Digging out the low-surface-brightness universe, **BIG** Data, **BIG** responsibility

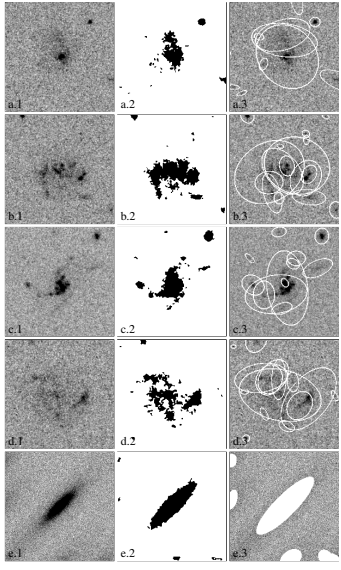
(a practical solution for reproducibile science)

Mohammad Akhlaghi

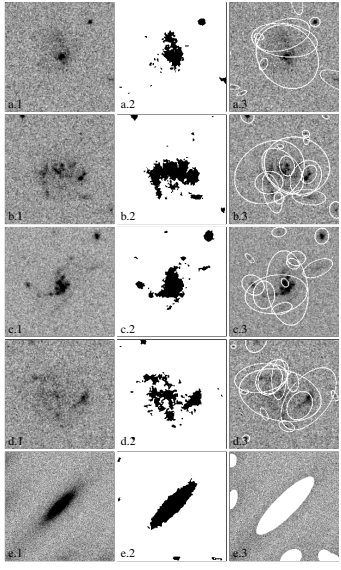
Instituto de Astrofísica de Canarias (IAC),
Tenerife, Spain

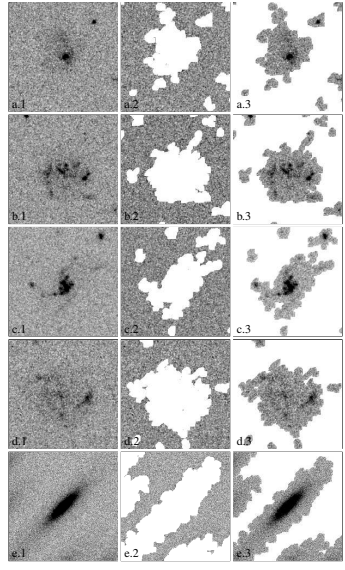
IAU Symposium 355, Session 2, July 8th, 2019
Tenerife, Spain

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)



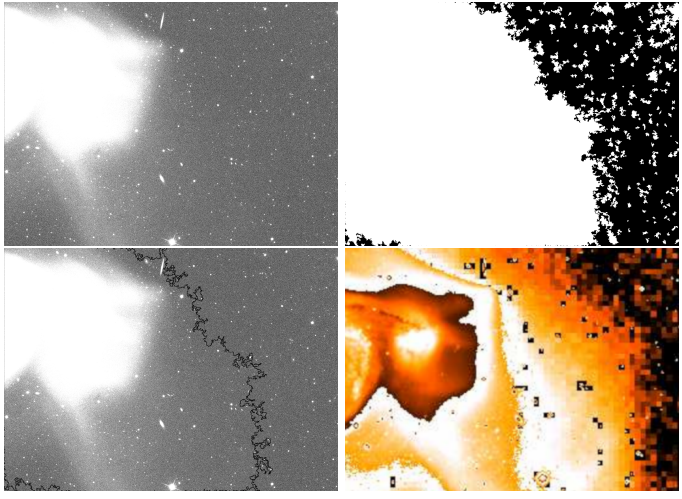


Reproducibility is critically important for low surface brightness astronomy

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

- Outer wing detected to $\mathrm{S/N} = 1/20$ or ~ 30 mag/arcsec^2.
- 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).
- 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).

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).

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).

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).

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).

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).

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).**

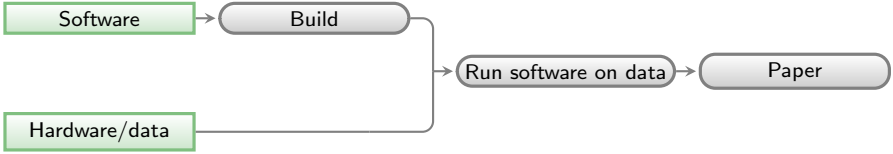
http://slittlefair.staff.shef.ac.uk

Software/Deterministic reproducibility

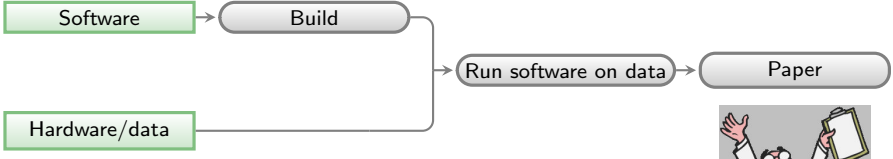
- Involves data analysis, or simulations.
- Starts after data is collected/digitized.
- Example: $2 + 2 = 4$ (i.e., sum of datasets).
- **DISCUSSED HERE.**

https://tsongas.com

General outline of a project

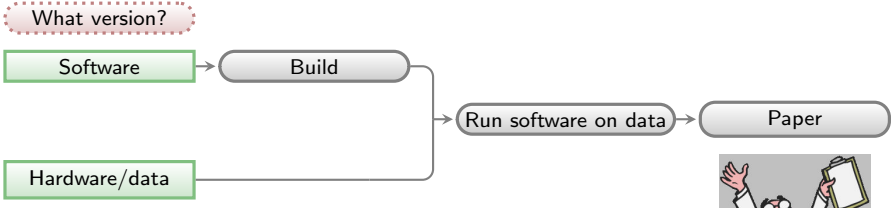


General outline of a project



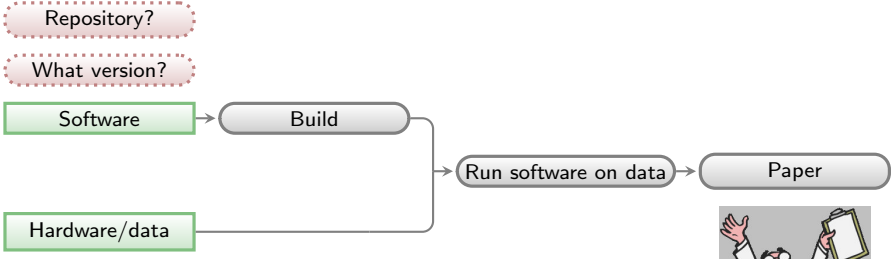
https://heywhatwhatdidyousay.wordpress.com

General outline of a project



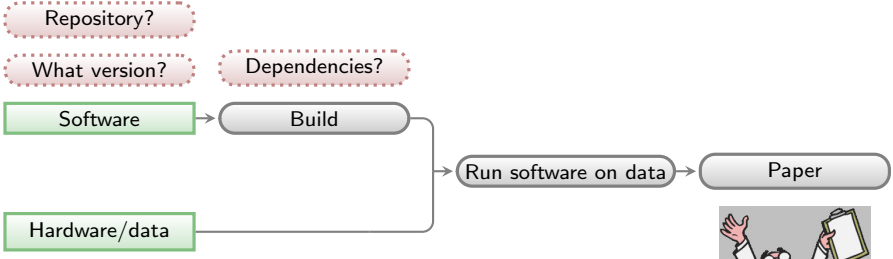
https://heywhatwhatdidyousay.wordpress.com

General outline of a project



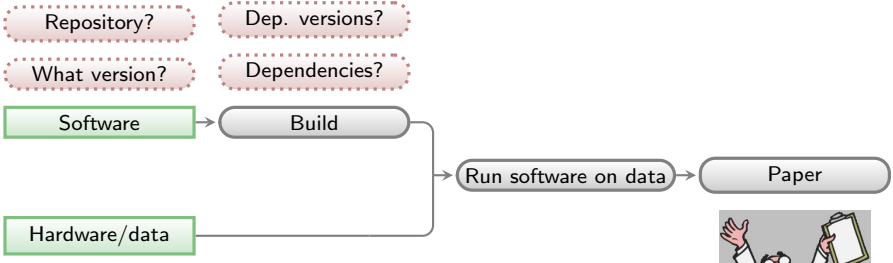
https://heywhatwhatdidyousay.wordpress.com

General outline of a project



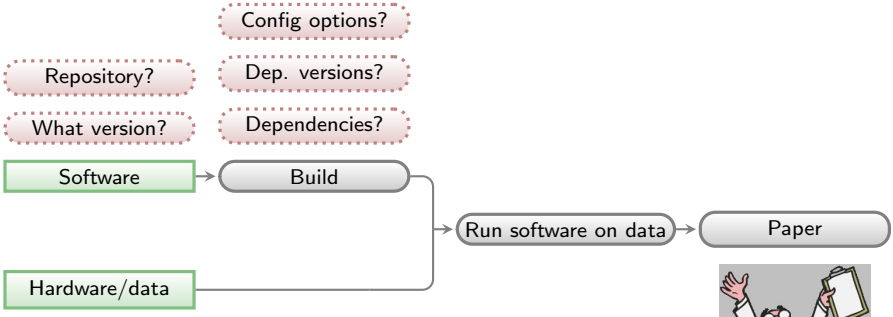
https://heywhatwhatdidyousay.wordpress.com

