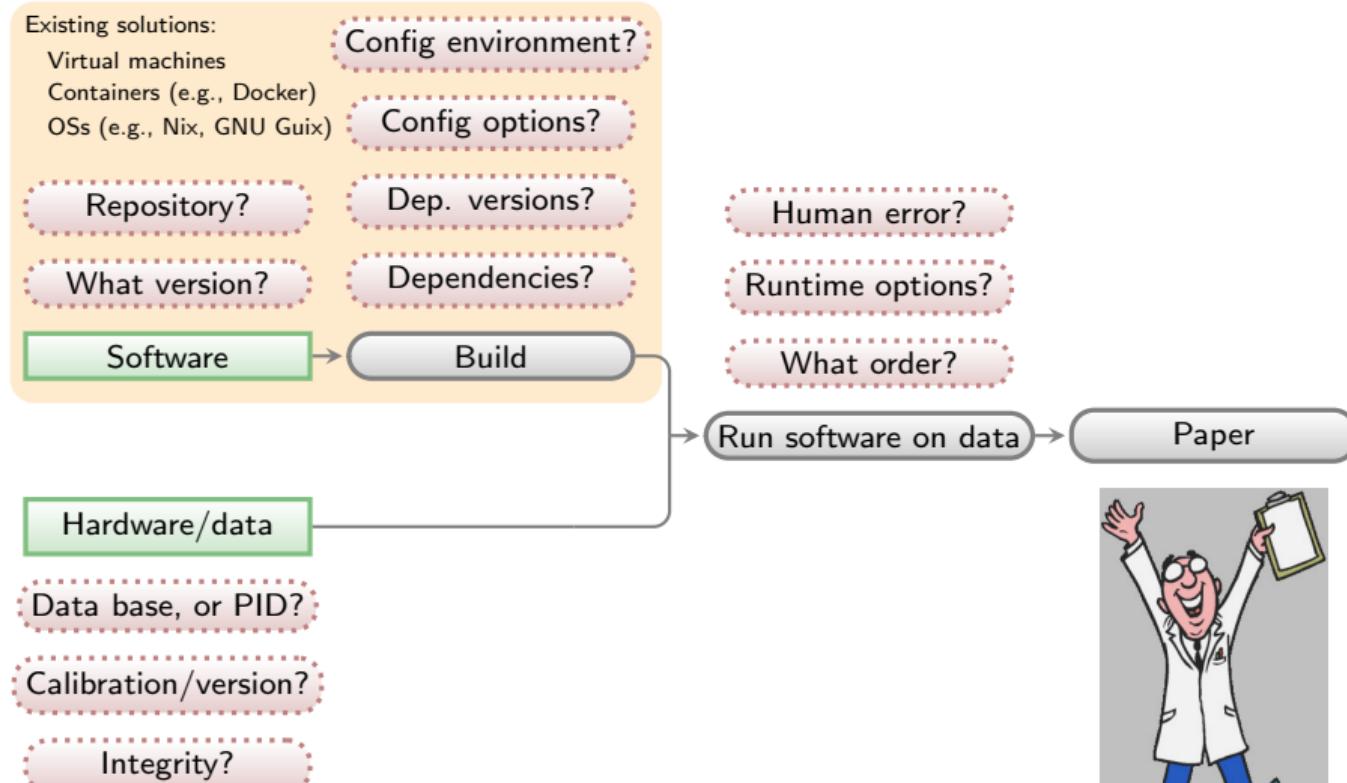
General outline of a project



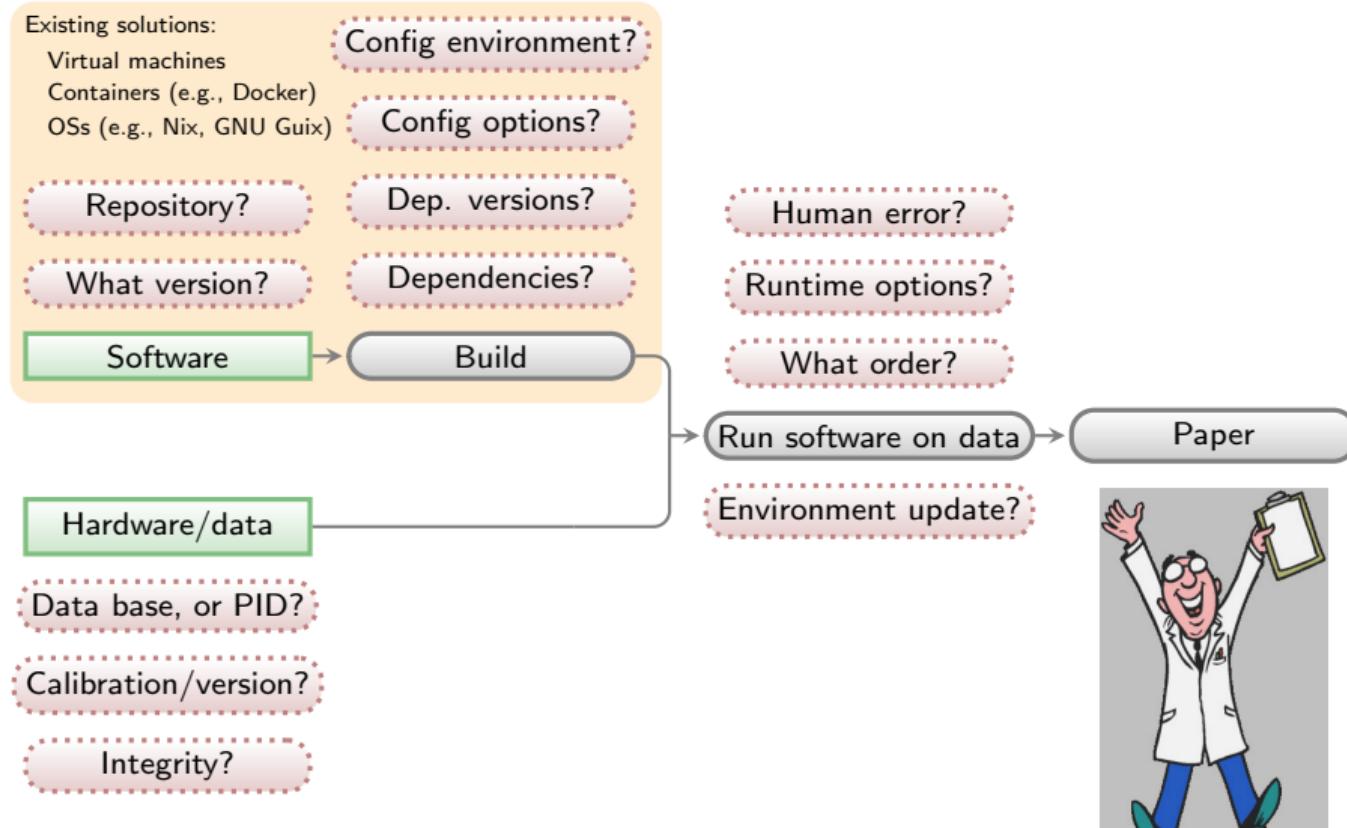
General outline of a project



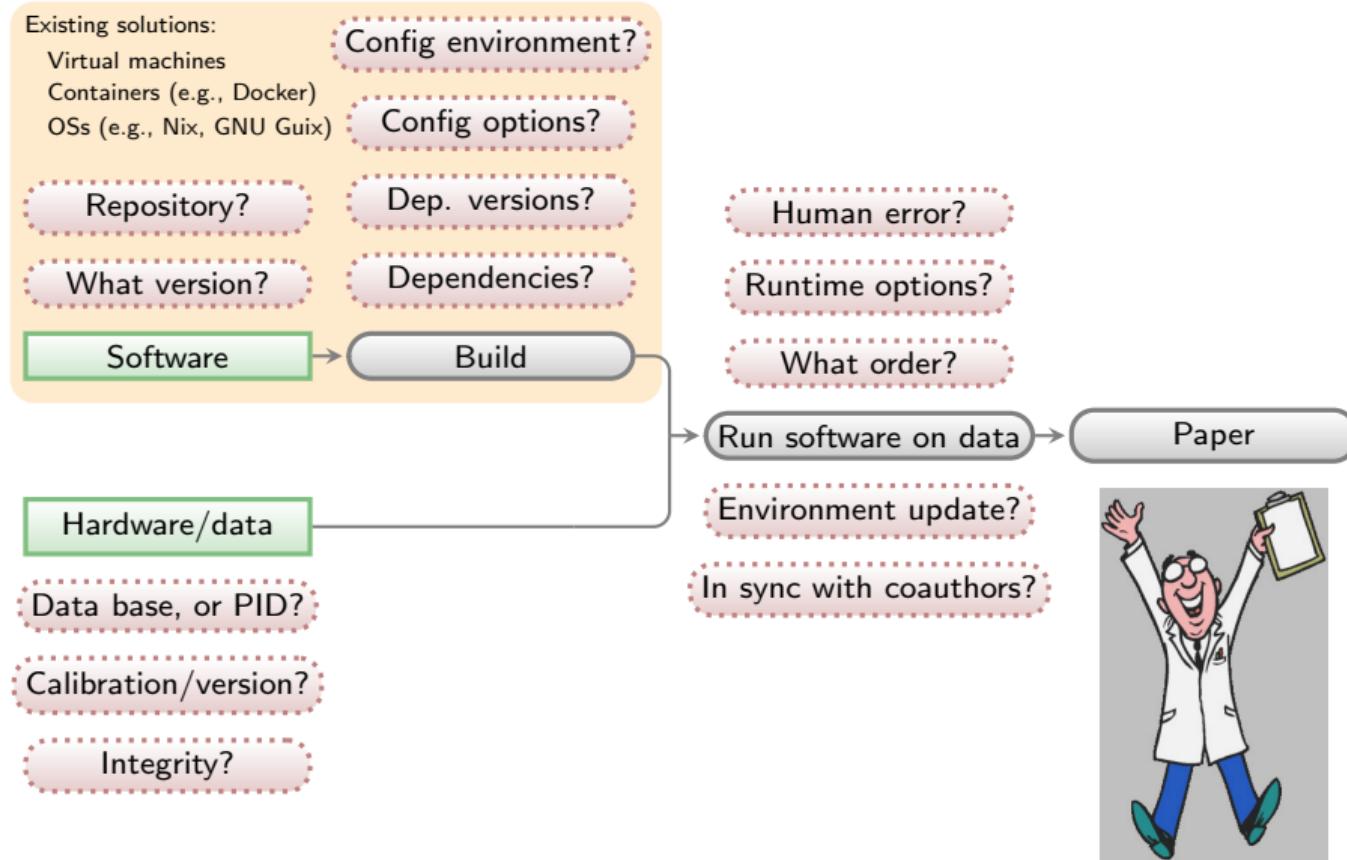
General outline of a project



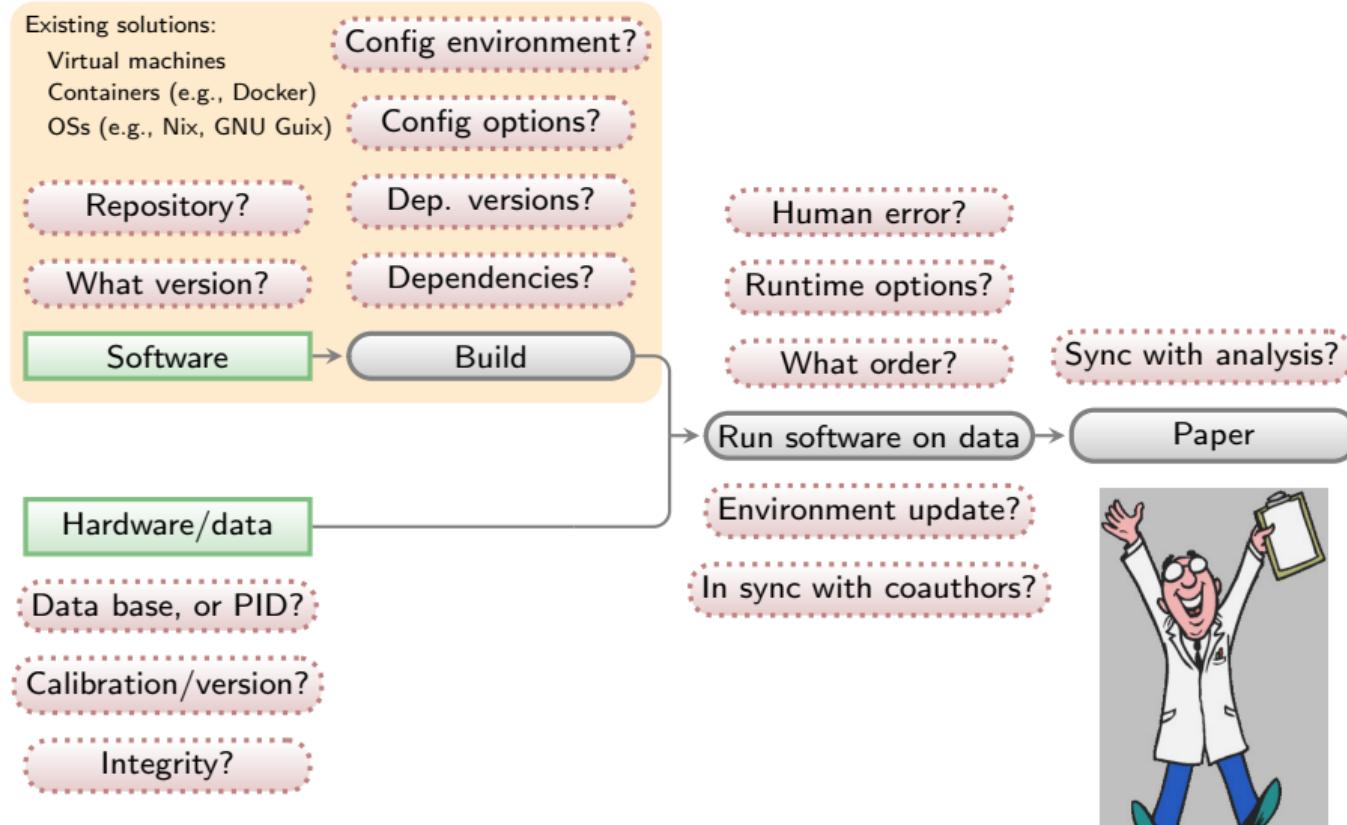
General outline of a project



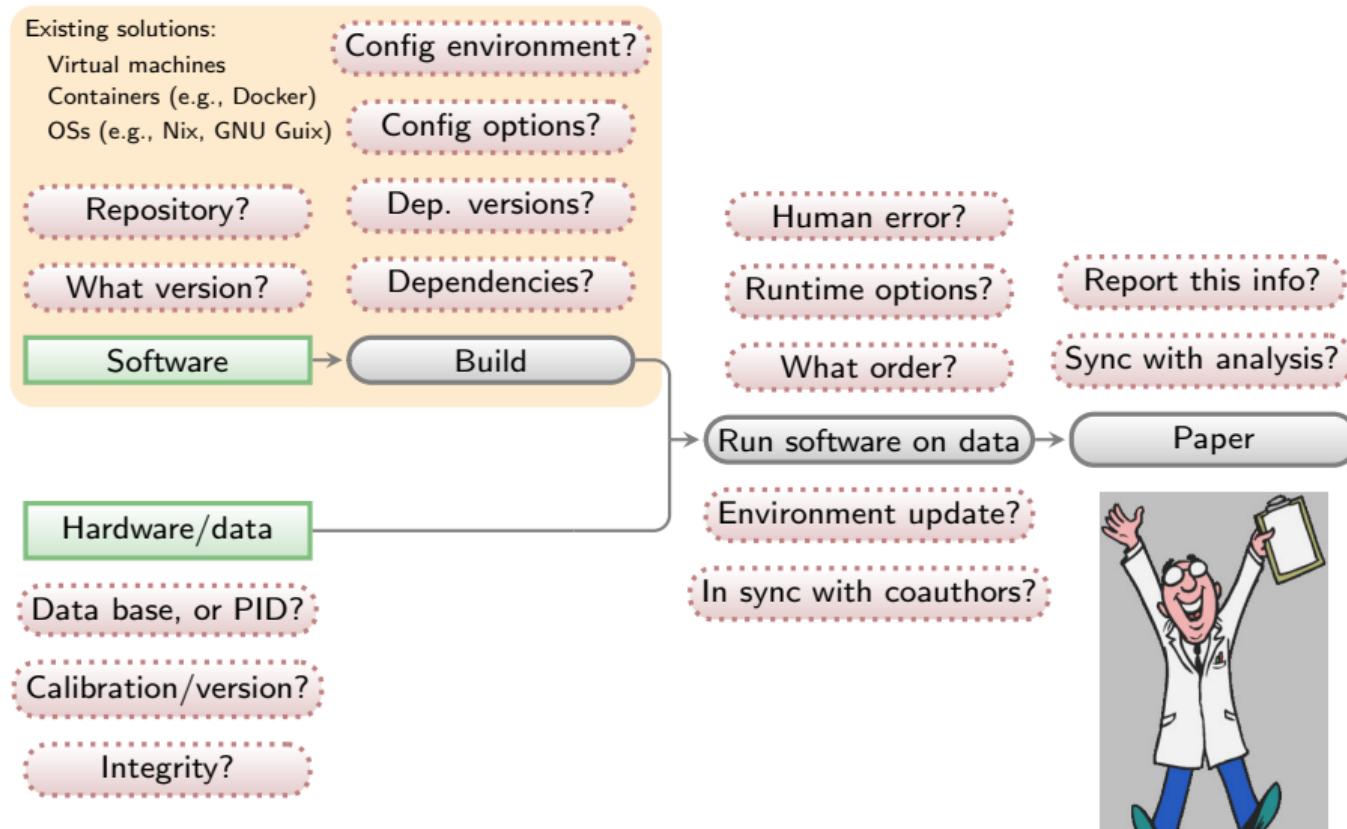
General outline of a project



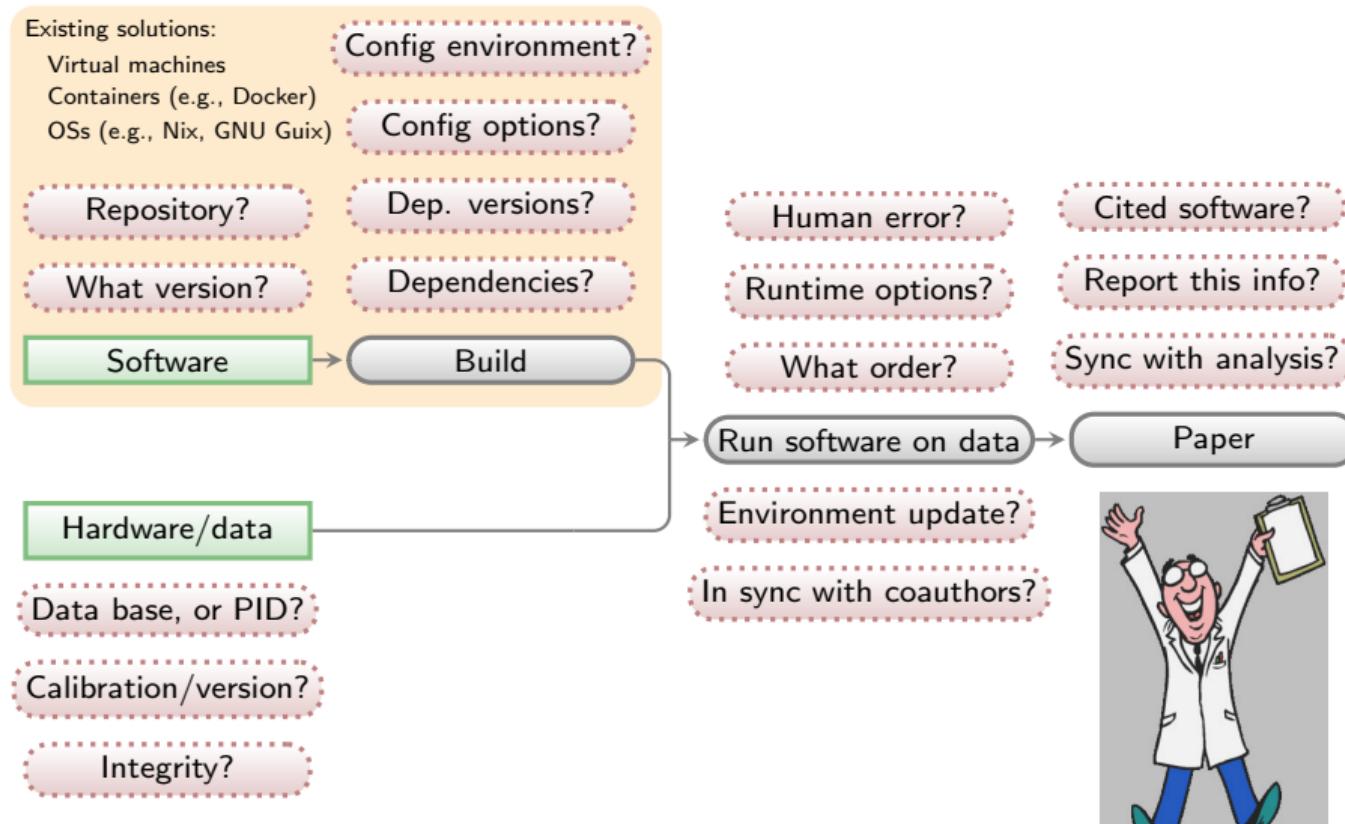
General outline of a project



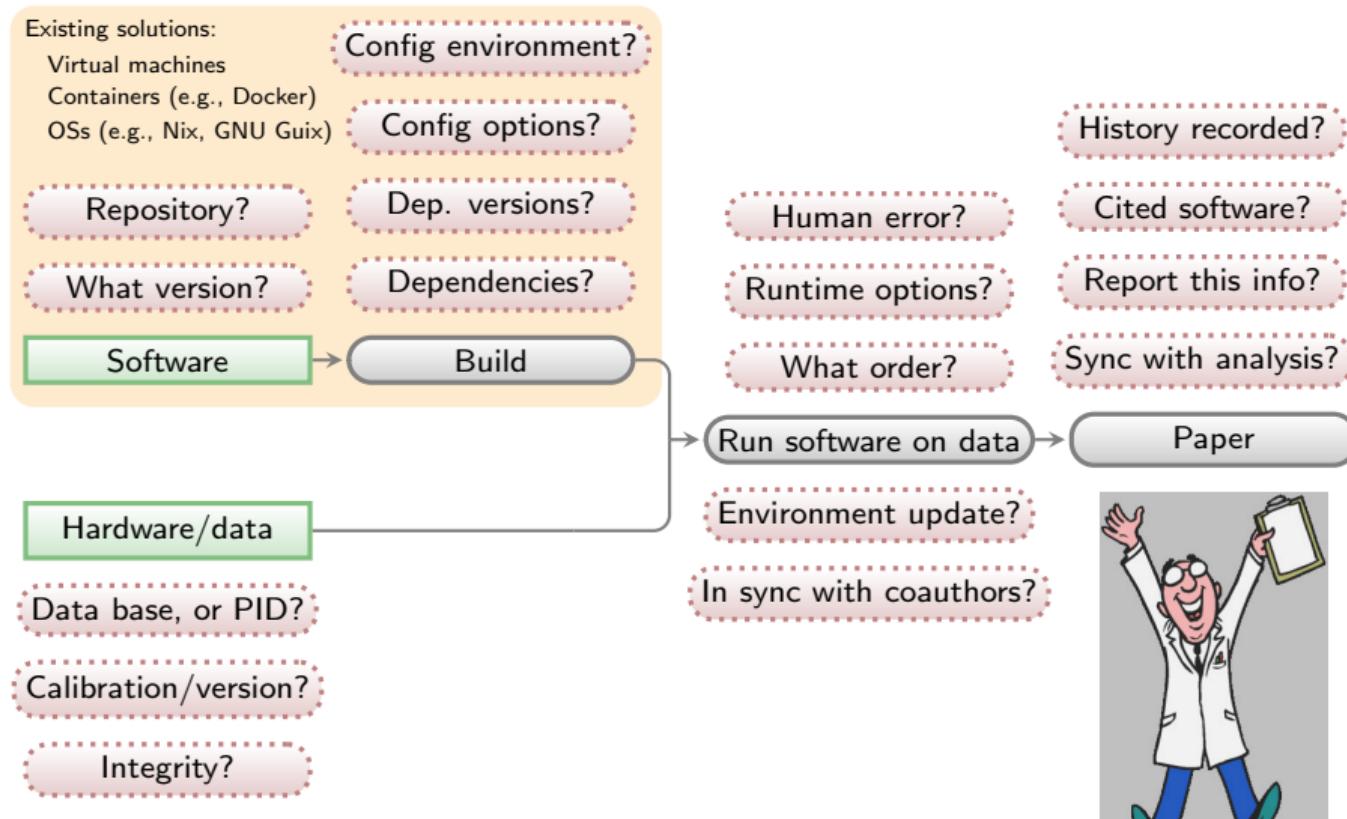
General outline of a project



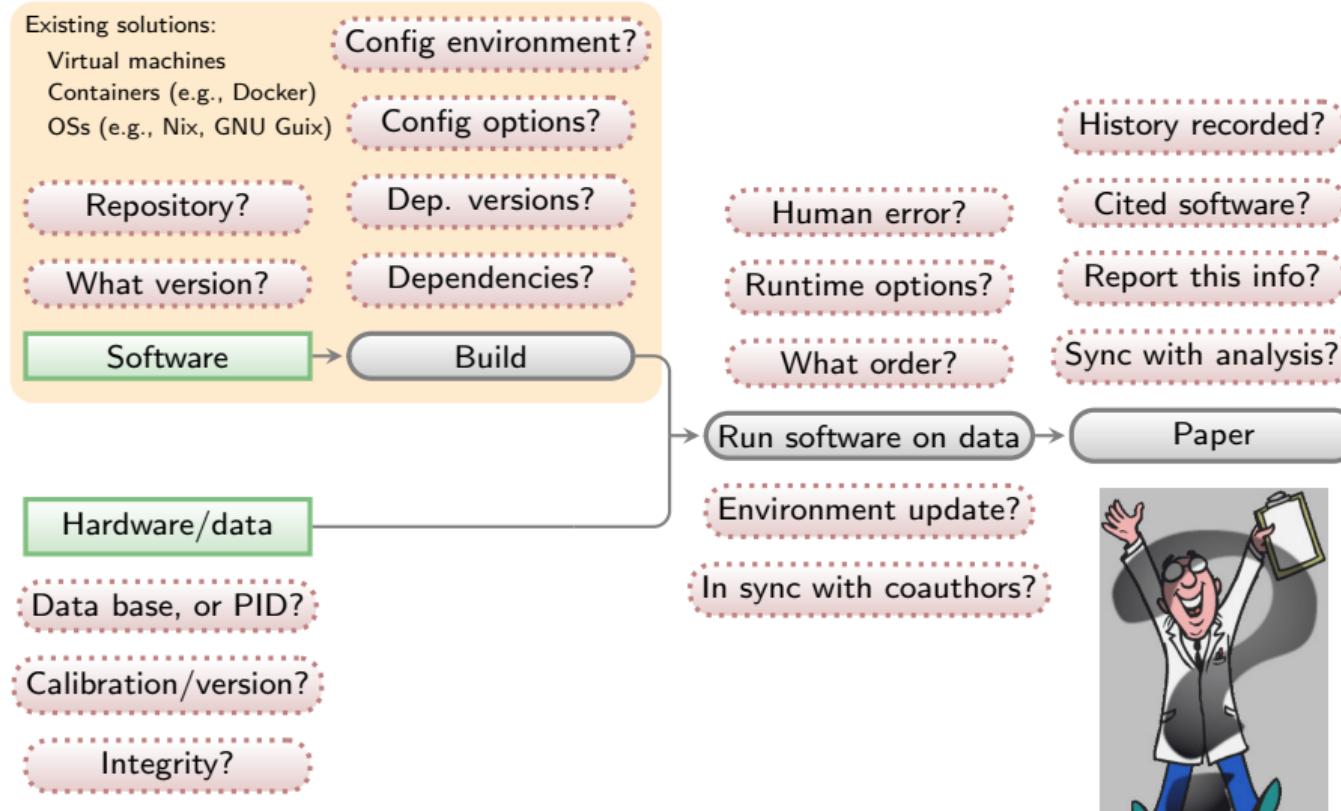
General outline of a project