General outline of a project



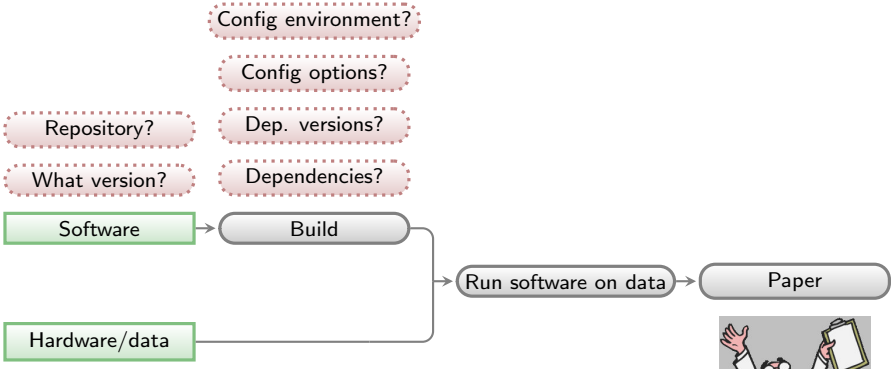
https://heywhatwhatdidyousay.wordpress.com

General outline of a project



https://heywhatwhatdidyousay.wordpress.com

General outline of a project



https://heywhatwhatdidyousay.wordpress.com

Example: Matplotlib (a Python visualization library) build dependencies

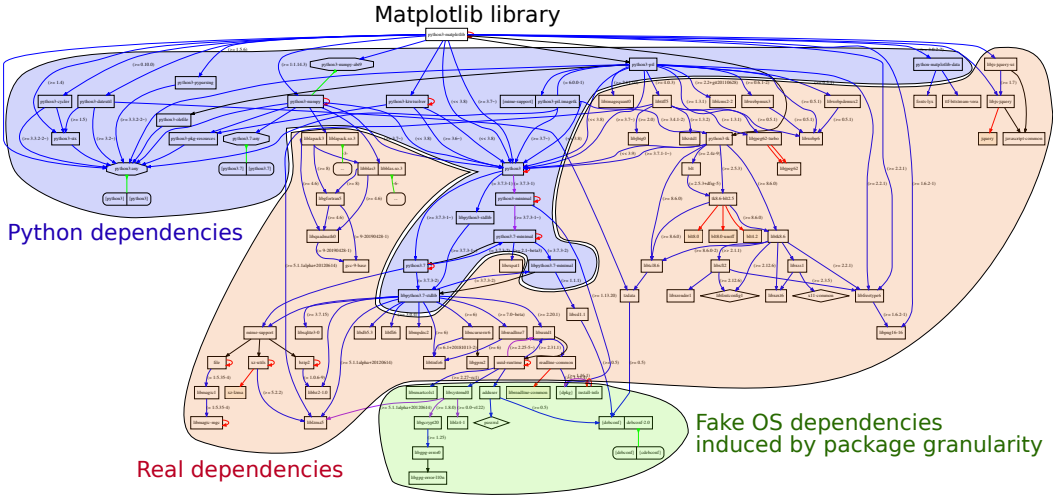
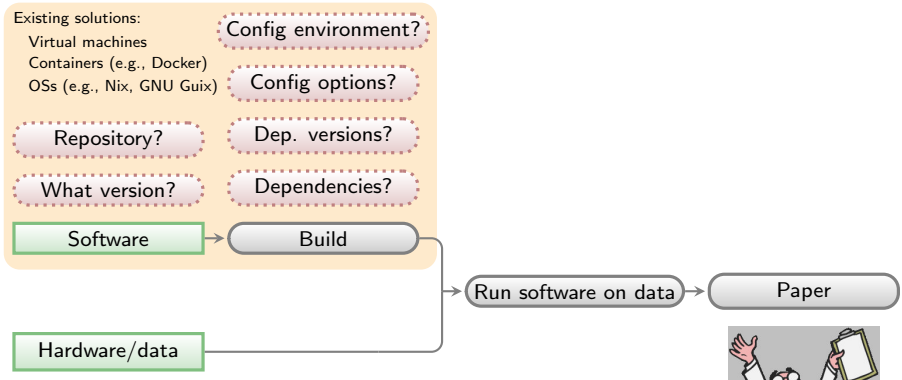


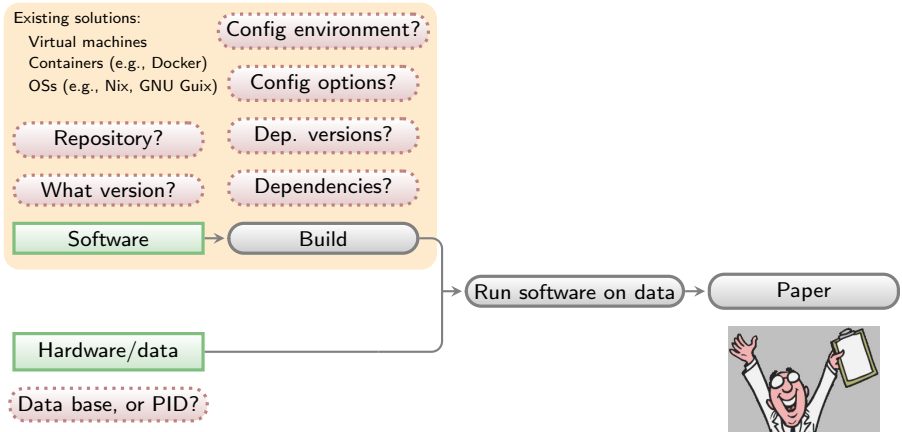
Fig. 1. Transitive dependencies of the software environment required by a simple "`import matplotlib`" command in the Python 3 interpreter.

General outline of a project



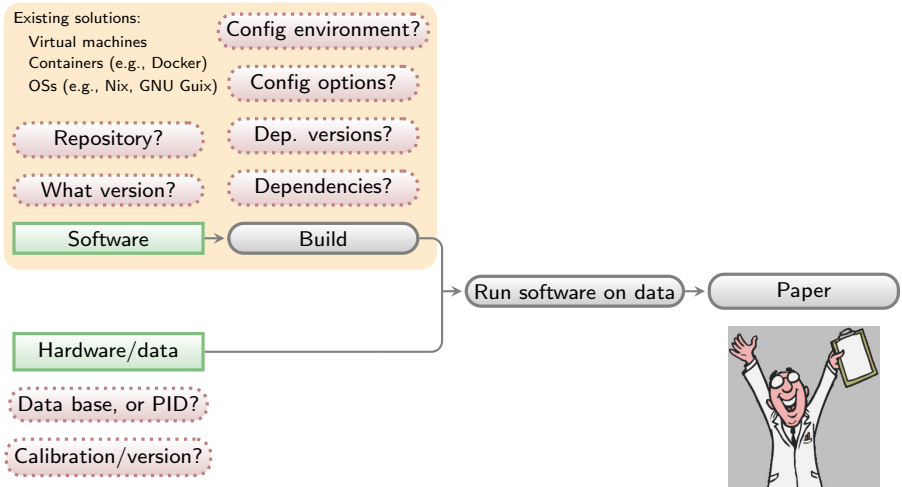
https://heywhatwhatdidyousay.wordpress.com

General outline of a project



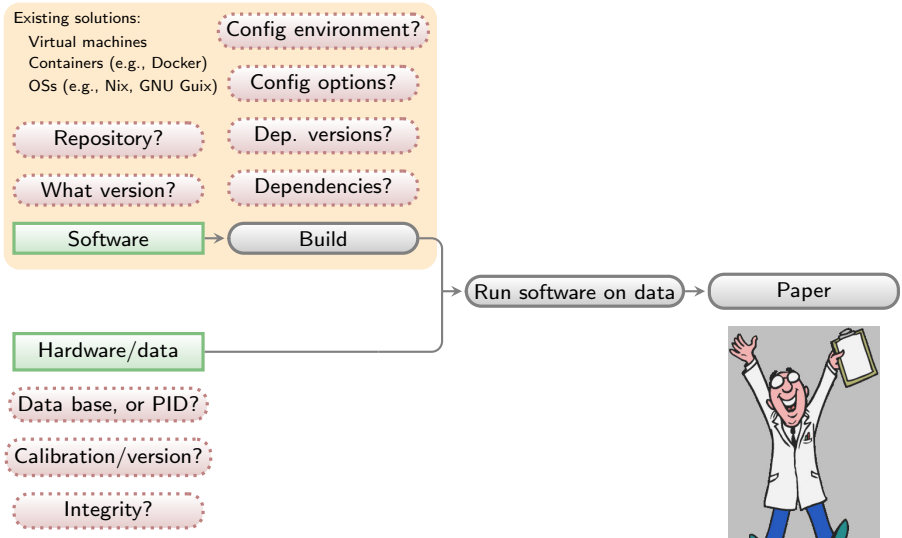
https://heywhatwhatdidyousay.wordpress.com