General outline of a project



General outline of a project



Science is a tricky business

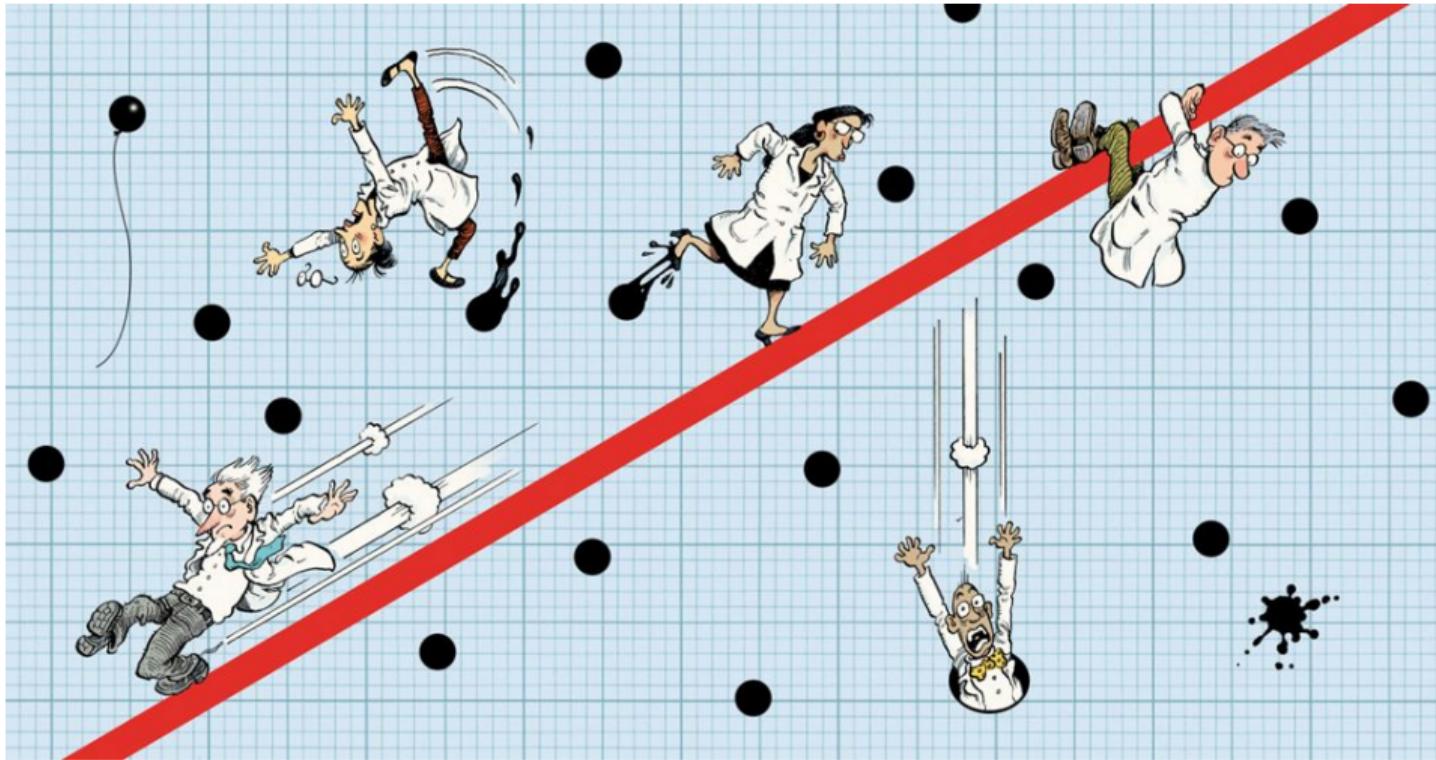


Image from nature.com ("Five ways to fix statistics", Nov 2017)

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.



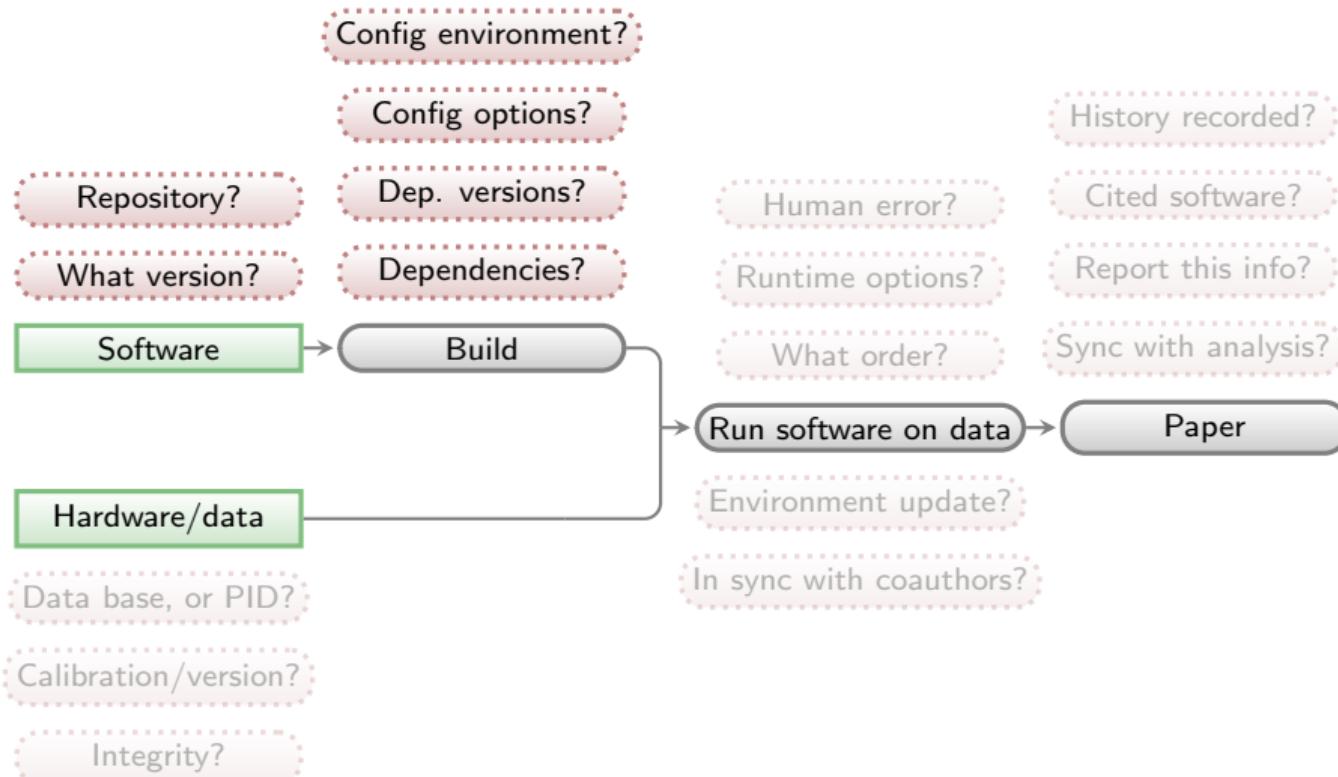
Necessity of (exactly) reproducible research

Don't forget that:

Science is defined by its METHOD, **not** its result.

- ▶ The software(s) used, configuration file(s), the order of steps taken, along with the input data are necessary for reproducibility.