General outline of a project



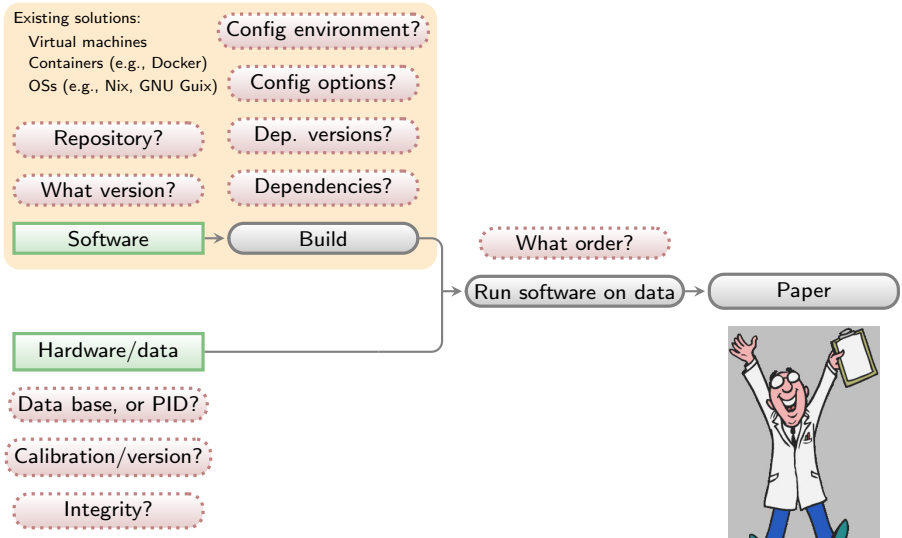
https://heywhatwhatdidyousay.wordpress.com

General outline of a project



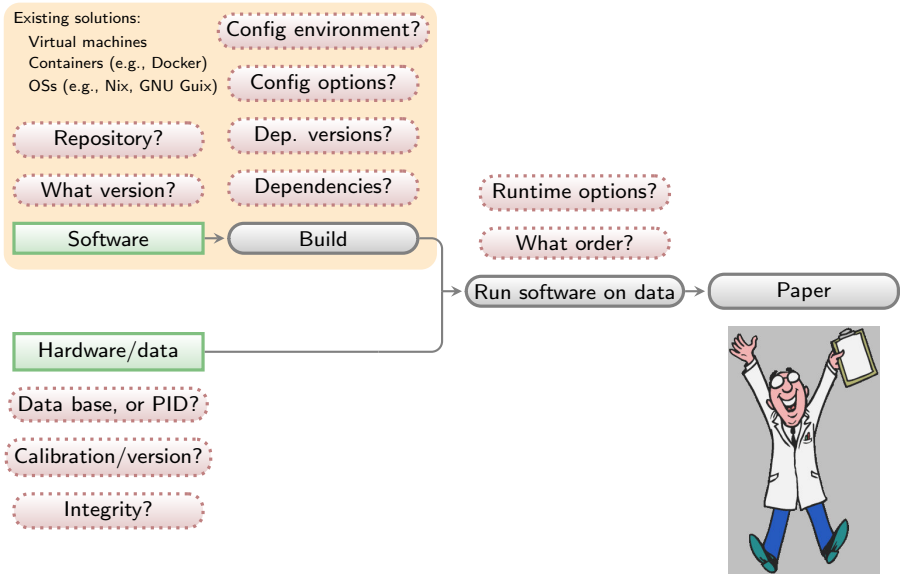
https://heywhatwhatdidyousay.wordpress.com

General outline of a project



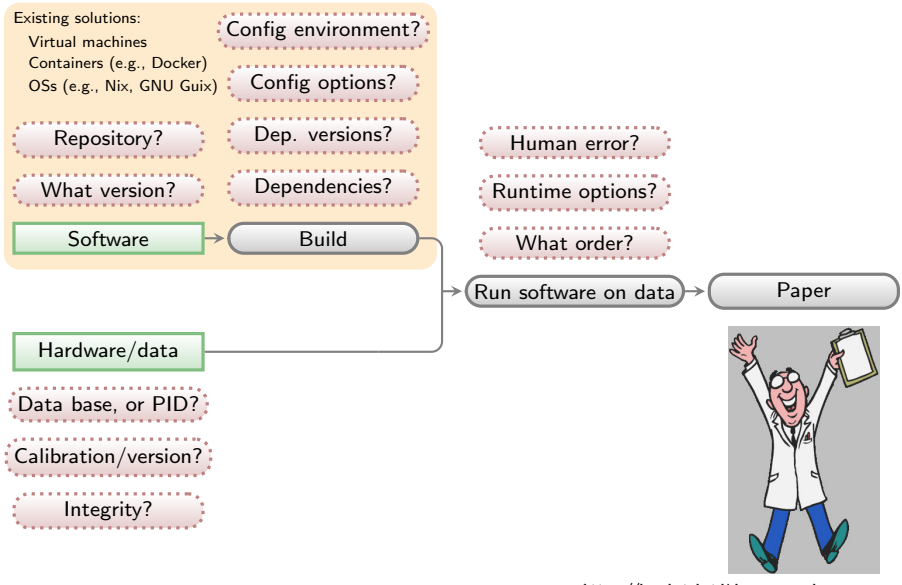
https://heywhatwhatdidyousay.wordpress.com

General outline of a project



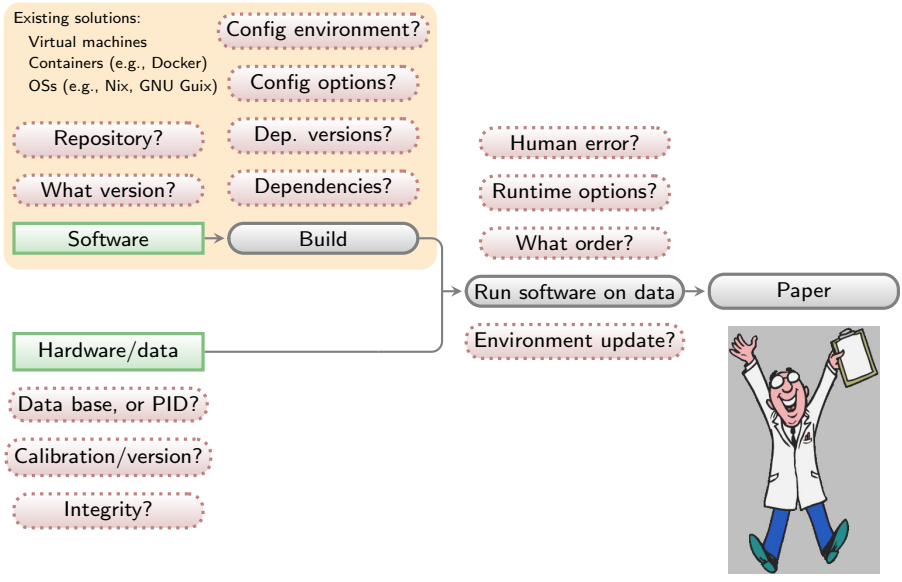
https://heywhatwhatdidyousay.wordpress.com

General outline of a project



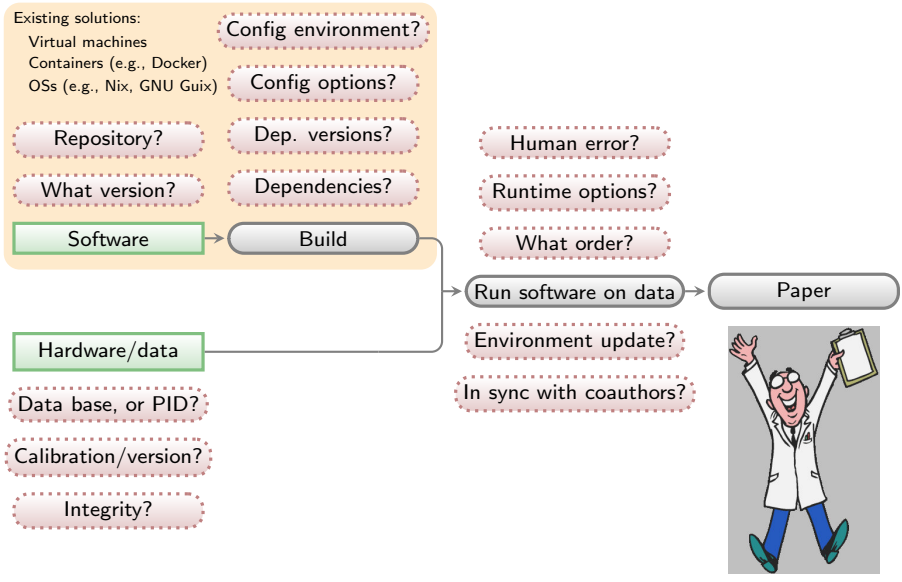
https://heywhatwhatdidyousay.wordpress.com