- ▶ A solution is proposed here, which if adopted from the start, can greatly simplify a scientific research project and allow full/exact reproducibility once it is published.
- ▶ In the next slides, we'll review the template from the highest level (final research paper) to the lowest (setting up the research environment).

General outline of a project

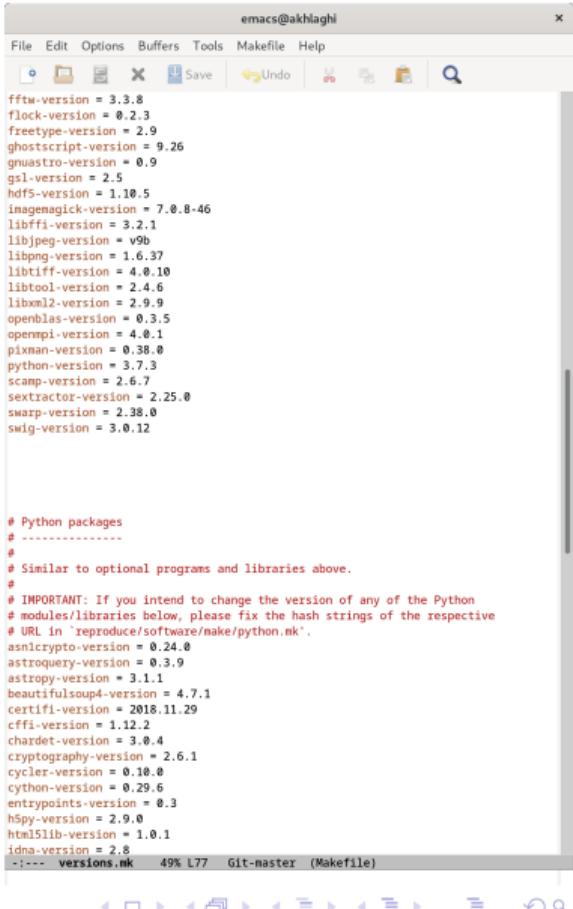


Predefined/exact software tools

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**.
- ▶ This template **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**).