General outline of a project



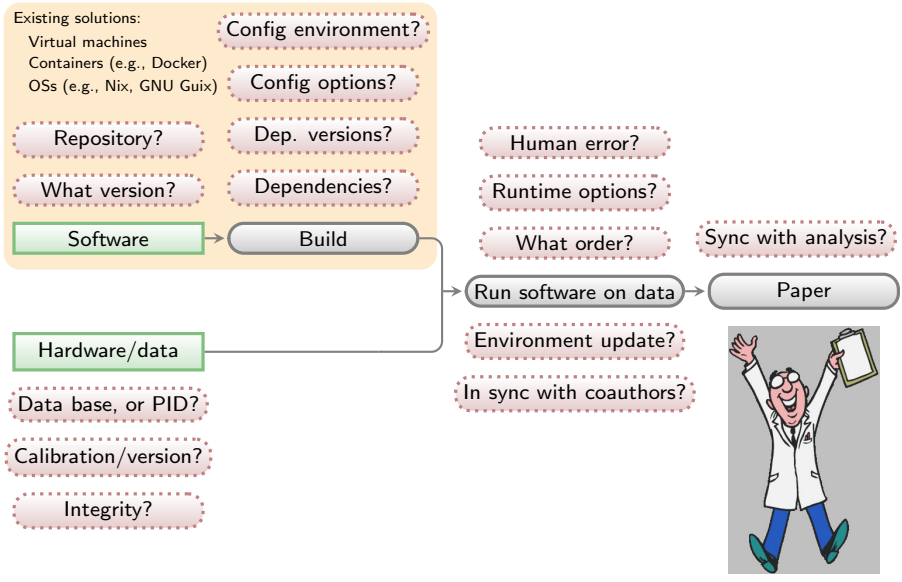
https://heywhatwhatdidyousay.wordpress.com

General outline of a project



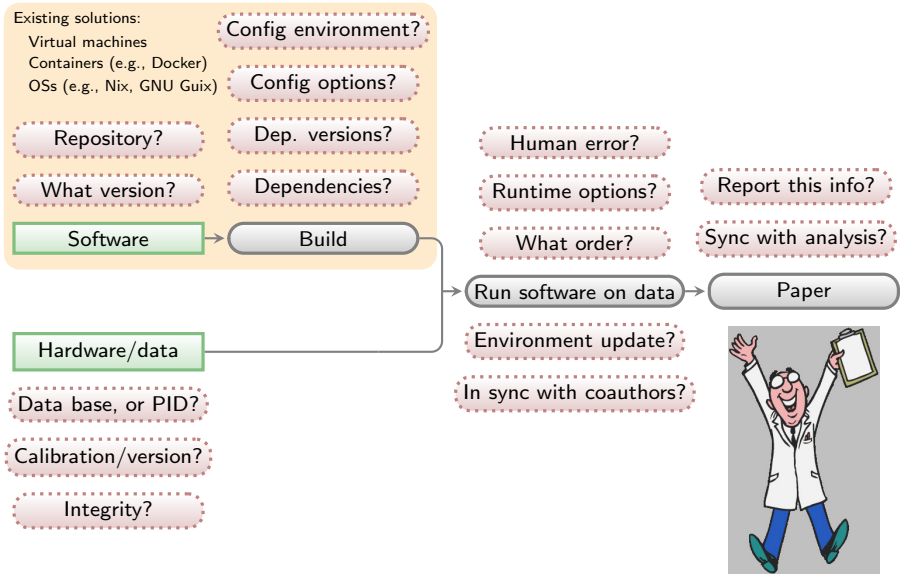
https://heywhatwhatdidyousay.wordpress.com

General outline of a project



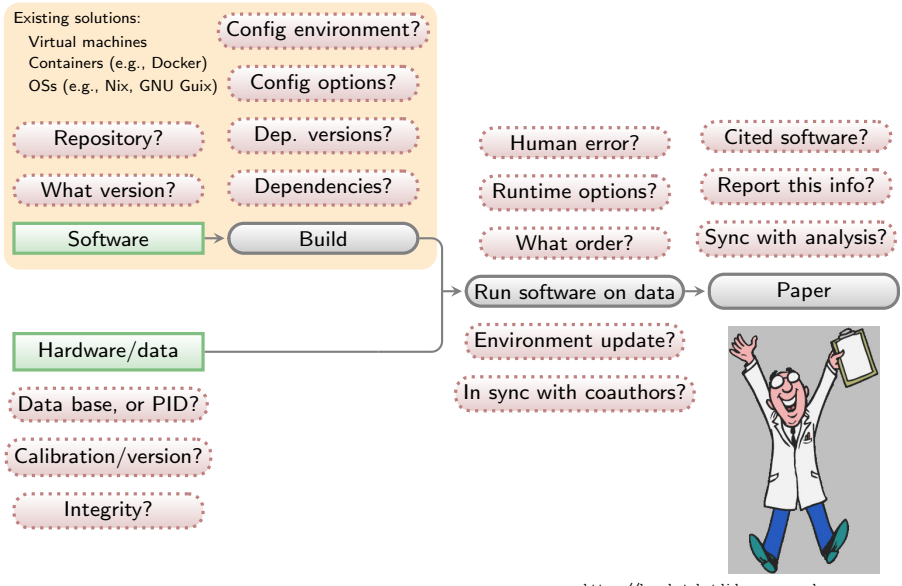
https://heywhatwhatdidyousay.wordpress.com

General outline of a project



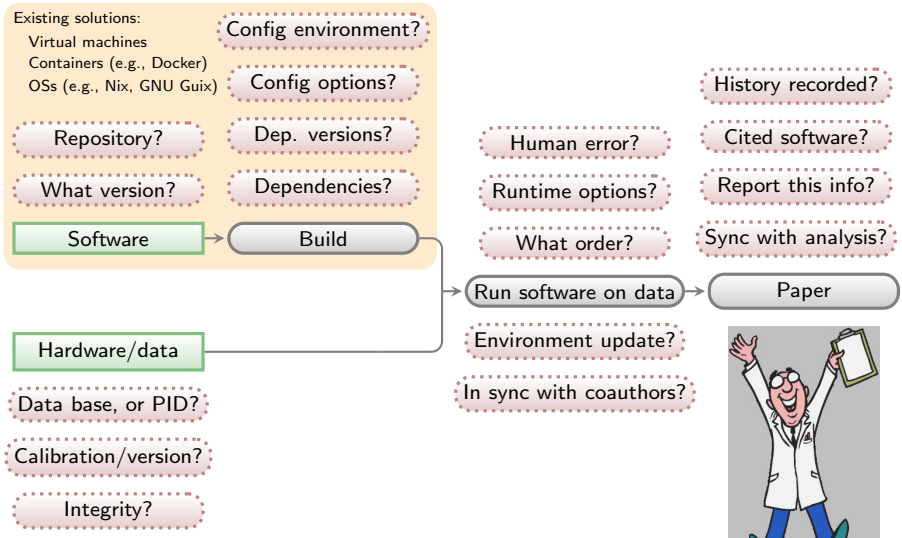
https://heywhatwhatdidyousay.wordpress.com

General outline of a project



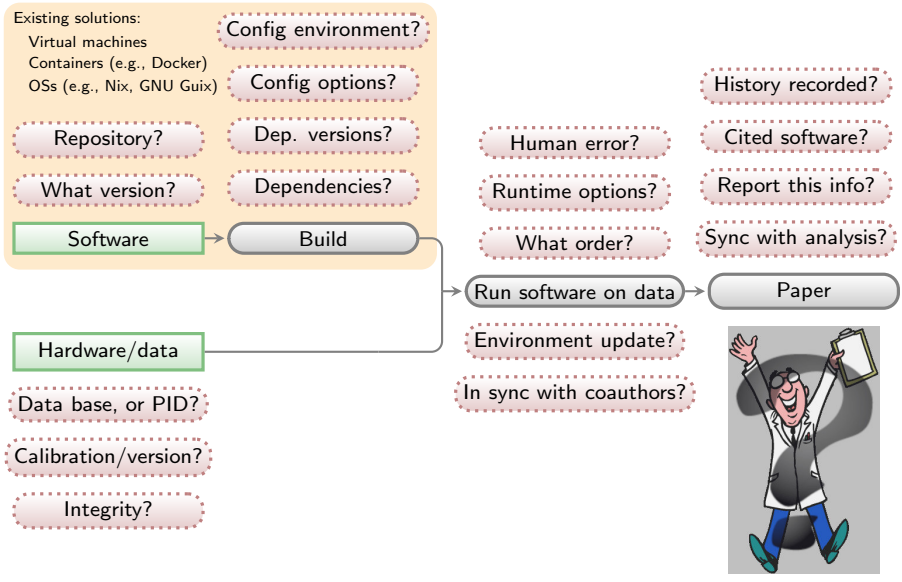
https://heywhatwhatdidyousay.wordpress.com

General outline of a project



https://heywhatwhatdidyousay.wordpress.com

General outline of a project



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

Science is a tricky business

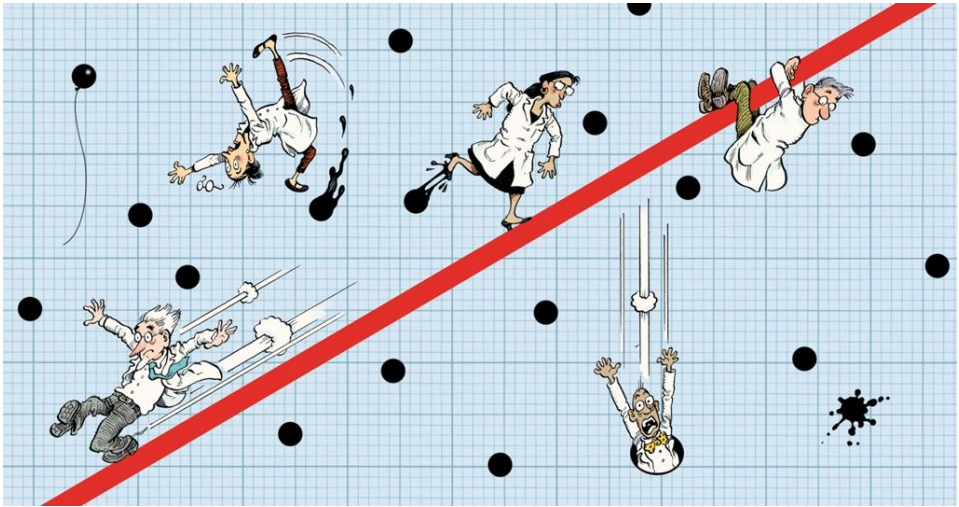


Image from nature.com ("Five ways to fix statistics", Nov 2017)

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.