The screenshot shows an Emacs window titled "emacs@akhlaghi". The buffer contains a list of software packages and their versions, starting with fftw, flock, freetype, ghostscript, gnuastro, gsl, hdf5, imagemagick, libffi, libjpeg, libpng, libtiff, libtool, libxml2, openblas, openmpi, pimlib, python, scamp, sexttractor, swarp, and swig. Below this, there is a section for Python packages, which includes asnicrypt, astroquery, astropy, beautifulsoup4, certifi, cffi, chardet, cryptography, cycler, cython, entrypoints, h5py, html5lib, idna, and requests. A note at the top of the Python section says: "# IMPORTANT: If you intend to change the version of any of the Python modules/libraries below, please fix the hash strings of the respective # URL in 'reproduce/software/make/python.mk'." The status bar at the bottom shows the file name "versions.mk", a progress bar at 49%, and the branch "Git-master (Makefile)".

```
fftw-version = 3.3.8
flock-version = 0.2.3
freetype-version = 2.9
ghostscript-version = 9.26
gnuastro-version = 0.9
gsl-version = 2.5
hdf5-version = 1.10.5
imagemagick-version = 7.0.8-46
libffi-version = 3.2.1
libjpeg-version = v9b
libpng-version = 1.6.37
libtiff-version = 4.0.10
libtool-version = 2.4.6
libxml2-version = 2.9.9
openblas-version = 0.3.5
openmpi-version = 4.0.1
pimlib-version = 0.38.0
python-version = 3.7.3
scamp-version = 2.6.7
sexttractor-version = 2.25.0
swarp-version = 2.38.0
swig-version = 3.0.12

# Python packages
# -----
#
# Similar to optional programs and libraries above.
#
# IMPORTANT: If you intend to change the version of any of the Python
# modules/libraries below, please fix the hash strings of the respective
# URL in 'reproduce/software/make/python.mk'.
asnicroptro-version = 0.24.0
astroquery-version = 0.3.9
astropy-version = 3.1.1
beautifulsoup4-version = 4.7.1
certifi-version = 2018.11.29
cffi-version = 1.12.2
chardet-version = 3.0.4
cryptography-version = 2.6.1
cycler-version = 0.10.0
cython-version = 0.29.6
entrypoints-version = 0.3
h5py-version = 2.9.0
html5lib-version = 1.0.1
idna-version = 2.8
requests-version = 2.22.0
urllib3-version = 1.25.7
versioneer-version = 1.0.0
wheel-version = 0.32.3
zipp-version = 3.0.4

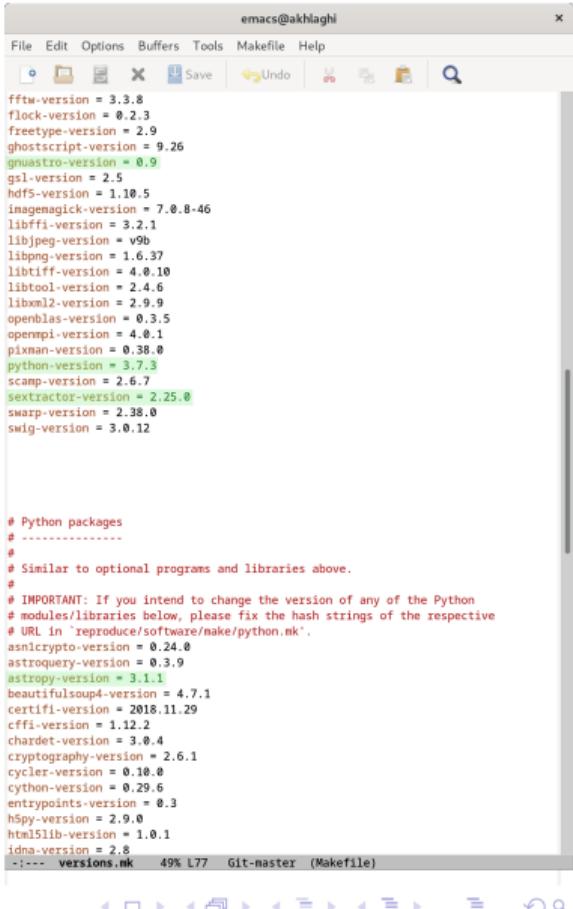
--- versions.mk 49% L77 Git-master (Makefile)
```

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**.
- ▶ This template **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**).



The screenshot shows an Emacs window titled "emacs@akhlaghi". The buffer contains a list of software packages and their versions, likely from a Makefile or configuration script. The list includes:

```
fftw-version = 3.3.8
flock-version = 0.2.3
freetype-version = 2.9
ghostscript-version = 9.26
gnaturo-version = 0.9
gsl-version = 2.5
hdf5-version = 1.10.5
imagemagick-version = 7.0.8-46
libffi-version = 3.2.1
libjpeg-version = v9b
libpng-version = 1.6.37
libtiff-version = 4.0.10
lrbtool-version = 2.4.6
linuxml2-version = 2.9.9
openblas-version = 0.3.5
openmpi-version = 4.0.1
pimman-version = 0.38.0
python-version = 3.7.3
scamp-version = 2.6.7
sextractor-version = 2.25.0
swarp-version = 2.38.0
swig-version = 3.0.12

# Python packages
# -----
# Similar to optional programs and libraries above.
#
# IMPORTANT: If you intend to change the version of any of the Python
# modules/libraries below, please fix the hash strings of the respective
# URL in 'reproduce/software/make/python.mk'.
asn1crypto-version = 0.24.0
astropy-version = 0.3.9
astropy-version = 3.1.1
beautifulsoup4-version = 4.7.1
certifi-version = 2018.11.29
cffi-version = 1.12.2
chardet-version = 3.0.4
cryptography-version = 2.6.1
cycler-version = 0.10.0
cython-version = 0.29.6
entrypoints-version = 0.3
h5py-version = 2.9.0
html5lib-version = 1.0.1
ida-version = 2.8
--- versions.mk 49% L77 Git-master (Makefile)
```

Controlled environment and build instructions

```
emacs@akhlaghi ~
```

File Edit Options Buffers Tools Makefile Help

include reproduce/software/config/installation/texlive.mk
include reproduce/software/config/installation/versions.mk

```
lockdir = ${_BODIR}/locks
tdir = ${_BODIR}/software/tarballs
ddir = ${_BODIR}/software/build-tmp
idir = ${_BODIR}/software/installed
ibdir = ${_BODIR}/software/installed/bin
ilidir = ${_BODIR}/software/installed/lib
dtxedir = ${_shell pad}/reproduce/software/bittree
itidir = ${_BODIR}/software/installed/version-info/tex
ictddir = ${_BODIR}/software/installed/version-info/cite
ipydir = ${_BODIR}/software/installed/version-info/python
ibidir = ${_BODIR}/software/installed/version-info/proglib

# Set the top-level software to build.
all: ${foreach x, ${top-level-programs}, ${!ibdir}${x} } \
      ${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
# 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 below 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
#    user's environment.
.ONESHELL:
.SHELLFLAGS := --noprofile --norc -ec
export CCACHE_DISABLE := 1
export PATH := ${ibdir}
export SHELL := ${ibdir}/bash
export CPPFLAGS := -I${ilidir}/include
export PKG_CONFIG_PATH := ${ibdir}/pkgconfig
export PKG_CONFIG_LIBDIR := ${ibdir}
export LD_RUN_PATH := ${ibdir}:${il64dir}
export LD_LIBRARY_PATH := ${ibdir}:${il64dir}
export LDFLAGS := ${rpath_command} -L${ibdir}

# 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
U:--- high-level.mk 4% 181 Git:master (Makefile)
```

Controlled environment and build instructions

```
emacs@akhlaghi ~
```

File Edit Options Buffers Tools Makefile Help

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 = $(BDIR)/software/installed
ibdir = $(BDIR)/software/installed/bin
ilidir = $(BDIR)/software/installed/lib
dtxedir = $(shell pwd)/reproduce/software/bibtex
itidir = $(BDIR)/software/installed/version-info/tex
ictdir = $(BDIR)/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
# 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 below 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
#    user's environment.
.ONESHELL:
.SHELLFLAGS := --noprofile --norc -e
export CACHE_DISABLE := 1
export PATH := $(ibdir)
export SHELL := $(ibdir)/bash
export CPPFLAGS := -I$(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 LDFLAGS := $(ipath_command) -L$(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
U:--- high-level.mk 4% L81 Git:master (Makefile)
```

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)
libjpeg.so.9 => /PROJECT/libjpeg.so.9 (0x00007f6745d12000)
libwcs.so.6 => /PROJECT/libwcs.so.6 (0x00007f6745ba8000)
libcfitsio.so.8 => /PROJECT/libcfitsio.so.8 (0x00007f674588b000)
libcurl.so.4 => /PROJECT/libcurl.so.4 (0x00007f6745811000)
libssl.so.1.1 => /PROJECT/libssl.so.1.1 (0x00007f6745777000)
libcrypto.so.1.1 => /PROJECT/libcrypto.so.1.1 (0x00007f6745491000)
libz.so.1 => /PROJECT/libz.so.1 (0x00007f6745474000)
libgsl.so.23 => /PROJECT/libgsl.so.23 (0x00007f67451e3000)
libgslcblas.so.0 => /PROJECT/libgslcblas.so.0 (0x00007f67451a1000)
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)
librt.so.1 => /usr/lib/librt.so.1 (0x00007f6744e36000)
linux-vdso.so.1 (0x00007ffffdcbf7000)
/lib64/ld-linux-x86-64.so.2 => /usr/lib64/ld-linux-x86-64.so.2
```

macOS

```
$ 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/liblzma.5.dylib (comp ver 8.0.0, cur ver 8.4.0)
/PROJECT/libjpeg.9.dylib (comp ver 12.0.0, cur ver 12.0.0)
/PROJECT/libwcs.6.2.dylib (comp ver 6.0.0, cur ver 6.2.0)
/PROJECT/libcfitsio.8.dylib (comp ver 8.0.0, cur ver 8.3.47)
/PROJECT/libcurl.4.dylib (comp ver 10.0.0, cur ver 10.0.0)
/PROJECT/libssl.1.1.dylib (comp ver 1.1.0, cur ver 1.1.0)
/PROJECT/libcrypto.1.1.dylib (comp ver 1.1.0, cur ver 1.1.0)
/PROJECT/libz.1.dylib (comp ver 1.0.0, cur ver 1.2.11)
/PROJECT/libgsl.23.dylib (comp ver 25.0.0, cur ver 25.0.0)
/PROJECT/libgslcblas.0.dylib (comp ver 1.0.0, cur ver 1.0.0)
/usr/lib/libSystem.B.dylib (comp ver 1.0.0, cur ver 1252.50.4)
```

Project libraries: High-level libraries built for each project.

GNU C Library: Currently not installed, will be available on GNU/Linux systems soon.

System/linker libraries: Very low-level, we do not need to control.

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)

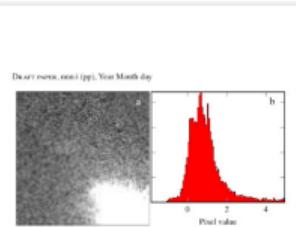


Figure 2: (a) An example image of the Wide-Field and Planetary Camera 2, on board the Hubble Space Telescope from 1993 to 2009. This is one of the sample images from the FITS standard webpage, kept as examples for this file format. (b) Histogram of pixel values in (a).

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. Notes, 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

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^[8].

This template also defines two `\bigrp` macros that allow you to mark text within your document as *new* and *older*. 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 `\highlightrgb{...}`, then the one that was marked *new* will become black (totally blends with the rest of the text) and the one marked `\textold{...}` will not be in the final PDF. You can thus use `\highlightrgb` to easily make copies of your research for existing co-authors (who are just interested in the new parts of 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>).

⁸ <http://mirrors.ctan.org/graphics/pgf/Downloads/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 Gnaastro, so it benefited from some grants. If you don't use Gnaastro in your final/customized project, please remove it from the paragraph below, only mentioning the reproducible paper template.

This research was partially done using GNU Astronomy Utilities (Gnuastro, gnuastro.org) and the reproducible paper template `gnuprime-2016-05-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 (12440412, 24253005), the European Research Council (ERC) advanced grant 330954-MILUSCAN, European Union's Horizon 2020 research and innovation programme under Marie Skłodowska-Curie grant agreement No 713467 to SUNRISE, and from the Spanish Ministry of Economy and Competitiveness (MINERCI) under grant number AYA2016-76219-P.

STAFF NAME, month (opp). Your Month day

6-24, rfontos 2016-06-24, ulem 2016-06-24, ulem 2016-06-24,
color 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

- kklaghi, M. and T. Ichikawa (Sept. 2015). *ApJS*, **220**, 1.
steppi Collaboration et al. (Oct. 2015). *A&A*, **558**, A33.
steppi Collaboration et al. (Sept. 2016). *A&A*, **596**, A1.
acca, R. et al. (Nov. 2017). *A&A*, **608**, A11.
sheth, S. et al. (Mar. 2018). *C&M*, **13**, 1.
mizutani, J. D. (2007). *C&E*, **9**, 90.
ishizuka, K. J. and M. Arribas (Mar. 2011). *C&M*, **13**, 9.
updegraff, T. E. (May 2007). *C&E*, **9**, 10.
van der Walt, S., et al. (Mar. 2011). *C&E*, **13**, 22.

Software citation automatically generated in paper (including Astropy)

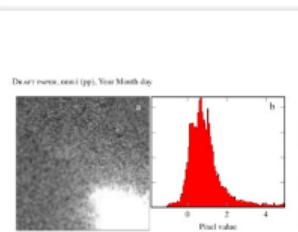


Figure 2: (a) An example image of the Wide-Field and Planetary Camera 2, on board the Hubble Space Telescope from 1993 to 2009. This is one of the sample images from the FITS standard webpage, kept as examples for this file format. (b) Histogram of pixel values in (a).

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 **GPIPlots** 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^[3].

This template also defines two `\bigrX` macros that allow you to mark text within your document as *new* and *sister*. 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 invisible with the rest of the text) and the one marked `\textcolor{teal}{text}` will not be in the final PDF. You can thus use `\highlightchanges` to easily make copies of your research for existing co-authors (who are just interested in the new parts of 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>).

⁴ See also my article "Non-Democratic Elements of Democracy," *Journal of Democracy*, 19, 3 (2008), 10–25.

<http://www.forsalebyowner.com/pennsylvania/philadelphia>

3. ACKNOWLEDGEMENTS

Please include the following two paragraphs in the Acknowledgments section of your paper. This reproducible paper template was developed in parallel with Guastro, so it benefited from the same grants. If you don't use Guastro 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 (<https://www.gnu.org/software/gnuastro/>) and the reproducible paper template v0-364-264-gf8c5d0. Work on Guazeto 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, 24230531), the European Research Council (ERC) advanced grant 339659-MUSCOS, European Union's Horizon 2020 research and innovation programme under Marie Skłodowska Curie grant Agreement No 721463 to the SUNDAIL ITN, and from the Spanish Ministry of Economy and Competitiveness (MINECO) under grant number AYA2016-76219-P.

WATSON, MARY (Mrs). Near Mouth dry

6-24, tfronts 2016-06-24, ulam 2016-06-24, ulam 2016-06-24,
color 2.12, xcolor 2.12, xkeyval 2.7a and xkeyval 2.7a. We are
very grateful to all their creators for freely providing this neces-
sary infrastructure. This research (and many others) would not
be possible without them.

References

- khalghi, M. and T. Ichikawa (Sept. 2015). *ApJL*, 229, 1.
Strropy Collaboration et al. (Oct. 2016). *A&A*, 558, A33.
Strropy Collaboration et al. (Sept. 2018). *AJ*, 156, 123.
taou, R. et al. (Nov. 2017). *A&A*, 608, A11.
shii, S. et al. (Mar. 2018). *C&E*, 13, 11.
auer, J. D. (2007). *C&E*, 9, 99.
ikuta, K. J. and M. Arratia (Mar. 2011). *C&E*, 13, 9.
upham, T. E. (May 2007). *C&E*, 9, 10.
n der Wahl, S. et al. (Mar. 2011). *C&E*, 13, 22.

YOUR NAME OR A.:

Software citation automatically generated in paper (only GNU Astronomy Utilities)

Appendix A: Software acknowledgement

The reproducible paper template that is customized for this project automatically installs all the necessary software. Directly listing all the high-level software and their versions is done with two primary motives: 1) software citation and acknowledgement of the hard work (or part of different software projects) that this project used; 2) reproducibility (future releases).

This research paper does not list every file from software programs and libraries: Bzip2 1.0.6, CHTFSIO 3.47, CMaker 3.14.2, cURL 7.63.0, Discreet Rock 0.2.3, File 5.36, Git 2.22.0, GNU Astronomy Utilities 0.9.70-tiffs (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 C Compiler 9.1.0-Difflib 3.7, GNU Emacs 26.1.109-1, GNU Grep 3.3, GNU Guile 1.10, GNU Integer Set Library 0.18, GNU Liblzma 2.4.6, GNU M4 1.4.18, GNU Make 4.2.0, GNU Multiple Precision Arithmetic Library 6.1.2, GNU Multiple-Precision Complex Library, GNU Multiple-Precision Floating-Point Reliable Library 4.0.2, GNU TLIBRARY 8.1, GNU Readline 8.0, GNU Scientific Library 2.5, GNU Sed 4.7.1, GMP 6.1.2, Jansson 3.1, L10N 2.13, GNU Witch 2.21, GPL Ghostscript 9.26, Libbed 0.9.1, Libgit2 0.28.2, Liblouis 0.96, Libtiff4 4.0.10, Lzip 1.20, Metastore (forked) 1.1.2.23-fa9f70b, OpenSSL 1.1.1a, PatchELF 0.9, pkg-config 0.29.2, Unzip 6.0, WCSLIB 6.2, XZ Utils 5.2.4, Zip 3.0 and Zlib 1.2.11. The source of LaTeX was converted to PDF using the PDF using the following packages: biblatex 3.12, caption 2018-10-05, charter 2016-06-24, counter 2016-06-24, csquotes 5.2d, datenume 2.6b, ec 1.0, environ 0.3, eso-pic 2.5f, extsizes 1.4a, fancyhdr 3.10, filenamem 3.05, fontaxes 1.0aI, fontenc 5.5b, fy 2.1d, helvetin 2016-06-24, hyperref 4.4.1, logreq 1.0, newsys 1.0, pgf 1.3.2, pgfplots 1.16, preprint 2011, setkeys 8.7a, subwidth 2.0, tabularx 2.02, tcolorbox 2.92, tcolorbox 2.50.1, times 2016-06-24, tfsense 2.10.2, vimtex 2018-01-11, tufors 2016-06-24, xcolor 2016-06-24, xcolor 2.12 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.

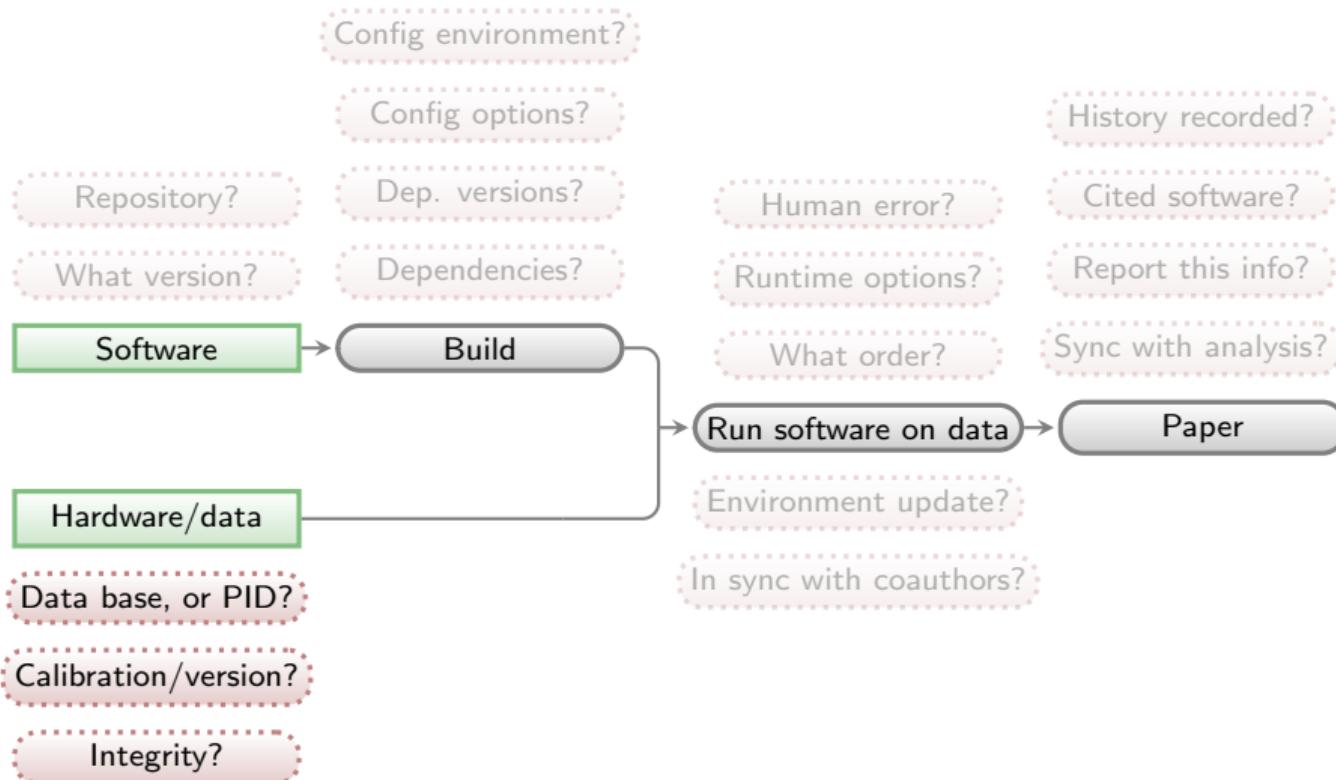
Software citation automatically generated in paper (only GNU Astronomy Utilities)

Appendix A: Software acknowledgement

The reproducible paper template that is customized for this project automatically installs all the necessary software. Directly listing all the high-level software and their versions is done with two primary motives: 1) software citation and acknowledgement of the hard work (or part of different software projects) that this project utilized. 2) reproducibility (for future readers).

This paper was built using the following software programs and libraries: Bzip2 1.0.6, CHTFSO 3.47, CMaker 3.14.2, cURL 7.63.0, Discreet Rock 0.2.3, File 5.36, Git 2.22.0, GNU Astronomy Utilities 0.9.70-tiffs (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 4.7.0, GNU GDB 8.3, GNU Gcov 3.3, GNU GDBM 1.10, GNU Integer Set Library 0.18, GNU Linpack 2.4.6, GNU M4 1.4.18, GNU Make 4.2.0, GNU Multiple Precision Arithmetic Library 6.1.2, GNU Multiple-Precision Complex Library, GNU Multiple-Precision Floating-Point Reliable Library 4.0.2, GNU MPFR 4.0.4, GNU Readline 8.0, GNU Scientific Library 2.5, GNU Sed 4.5, GNU Tar 1.30, GMP 6.1.0, GMPFR 2.3.1, GNU Witch 2.21, GPL Ghostscript 9.26, Libbed 0.9.1, Libgiz2 0.28.2, libjpeg v9b, Libtiff 4.0.10, Lzip 1.20, Metastore (forked) 1.1.2.23-fa9f70b, OpenSSL 1.1.1a, PatchELF 0.9, pkg-config 0.29.2, Umlib 6.0, WCSLIB 6.2, XZ Utils 5.2.4, Zip 3.0 and Zlib 1.2.11. The TeX source of the paper was compiled to generate the PDF using the following tools: bib2bib 1.12, biblatex 3.12, caption 2018-10-05, chapter 2016-06-24, color 2016-06-24, csquotes 5.2d, datatime 2.6b, ec 1.0, environ 0.3, etoolbox 2.5f, extsizes 1.4a, fancyhdr 3.10, filenamemangle 3.05, fontaxes 1.0a, fontenc 5.5b, fy 2.1d, helvetea 2016-06-24, hyperref 4.4.1, logreq 1.0, news 1.3, pgf 3.1.2, pgfplots 1.16, preprint 2011, setkeys 8.7a, setspace 2.0.1, siunitx 2.3.1, smartref 2019-02-28, times 2016-06-24, tesseract 2.10.2, tiffformat 1.1, tuftebook 2016-06-24, tulem 2016-06-24, xcolor 2.12 and xkeyval 2.7e. 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.

General outline of a project



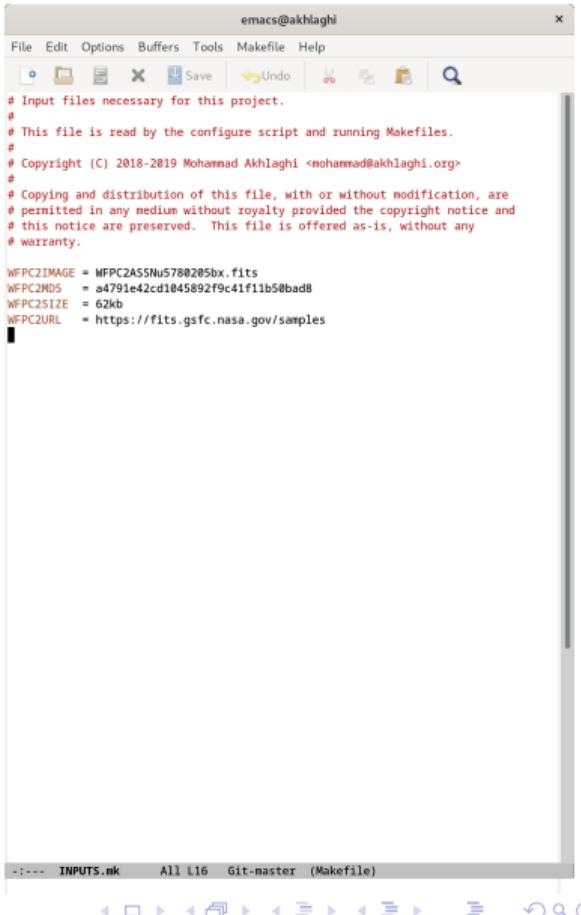
Input data source and integrity is documented and checked

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).