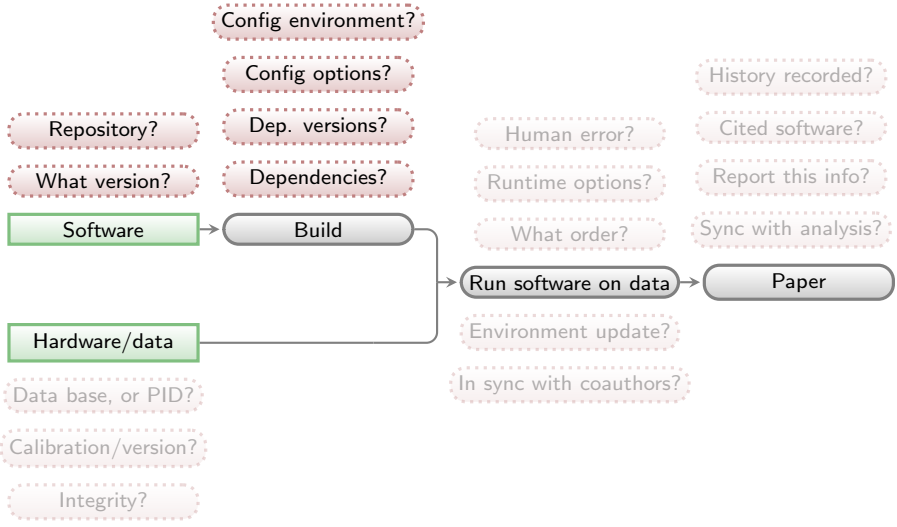
Necessity of (exactly) reproducible research

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

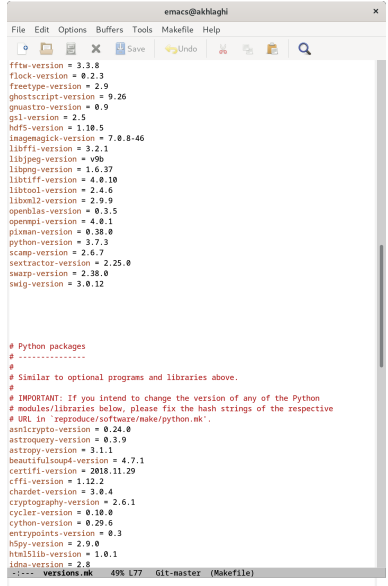


Predefined/exact software tools

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`).

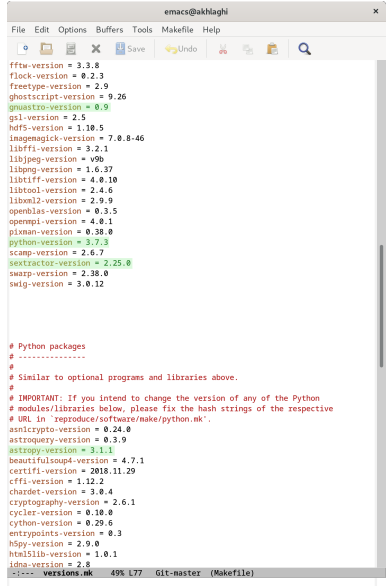


Predefined/exact software tools

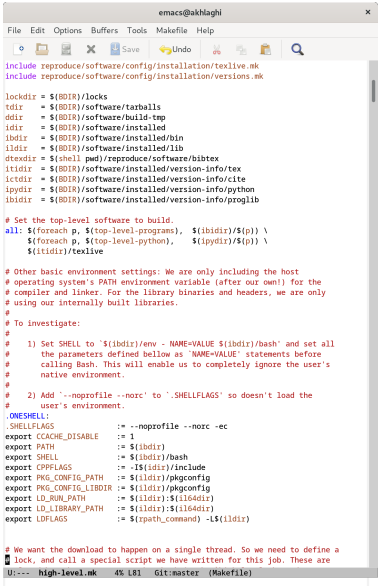
Reproducibility & software

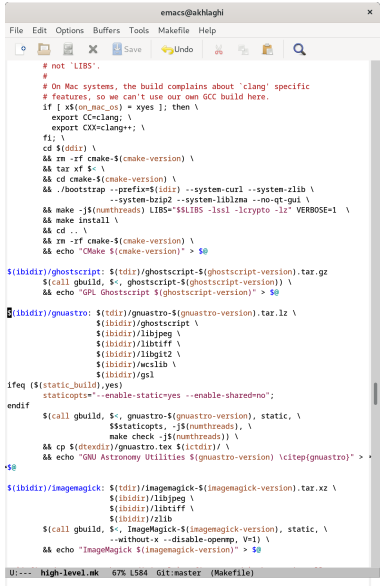
Reproducing the environment (specific software versions, build instructions and dependencies) is also critically important for reproducibility.

- *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`).

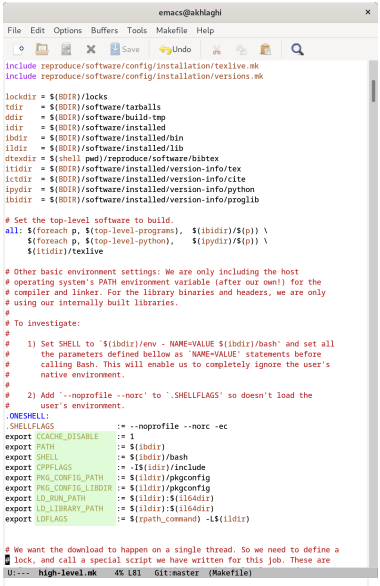


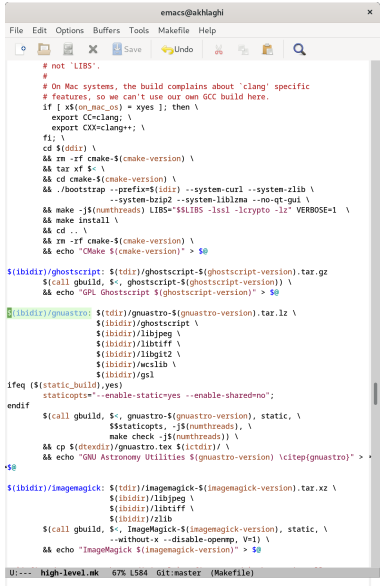
Controlled environment and build instructions





Controlled environment and build instructions





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 (0x00007fffdcbf7000)
    /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

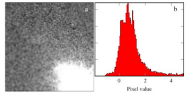


Figure 2: (a) An example image of the Wide-Field Planetary Camera 2, on board the Hubble Space Telescope from 1993 to 2009. This is one of the sample images from the FITS standard webpage, kept as examples for this file format. (b) Histogram of pixel values in (a).

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

Furthermore, since PGFPlots is built by LaTeX it respects all the properties of your text (for example line width and fonts and etc). Therefore the final plot blends in your paper much more nicely. It also has a wonderful manual[3].

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

2. NOTICE AND CITATIONS

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

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

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

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

3. ACKNOWLEDGEMENTS

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

This research was partly done using GNU Astronomy Utilities (Gnuastro, ascl.net/1801.009), and the reproducible paper template v0.364-g28dfc0.dirty. Work on Gnuastro and the reproducible paper template has been funded by the Japanese Ministry of Education, Culture, Sports, Science, and Technology (MEXT) scholarship and its Grant-in-Aid for Scientific Research (21244012, 24253003), the European Research Council (ERC) advanced grant 339659-MUSICOS, European Union's Horizon 2020 research and innovation programme under Marie Skłodowska-Curie grant agreement No 721463 to the SUNDIAL ITN, and from the Spanish Ministry of Economy and Competitiveness (MINECO) under grant number AYA2016-76219-P.