The screenshot shows a terminal window titled "emacs@akhlaghi". The window contains configuration settings for a project, specifically for WFCPC2. The settings include:

```
# 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.

WFCPC2IMAGE = WFCPC2ASSNu5780205bx.fits
WFCPC2MDS = a4791e42cd1045892f9c41f11b50bad8
WFCPC2SIZE = 62kb
WFCPC2URL = https://fits.gsfc.nasa.gov/samples
```

The status bar at the bottom of the terminal window shows the current directory as "INPUTS.mk All L16 Git-master (Makefile)".

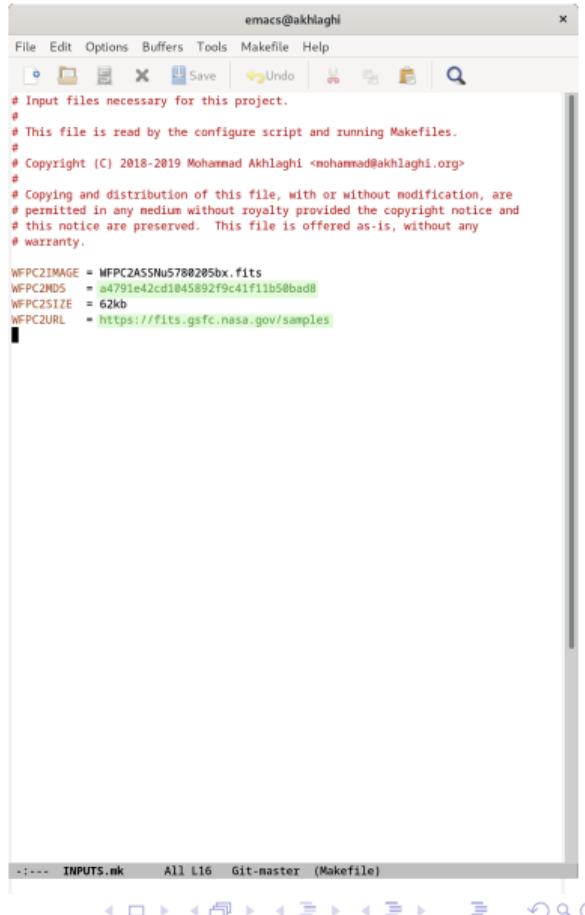
Input data source and integrity is documented and checked

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).

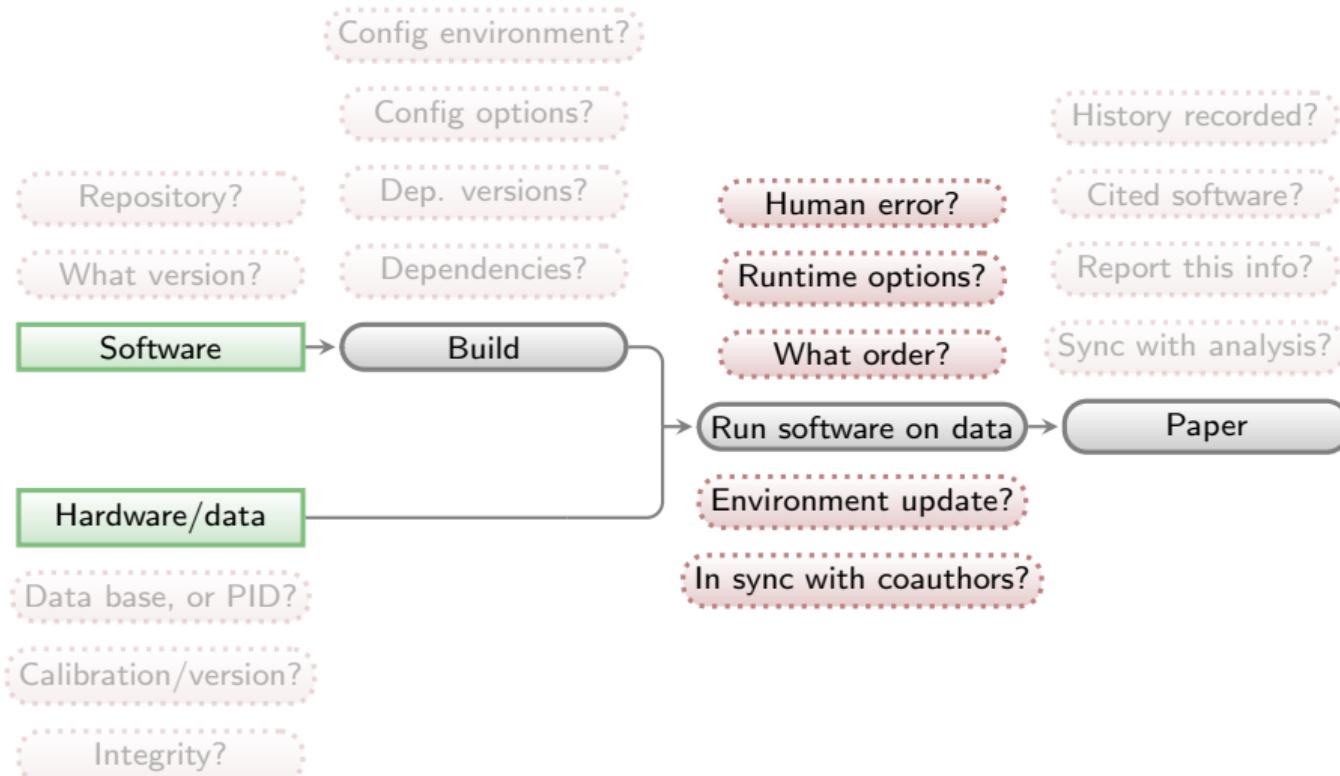


```
emacs@akhlaghi
File Edit Options Buffers Tools Makefile Help
Save Undo
# Input files necessary for this project.
#
# This file is read by the configure script and running Makefiles.
#
# Copyright (C) 2018-2019 Mohammad Akhlaghi <mohammad@akhlaghi.org>
#
# Copying and distribution of this file, with 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.

WFPC2IMAGE = WFPC2ASSNu5780205bx.fits
WFPC2MDS = a4791e42cd1045892f9c41f11b5@bad8
WFPC2SIZE = 62kb
WFPC2URL = https://fits.gsfc.nasa.gov/samples
```

-:--- INPUTS.mk All L16 Git-master (Makefile)

General outline of a project



Reproducible science: Template 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](#)).



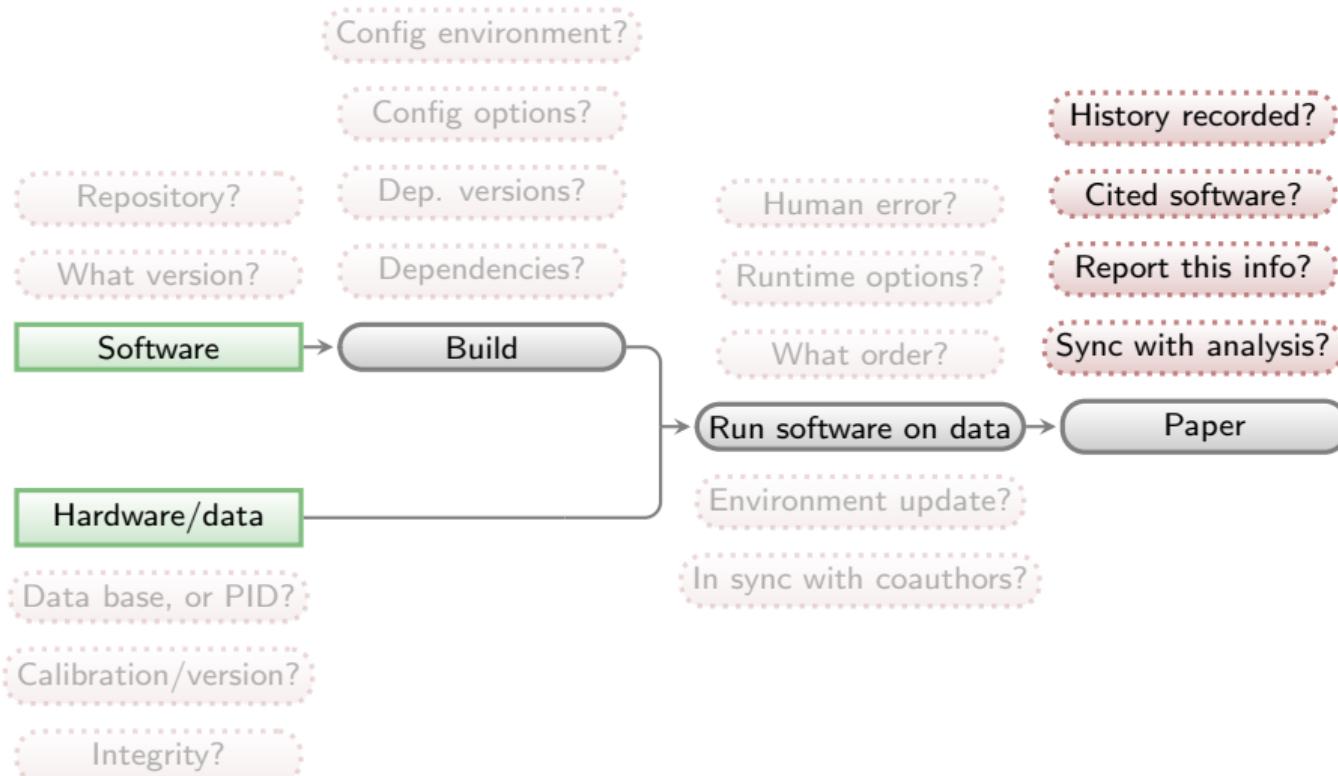
The screenshot shows a terminal window titled "emacs@akhlaghi" with a Makefile for "NoiseChisel". The file contains several rules for generating various types of catalog files (e.g., udf, rawcat, nccat) from raw data. The code uses shell-like syntax with variables like \$ncfdir, \$alffilter, and \$f. It includes comments explaining the purpose of each step, such as generating a NoiseChisel segmentation map and running it on raw aperture catalogs. The terminal also shows the bottom part of the Makefile, which includes a rule for generating LaTeX files and a note about zero-point calculations.

```
# Run NoiseChisel
#
# NoiseChisel's output is needed for several things down the line: Its
# Sky and Sky standard deviation outputs will be used in the several
# runs of MakeCatalog. Its detections are also going to be used to
# create a NoiseChisel segmentation map. We also need the Sky values
# for the raw aperture catalogs, so we'll also run NoiseChisel on the
# images with a gradient..
alff = $(acsf) $(wfcf)
ncfdir = $(fdir)/noisechisel
$(ncfdir): | $(fdir); mkdir $@
noisechisel: | $(foreach f, $(alffilters), $(ncfdir)/udf_$(f).fits)
              $(foreach f, $(xdfwfc3lrf), $(ncfdir)/xdf_$(f).fits)
              $(foreach f, $(xdfwfc3irf), $(ncfdir)/grd_$(f).fits)
$noisechisel: | $(ncfdir):%: $!depth)% .gmuastro/astnoisechisel.conf \
               | $(ncfdir)
if [ $* == "udf f225w.fits" ] || [ $* == "udf f275w.fits" ]
|| [ $* == "udf f336w.fits" ]; then extraopt="--qthresh=0.4";
else extraopt= ""; fi;
astnoisechisel $extraopt --detquant=0.9 --segquant=0.9 < -o$@

# Pure NoiseChisel catalog on each filter/depth
#
# Catalog of all of NoiseChisel's clumps on each filter. Do not
# confuse this with the aperture photometry catalog that is also
# generated by MakeCatalog. For the same filter, both catalogs use the
# same image, sky and sky standard deviation images, but the labeled
# images differ. Here NoiseChisel's labeling is used, there an
# aperture labeled image is created separately.
nccatdir = $(catdir)/noisechisel
nccatdir = $(catdir)/noisechisel/raw
nccatcat = $(foreach f, $(alffilters), $(nccatdir)/udf_$(f)_c.txt)
           $(foreach f, $(xdwf3lrf), $(nccatdir)/xdf_$(f)_c.txt)
$inccatdir: | $(catdir); mkdir $@
$inccatcat: | $(nccatdir); mkdir $@
$inccatcat: | $(nccatdir)% c.txt: $(ncfdir)% .fits
$inccatcat: | $(nccatdir)% c.txt: $(nccatdir)/ .gmuastro/astmkcatalog.conf | $(nccatdir)
           .zps=$$reproduce/src/zeropoints.sh $(word 2,$(subst _, ,*)) ;
astmkcatalog $@ --zeropoint=$$zps -o$(@D)/$@

# Write values for LaTeX
#
#----- raw-cats.mk 23% L46 Git-master (GNUmakefile)
```

General outline of a project



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](https://arxiv.org/abs/1505.01664)).

```
\begin{equation}
\label{tSNeq}
\mathrm{S/N}_T = \frac{\sqrt{N} \sigma_s}{\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 $\mathrm{S/N}_T$ 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 $\mathrm{S/N}$ of false detections in real, reduced/co-added images. A comparison of scales on the $\mathrm{S/N}$ 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 $\mathrm{S/N}$ of $\$oneLargedettfmax$, and \$fourdettfmax$. \square$

smaller than `--detsminarea` 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,

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

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 $\mathrm{S/N}$ of false detections in real, reduced/co-added images. A comparison of scales on the $\mathrm{S/N}$ 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 $\mathrm{S/N}$ of 1.89, 2.37, and 4.77.

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

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](https://arxiv.org/abs/1505.01664)).