This research was done with the following free software programs and libraries: Bzip2 1.0.6, CFITSIO 3.45, CMake 3.14.2, cURL 7.63.0, Dmcentaj flock 0.2.3, File 5.36, FreeType 2.9, Git 2.21.0, GNU Astronomy Utilities 0.9 (Akhlaghi and Ichikawa 2015), GNU AWK 5.0.0, GNU Bash 5.0.7, GNU Binutils 2.32, GNU Compiler Collection (GCC) 9.1.0, GNU Coreutils 8.31, GNU Diffutils 3.7, GNU Findutils 4.6.0.199-e76c, GNU Grep 3.3, GNU Gzip 1.10, GNU Integer Set Library 0.18, GNU Libtool 2.4.6, GNU M4 1.4.18, GNU Make 4.2.90, GNU Multiple Precision Arithmetic Library 6.1.2, GNU Multiple Precision Complex library, GNU Multiple Precision Floating-Point Reliably 4.0.2, GNU NCURSES 6.1, GNU Readline 8.0, GNU Scientific Library 2.5, GNU Sed 4.7, GNU Tar 1.32, GNU Wget 1.20.3, GNU Which 2.21, GPL Ghostscript 9.26, HDF5 library 1.10.5, ImageMagick 7.0.8-46, Libbsd 0.9.1, Libgit2 0.26.0, Libjpeg v9b, Libpng 1.6.37, Libtiff 4.0.10, Lzip 1.20, Metastore (forked) 1.1.2-23-fa9170b, OpenBLAS 0.3.5, Open MPI 4.0.1, OpenSSL 1.1.1a, Patchelf 0.9, pkg-config 0.29.2, Python 3.7.3, Unzip 6.0, WCSLIB 6.2, XZ Utils 5.2.4, Zip 3.0 and Zlib 1.2.11. Within Python, the following modules were used: Astropy 3.1.1 (Astropy Collaboration et al. 2013), Astropy Collaboration et al. 2018), Cycler 0.10.0, Cython 0.29.6 (Behnel et al. 2011), h5py 2.9.0, Kiwisolver 1.0.1, Matplotlib 3.0.2 (Hunter 2007), Numpy 1.16.2 (van der Walt et al. 2011), pkgconfig 1.5.1., PyParsing 2.3.1, python-dateutil 2.8.0, Scipy 1.2.1 (Oliphant 2007; Millman and Aivazis 2011), Setuptools 40.8.0, Setuptools-scm 3.2.0 and Six 1.12.0. The LaTeX source of the paper was compiled to make the PDF using the following packages: biber 2.12, bibtex 2.12, bibtex 3.12, bibtexu 3.12, caption 2018-10-05, caption 2018-10-05, courier 2016-06-24, courier 2016-06-24, csquotes 5.2d, datetime 2.60, datetime 2.60, ec 1.0, ec 1.0, etoolbox 2.5f, etoolbox 2.5f, fancyhdr 3.10, fancyhdr 3.10, fontcount 3.05, fontcount 3.05, fontaxes 1.0d, fontaxes 1.0d, fontsite 5.5b, fontsmisc 5.5b, fp 2.1d, fp 2.1d, logreq 1.0, logreq 1.0, newtx 1.554, newtx 1.554, pgf 3.1.2, pgf 3.1.2, pgfplots 1.16, pgfplots 1.16, preprint 2011, preprint 2011, setspace 6.7a, setspace 6.7a, tex 3.14159265, tex 3.14159265, texgyre 2.501, texgyre 2.501, times 2016-06-24, times 2016-06-24, titlesec 2.10.2, titlesec 2.10.2, txfonts 2016-06-24, txfonts 2016-06-24, ulem 2016-06-24, ulem 2016-06-24, xcolor 2.12, xcolor 2.12, xkeyval 2.7a and xkeyval 2.7a. We are very grateful to all their creators for freely providing this necessary infrastructure. This research (and many others) would not be possible without them.

References

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

Software citation automatically generated in paper (including Astropy)

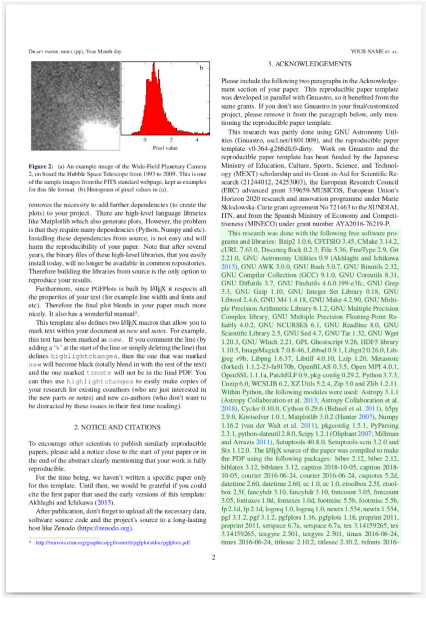


Figure 2: (a) An example image of the Wide-Field Planetary Camera 2, on board the Hubble Space Telescope from 1993 to 2009. This is one of the sample images from the FITS standard webpage, kept as examples for this file format. (b) Histogram of pixel values in (a).

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

Furthermore, since PGFPlots is built by LaTeX it respects all the properties of your text (for example line width and fonts and etc). Therefore the final plot blends in your paper much more nicely. It also has a wonderful manual[4].

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

2. NOTICE AND CITATIONS

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

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

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

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

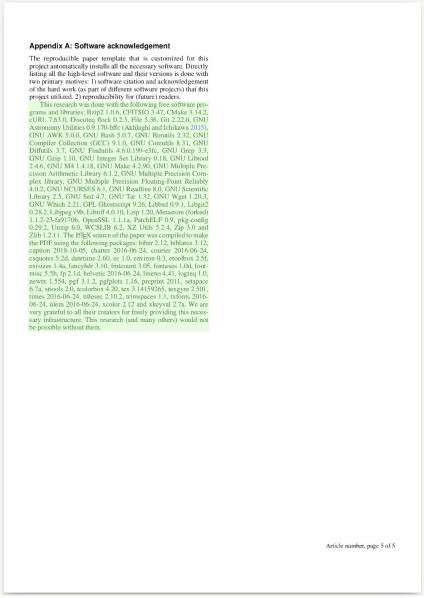
Software citation automatically generated in paper (only GNU Astronomy Utilities)

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 (as part of different software projects) that this project utilized. 2) reproducibility for (future) readers.

This research was done with the following free software programs and libraries: Bzip2 1.0.6, CFITSIO 3.47, CMake 3.14.2, cURL 7.63.0, Discoteq flock 0.2.3, File 5.36, Git 2.22.0, GNU Astronomy Utilities 0.9.170-bffc (Akhlaghi and Ichikawa 2015), GNU AWK 5.0.0, GNU Bash 5.0.7, GNU Binutils 2.32, GNU Compiler Collection (GCC) 9.1.0, GNU Coreutils 8.31, GNU Diffutils 3.7, GNU Findutils 4.6.0.199-e3fc, GNU Grep 3.3, GNU Gzip 1.10, GNU Integer Set Library 0.18, GNU Libtool 2.4.6, GNU M4 1.4.18, GNU Make 4.2.90, GNU Multiple Precision Arithmetic Library 6.1.2, GNU Multiple Precision Complex library, GNU Multiple Precision Floating-Point Reliably 4.0.2, GNU NCURSES 6.1, GNU Readline 8.0, GNU Scientific Library 2.5, GNU Sed 4.7, GNU Tar 1.32, GNU Wget 1.20.3, GNU Which 2.21, GPL Ghostscript 9.26, Libbsd 0.9.1, Libgit2 0.28.2, Libpng v96, Libtiff 4.0.10, Lzip 1.20, Metastore (forked) 1.1.2-23-fa9170b, 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 LaTeX source of the paper was compiled to make the PDF using the following packages: biber 2.12, biblatex 3.12, caption 2018-10-05, chartier 2016-06-24, courier 2016-06-24, csquotes 5.2d, datetime 2.60, ec 1.0, environ 0.3, etoolbox 2.5f, extsizes 1.4a, fancyhdr 3.10, fmtcount 3.05, fontaxes 1.0d, fontmisc 5.5b, fp 2.1d, helvetic 2016-06-24, hmemo 4.41, logreq 1.0, newtx 1.554, pgf 3.1.2, pgfplots 1.16, preprint 2011, setspace 6.7a, stmaryrd 2.0, xcolor@ox 4.20, tex 3.14159265, texgyre 2.501, times 2016-06-24, titlesec 2.10.2, trimspaces 1.1, txfonts 2016-06-24, 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)

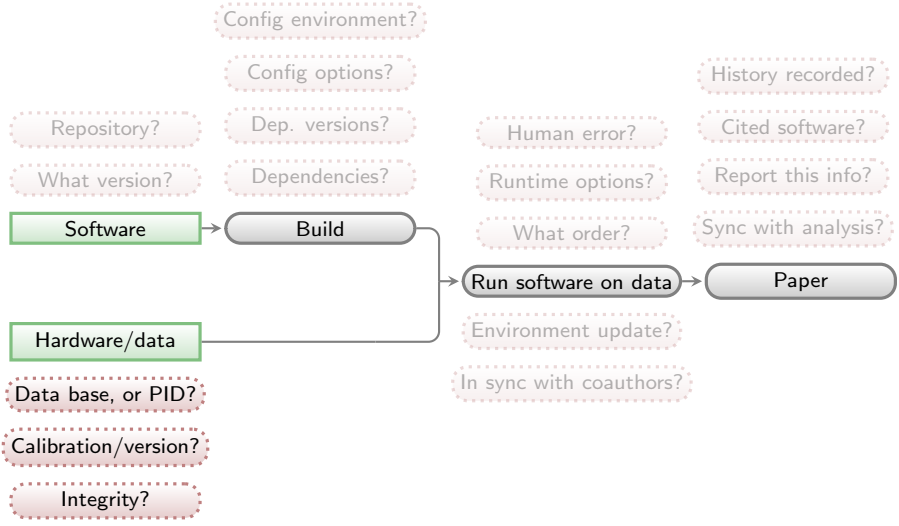


Appendix A: Software acknowledgement

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 (as part of different software projects) that this project utilized. 2) reproducibility for (future) readers.