```
\begin{equation}
\label{tSNeq}
\mathrm{S/N}_T = \frac{\sqrt{N} \sigma_s}{\sqrt{N(F - S_a)}}
= \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 $\mathrm{S/N}_T$ 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 $\mathrm{S/N}$ of false detections in real, reduced/co-added images. A comparison of scales on the $\mathrm{S/N}$ 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 $\mathrm{S/N}$ of $\$onelargedettfmax$, $\\sensitivitycdettfmax$, and $\\fourdettfmax$. \square$

smaller than `--detsminarea` 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,

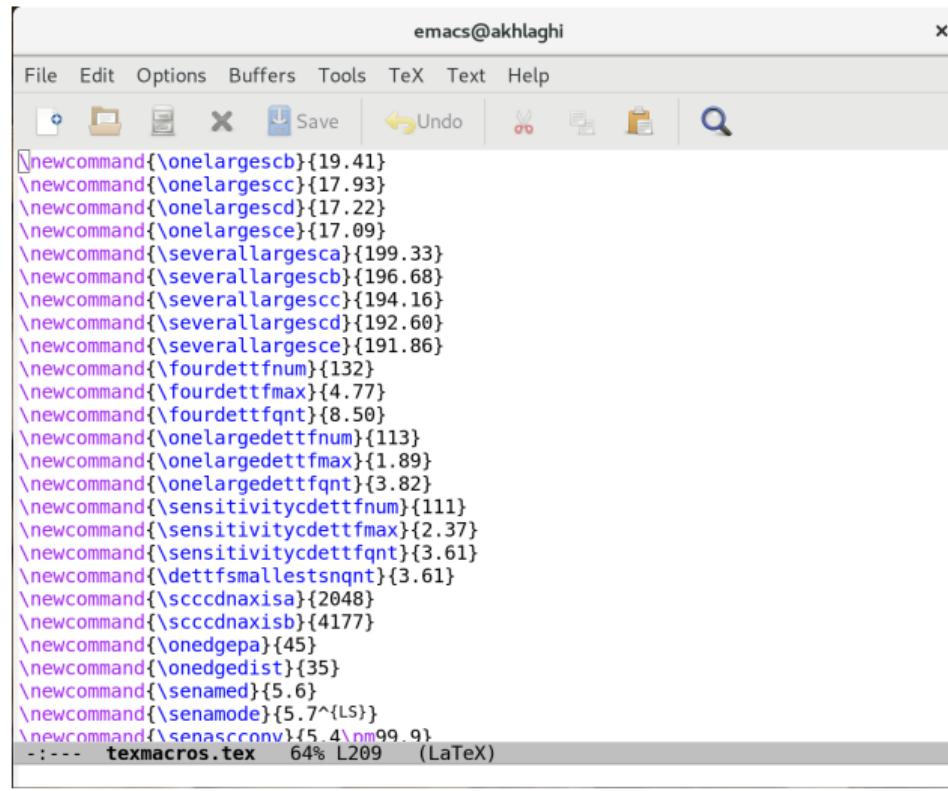
$$\mathrm{S/N}_T = \frac{NF - NS_a}{\sqrt{NF + N\sigma_s^2}} = \frac{\sqrt{N}(F - S_a)}{\sqrt{F + \sigma_s^2}}. \quad (3)$$

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 $\mathrm{S/N}$ of false detections in real, reduced/co-added images. A comparison of scales on the $\mathrm{S/N}$ 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 $\mathrm{S/N}$ 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**.

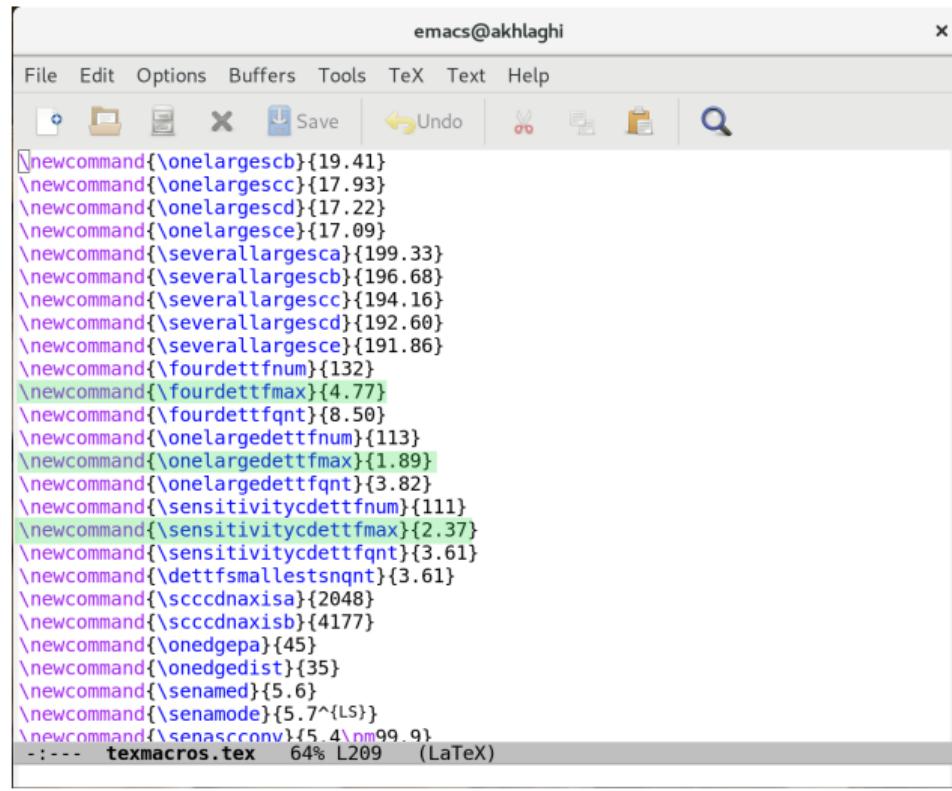


The screenshot shows an Emacs window titled "emacs@akhlaghi". The menu bar includes File, Edit, Options, Buffers, Tools, TeX, Text, and Help. The toolbar contains icons for Save, Undo, and other functions. The main buffer displays a series of LaTeX command definitions, all starting with "\newcommand". The commands are listed vertically, such as "\newcommand{\onelargescb}{19.41}" and "\newcommand{\severallargesca}{199.33}". At the bottom of the buffer, the file name "- :--- texmacros.tex" is shown along with the page number "64% L209" and the file type "(LaTeX)".

```
\newcommand{\onelargescb}{19.41}
\newcommand{\onelargescc}{17.93}
\newcommand{\onelargescd}{17.22}
\newcommand{\onelargesce}{17.09}
\newcommand{\severallargesca}{199.33}
\newcommand{\severallargesch}{196.68}
\newcommand{\severallargesc}{194.16}
\newcommand{\severallargescd}{192.60}
\newcommand{\severallargesce}{191.86}
\newcommand{\fourdettfnum}{132}
\newcommand{\fourdettfmax}{4.77}
\newcommand{\fourdettfqnt}{8.50}
\newcommand{\onelargedettfnum}{113}
\newcommand{\onelargedettfmax}{1.89}
\newcommand{\onelargedettfqnt}{3.82}
\newcommand{\sensitivitycdettfnum}{111}
\newcommand{\sensitivitycdettfmax}{2.37}
\newcommand{\sensitivitycdettfqnt}{3.61}
\newcommand{\dettfsmallestsnqnt}{3.61}
\newcommand{\scccdnavisa}{2048}
\newcommand{\scccdnavisb}{4177}
\newcommand{\onedgepa}{45}
\newcommand{\onedgedist}{35}
\newcommand{\senamed}{5.6}
\newcommand{\senemode}{5.7^{(L5)}}
\newcommand{\senascconv}{5.4 \text{nm}^{99.9}}
```

Analysis step results/values concatenated into a single file.

All \LaTeX macros come from a **single file**.



The screenshot shows an Emacs window titled "emacs@akhlaghi". The menu bar includes File, Edit, Options, Buffers, Tools, TeX, Text, and Help. Below the menu is a toolbar with icons for Save, Undo, and others. The main buffer contains a list of LaTeX commands, each consisting of a command name followed by a brace-enclosed argument. The commands are color-coded: some are blue, some are green, and some are purple. The file name "texmacros.tex" is visible at the bottom of the buffer area. The status bar at the bottom right shows "texmacros.tex" and "64% L209 (LaTeX)".

```
\newcommand{\onelargescb}{19.41}
\newcommand{\onelargescc}{17.93}
\newcommand{\onelargescd}{17.22}
\newcommand{\onelargesce}{17.09}
\newcommand{\severallargesca}{199.33}
\newcommand{\severallargesch}{196.68}
\newcommand{\severallargesc}{194.16}
\newcommand{\severallargescd}{192.60}
\newcommand{\severallargesce}{191.86}
\newcommand{\fourdettfnum}{132}
\newcommand{\fourdettfmax}{4.77}
\newcommand{\fourdettfqnt}{8.50}
\newcommand{\onelargedettfnum}{113}
\newcommand{\onelargedettfmax}{1.89}
\newcommand{\onelargedettfqnt}{3.82}
\newcommand{\sensitivitycdettfnum}{111}
\newcommand{\sensitivitycdettfmax}{2.37}
\newcommand{\sensitivitycdettfqnt}{3.61}
\newcommand{\dettfsmallestsnqnt}{3.61}
\newcommand{\scccdnavisa}{2048}
\newcommand{\scccdnavisb}{4177}
\newcommand{\onedgepa}{45}
\newcommand{\onedgedist}{35}
\newcommand{\senamed}{5.6}
\newcommand{\senamode}{5.7^{LS}}
\newcommand{\senascconv}{5.4\text{nm}^{99.9}}
- :--- texmacros.tex 64% L209 (LaTeX)
```

Analysis results stored as \LaTeX macros

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

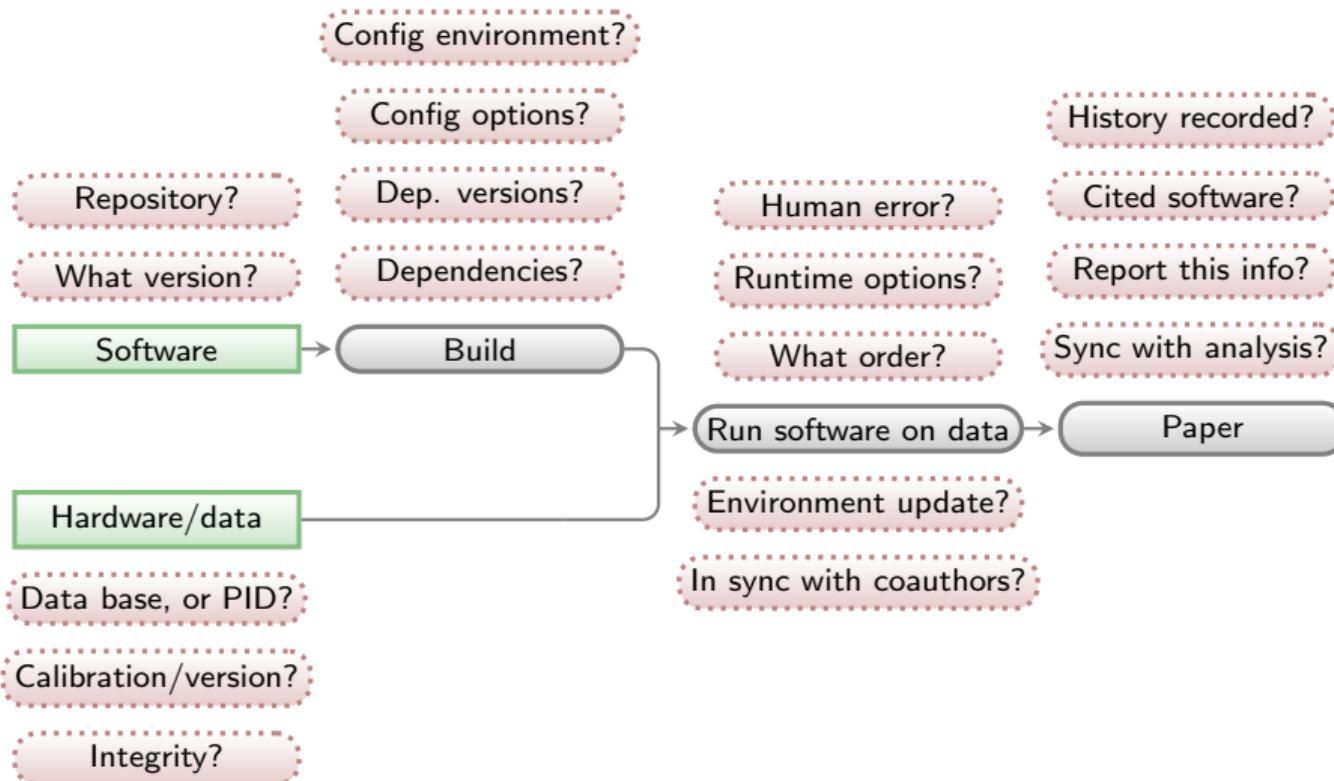
```
# Numbers for dettf.tex:  
sqnt=999999  
function dettfhist  
{  
    # Set the file name.  
    if [ $2 == 4 ]; then          obase=four;  
    elif [ $2 = sensitivity3 ]; then    obase=sensitivityc;  
    else                          obase=$2;  
    fi  
    if [ $2 == onelarge ]; then ind=_7; 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
```