This research was done with the following free software programs and libraries: Bzip2 1.0.6, CFITSIO 3.47, CMake 3.14.2, cURL 7.63.0, Discosig Bash 0.2.3, File 5.36, Git 2.22.0, GNU Astronomy Utilities 0.9.170-bffc (Akhlaghi and Ichikawa 2015), GNU AWK 5.0.0, GNU Bash 5.0.7, GNU Binutils 2.32, GNU Compiler Collection (GCC) 9.1.0, GNU Coreutils 8.31, GNU Diffutils 3.7, GNU Findutils 4.6.0.199-e3fc, GNU Grep 3.3, GNU Gzip 1.10, GNU Integer Set Library 0.18, GNU Libtool 2.4.6, GNU M4 1.4.18, GNU Make 4.2.90, GNU Multiple Precision Arithmetic Library 6.1.2, GNU Multiple Precision Complex library, GNU Multiple Precision Floating-Point Reliably 4.0.2, GNU NCURSES 6.1, GNU Readline 8.0, GNU Scientific Library 2.5, GNU Sed 4.7, GNU Tar 1.32, GNU Wget 1.20.3, GNU Which 2.21, GPL Ghostscript 9.26, Libbsd 0.9.1, Libgit2 0.28.2, Libpng v96, Libtiff 4.0.10, Lzip 1.20, Metastore (forked) 1.1.2-23-fa9170b, 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 LaTeX source of the paper was compiled to make the PDF using the following packages: biber 2.12, biblatex 3.12, caption 2018-10-05, charter 2016-06-24, courier 2016-06-24, cuppates 5.2d, datetime 2.60, ec 1.0, environ 0.3, etoolbox 2.5f, extsizes 1.4a, fancyhdr 3.10, fmtcount 3.05, fontaxes 1.0d, fourier 5.5b, fp 2.1d, helvetic 2016-06-24, listerine 4.41, logreq 1.0, newtx 1.554, pgf 3.1.2, pgfplots 1.16, preprint 2011, setspace 6.7a, situols 2.0, scalerbox 4.20, tex 3.14159265, texgyre 2.501, times 2016-06-24, titlesec 2.10.2, trimspaces 1.1, txfonts 2016-06-24, xkem 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.

General outline of a project



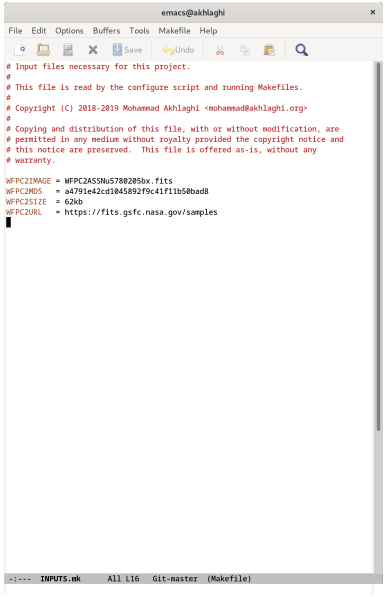
Input data source and integrity is documented and checked

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).



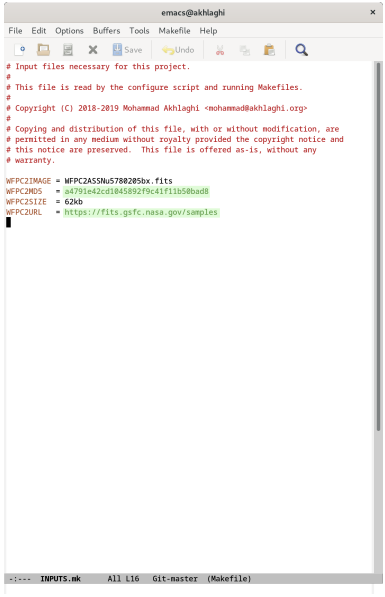
Input data source and integrity is documented and checked

Stored information about each input file:

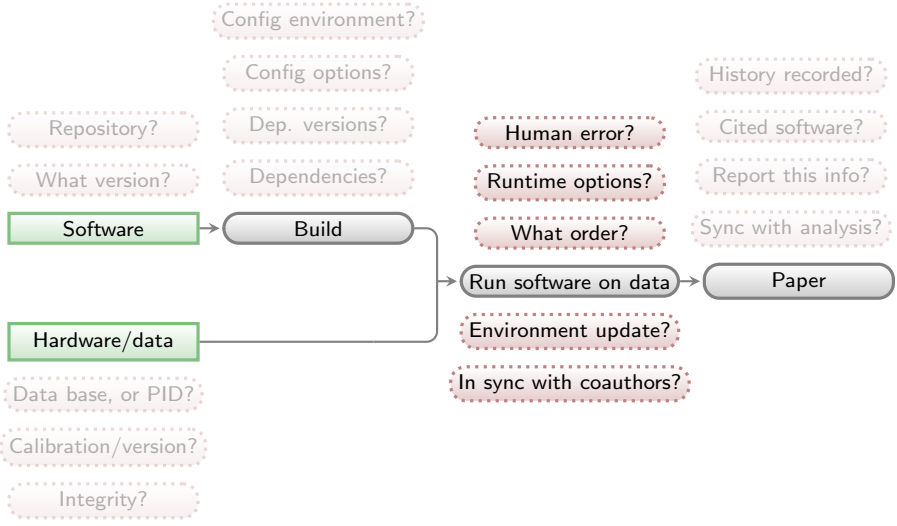
- PID (where available).
- Download URL.
- MD5-sum to check integrity.

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).



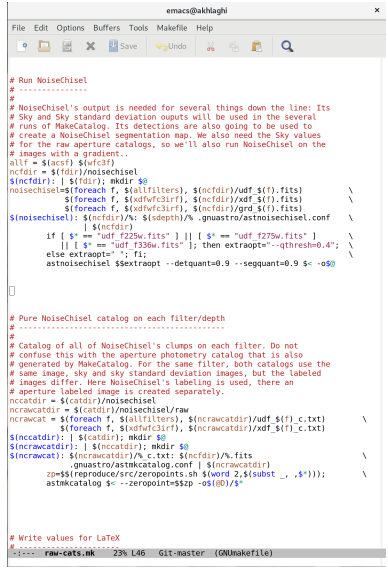
General outline of a project



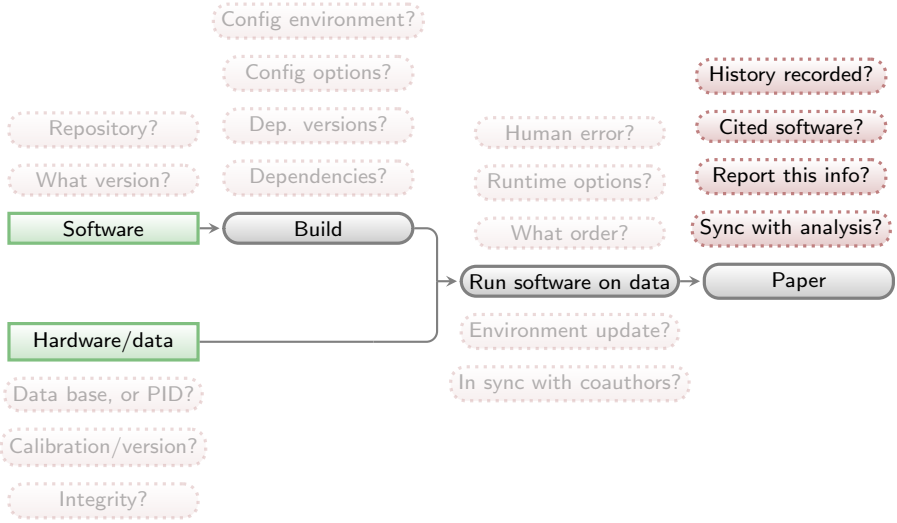
Reproducible science: Template is managed through a Makefile

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).



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).

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

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

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

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

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 S/N_T from the objects in R_s for the three examples in Figure 7 can be seen in column 5 (top) of that figure. Image processing effects, mainly due to shifting, rotating, and re-sampling the images for co-adding, on the real data further increase the size and count, and hence, the S/N of false detections in real, reduced/co-added images. A comparison of scales on the S/N histograms between the mock ((a.5.1) and (b.5.1)) and real (c.5.1) examples in Figure 7 shows the effect quantitatively. In the histograms of Figure 7, the bin with the largest number of false pseudo-detections respectively has an S/N of 1.89, 2.37, and 4.77.

The 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).

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

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

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

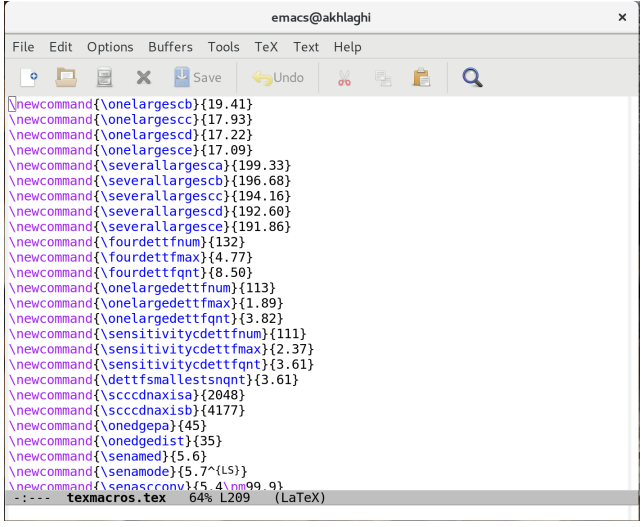
$$S/N_T = \frac{NF - NS_a}{\sqrt{NF + N\sigma_S^2}} = \frac{\sqrt{N}(F - S_a)}{\sqrt{F + \sigma_S^2}}. \quad (3)$$

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 S/N_T from the objects in R_s for the three examples in Figure 7 can be seen in column 5 (top) of that figure. Image processing effects, mainly due to shifting, rotating, and re-sampling the images for co-adding, on the real data further increase the size and count, and hence, the S/N of false detections in real, reduced/co-added images. A comparison of scales on the S/N histograms between the mock ((a.5.1) and (b.5.1)) and real (c.5.1) examples in Figure 7 shows the effect quantitatively. In the histograms of Figure 7, the bin with the largest number of false pseudo-detections respectively has an S/N of 1.89, 2.37, and 4.77.

The S/N_T distribution of detections in R_s provides a very ro-

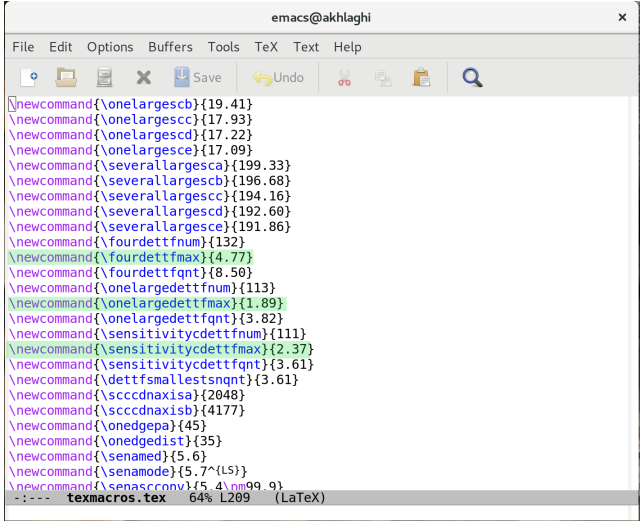
Analysis step results/values concatenated into a single file.

All LaTeX macros come from a single file.



Analysis step results/values concatenated into a single file.

All LaTeX macros come from a single file.



Analysis results stored as LaTeX macros

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

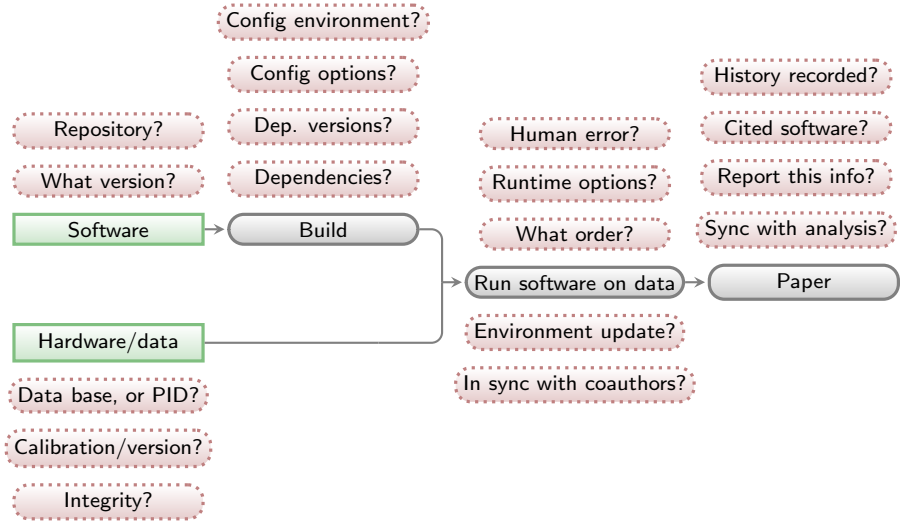
```
# Numbers for dettf.tex:
sqnt=9999999
function dettfhist
{
    # Set the file name.
    if [ $2 == 4 ]; then                obase=four;
    elif [ $2 = sensitivity3 ]; then    obase=sensitivityc;
    else                                obase=$2;
    fi
    if [ $2 == onelarge ]; then ind="_7"; else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

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

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

Analysis results stored as LaTeX macros

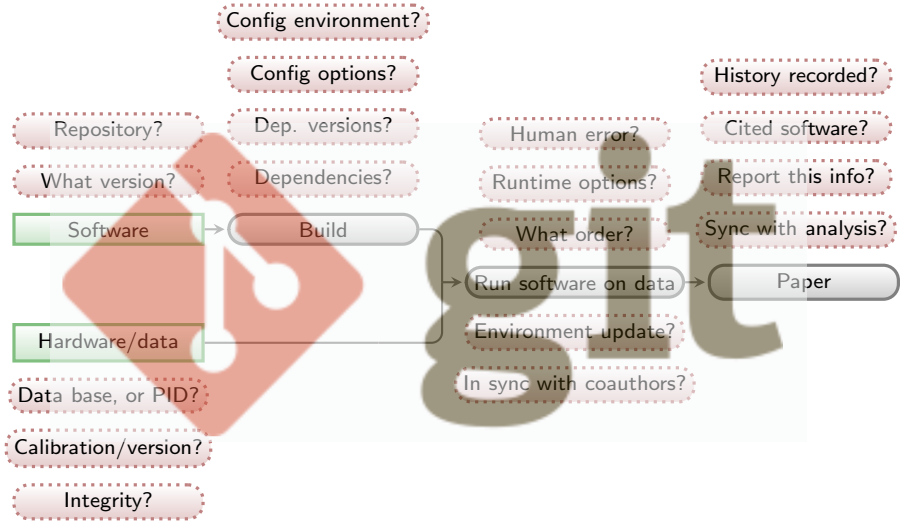
The analysis scripts write/update the LaTeX macro values automatically.

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

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

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

Everything in plain text (machine and human readable)



Everything in plain text (machine and human readable)



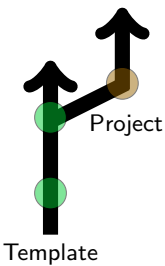
New projects branch from template

- Template's history is recorded.

Template

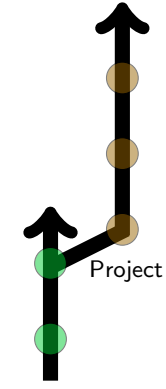
New projects branch from template

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

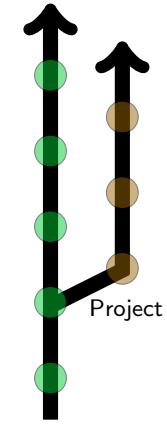


New projects branch from template

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

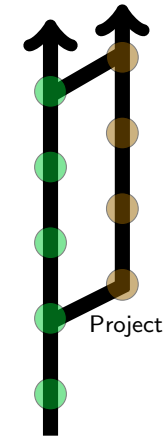


New projects branch from template



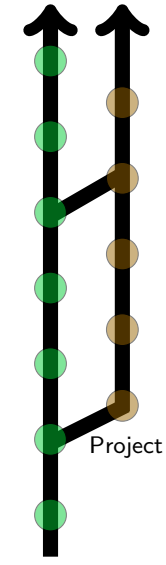
- Template's history is recorded.
- New project: a branch from the template.
- Research progresses in the project branch.
- Template will evolve (improved infrastructure).

New projects branch from template



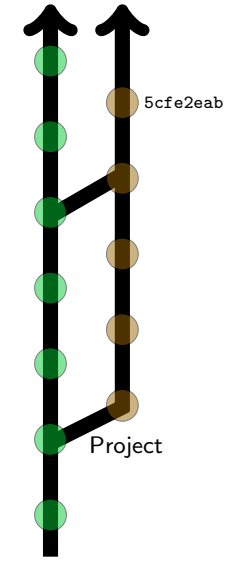
- Template's history is recorded.
- New project: a branch from the template.
- Research progresses in the project branch.
- Template will evolve (improved infrastructure).
- Template can be imported/merged back into project.

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 $\sim$ 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

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`