Analysis results stored as \LaTeX macros

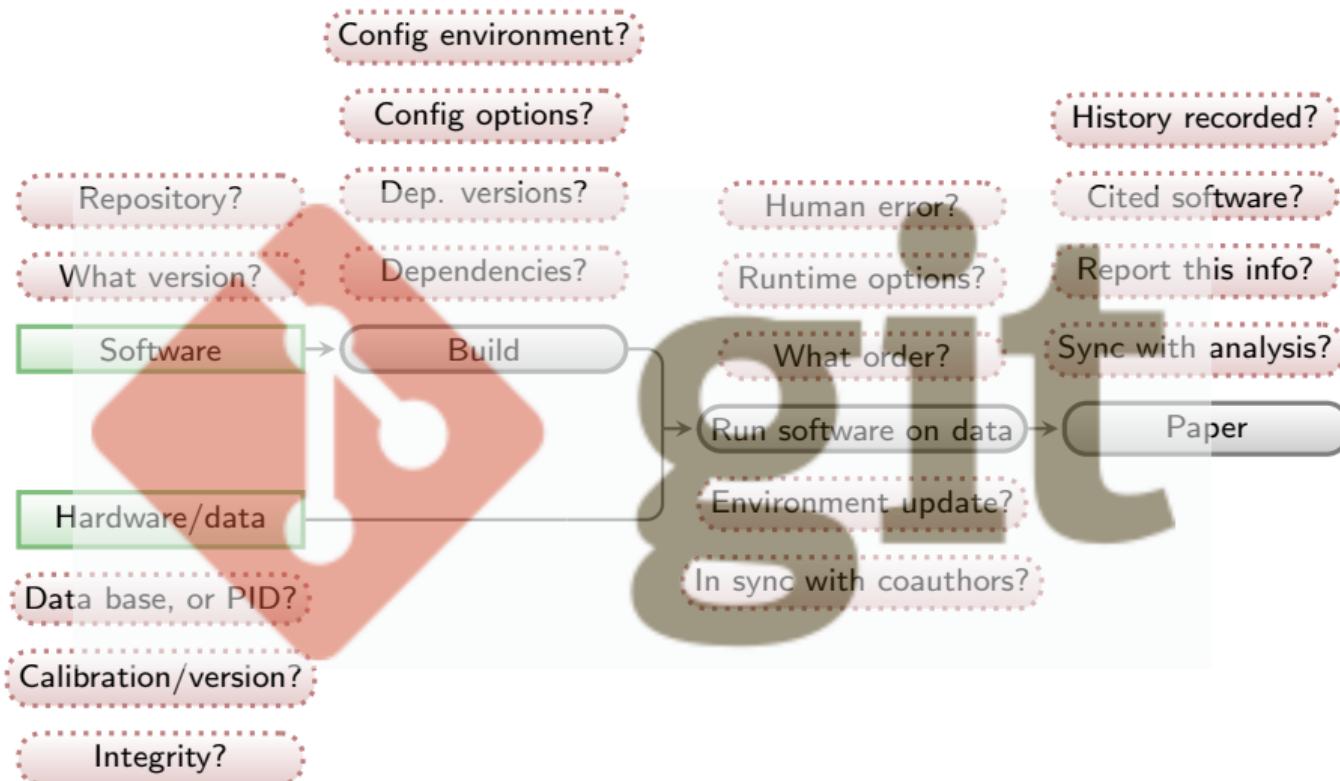
The analysis scripts write/update the \LaTeX macro values automatically.

```
# Numbers for dettf.tex:  
sqnt=999999  
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
```

Everything in plain text (machine and human readable)

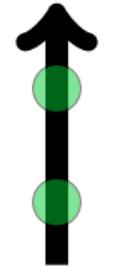


Everything in plain text (machine and human readable)



New projects branch from template

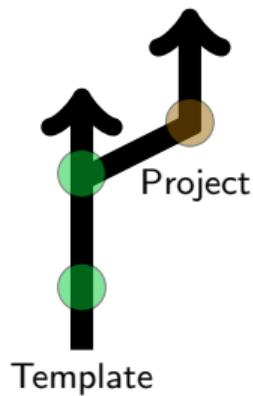
- ▶ Template's history is recorded.



Template

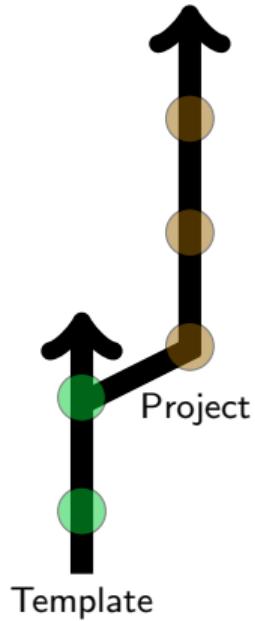
New projects branch from template

- ▶ Template's history is recorded.
- ▶ New project: a branch from the template.

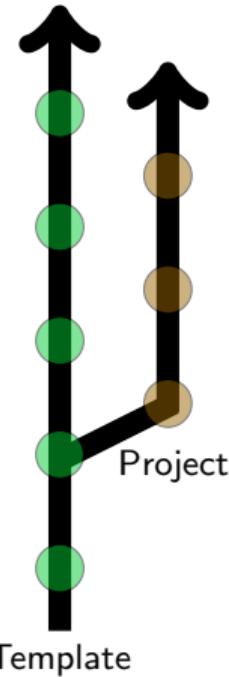


New projects branch from template

- ▶ Template's history is recorded.
- ▶ New project: a branch from the template.
- ▶ Research progresses in the project branch.

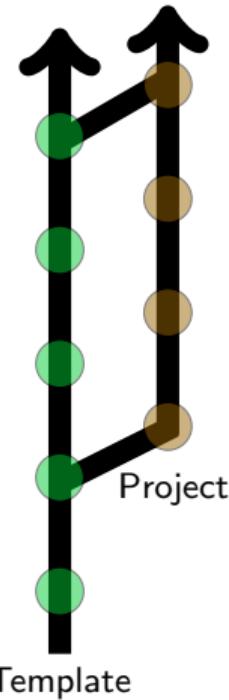


New projects branch from template



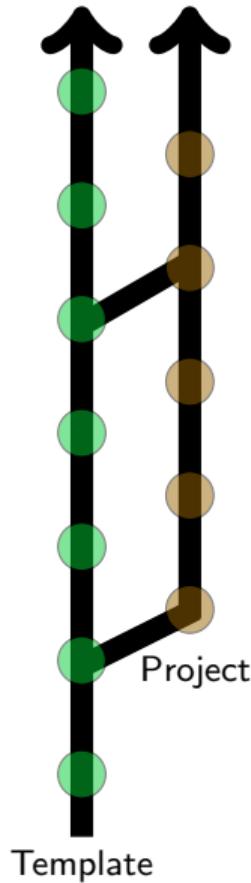
- ▶ Template's history is recorded.
- ▶ New project: a branch from the template.
- ▶ Research progresses in the project branch.
- ▶ Template will evolve (improved infrastructure).

New projects branch from template



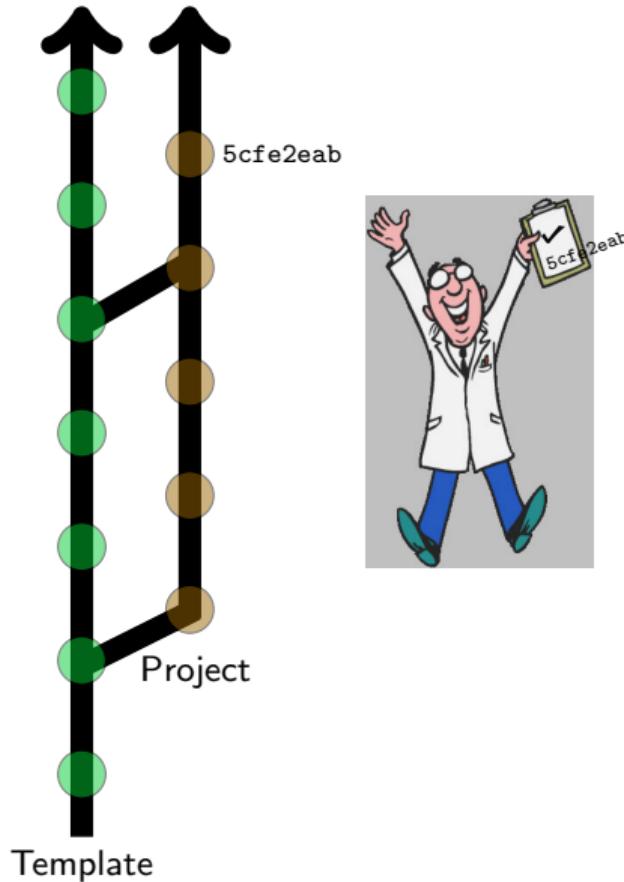
- ▶ Template's history is recorded.
- ▶ New project: a branch from the template.
- ▶ Research progresses in the project branch.
- ▶ Template will evolve (improved infrastructure).
- ▶ Template can be imported/merged back into project.

New projects branch from template



- ▶ Template's history is recorded.
- ▶ New project: a branch from the template.
- ▶ 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**.

New projects branch from template



- ▶ Template's history is recorded.
- ▶ New project: a branch from the template.
- ▶ 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 hash** is enough to verify the integrity of the result.

Project source and its execution

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

Publication of the project

A reproducible project using this template 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 template) can be **published** in

- ▶ **arXiv**: uploaded with the \TeX source to always stay with the paper (for example [arXiv:1505.01664](#)). The file containing all macros must also be uploaded so arXiv's server can easily build the \LaTeX source.
- ▶ **Zenodo**: Along with all the input datasets (many Gigabytes) and software (for example [zenodo.1164774](#)) and given a unique DOI.

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!).
- ▶ Applying **machine learning** on reproducible research projects will allow us to solve **Johan's Big Data Challenges** (from this morning):
 - ▶ "*Extract the relevant parameters automatically*" .
 - ▶ "*Translate the science we will see this week to enormous samples*" .
 - ▶ "*Believe the results when no one will have time to reproduce*" .
 - ▶ "*Have confidence in results derived using machine learning or AI*" .
- ▶ **Sixth paradigm of scientific discovery?**

GOOD NEWS: RDA adoption grant to IAC for this template



HORizon 2020

For this template, the **IAC** is selected as a **Top European organization** funded to adopt RDA Recommendations and Outputs.

- ▶ Research Data Alliance was launched by the [European Commission](#), NSF, National Institute of Standards and Technology, and the Australian Government's Department of Innovation.
- ▶ RDA Outputs are the technical and social infrastructure solutions developed by RDA Working Groups or Interest Groups that enable data sharing, exchange, and interoperability.

Summary:

A fully working template/framework is introduced 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*.
- ▶ A modification in one 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 <http://akhlaghi.org/pdf/reproducible-paper.pdf>.

For a technical description of the template's implementation, as well as a checklist to customize it, and tips on good practices, please see this page:

<https://gitlab.com/makhlaghi/reproducible-paper/blob/master/README-hacking.md>