Reproducibility

Tim Rolff

3. Juli 2017

Reproducibility

Tim Rolff
3. Juli 2017

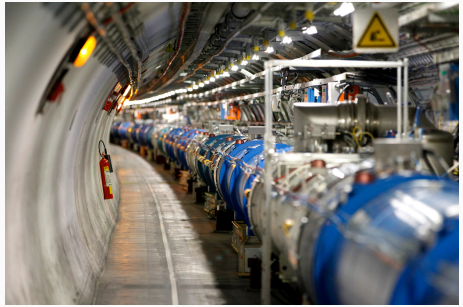
1 Why do we need Reproducibility
2 What is Reproducible Research?
3 Reproducibility in Empirical Science by Example
4 Reproducibility in Simulations
5 Best Practices

1 Why do we need Reproducibility
2 What is Reproducible Research?
3 Reproducibility in Empirical Science by Example
4 Reproducibility in Simulations
5 Best Practices

Why do we need Reproducibility

Why do we need Reproducibility

Reproducibility :: Why is it important?



Reproducibility :: Why is it important?

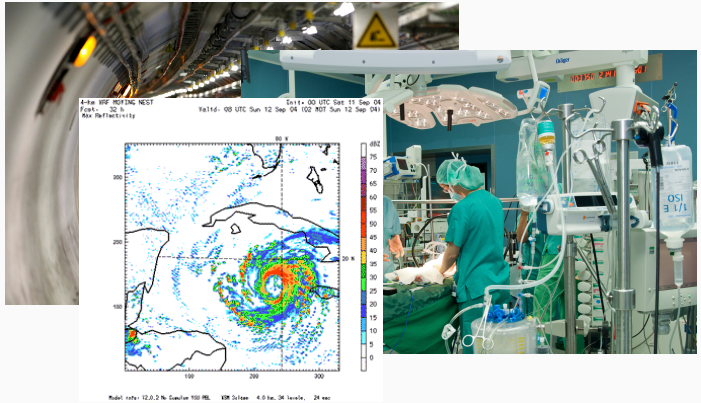
1. Because it affects everyone.
2. Supporting pillar of the empirical sciences
3. All scientific results should be reproducible in order to confirm the experiment and the resulting knowledge and to build upon them.
4. 47 out of 53 paper about cancer aren't reproducable ([Beg12])
   4.1 Medical treatment is based on research!
5. Not always possible to validate results from other researchers because of some of the experiments require a huge effort, e.g. LHC (Higgs particle, etc...) or stochastic processes

Reproducibility :: Why is it important?

Reproducibility :: Why is it important?

1. Because it affects everyone.
2. Supporting pillar of the empirical sciences
3. All scientific results should be reproducible in order to confirm the experiment and the resulting knowledge and to build upon them.
4. 47 out of 53 paper about cancer aren't reproducable ([Beg12])
   4.1 Medical treatment is based on research!
5. Not always possible to validate results from other researchers because of some of the experiments require a huge effort, e.g. LHC (Higgs particle, etc...) or stochastic processes

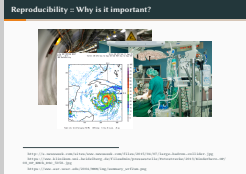
Reproducibility :: Why is it important?



http://s.newsweek.com/sites/www.newsweek.com/files/2015/04/07/large-hadron-collider.jpg

https://www.klinikum.uni-heidelberg.de/fileadmin/pressestelle/Fotostrecke/2013/Kinderherz-OP/08_OP_KHCh_DSC_5058.jpg

https://www.asr.ucar.edu/2004/MMM/img/summary_wrf2sm.png



1. Because it affects everyone.
2. Supporting pillar of the empirical sciences
3. All scientific results should be reproducible in order to confirm the experiment and the resulting knowledge and to build upon them.
4. 47 out of 53 paper about cancer aren't reproducable ([Beg12])
   4.1 Medical treatment is based on research!
5. Not always possible to validate results from other researchers because of some of the experiments require a huge effort, e.g. LHC (Higgs particle, etc...) or stochastic processes

What is Reproducible Research?

What is
Reproducible Research?

Reproducibility :: A Definition

> *Reproducibility is the ability of an entire analysis of an experiment or study to be duplicated, either by the same researcher or by someone else working independently, whereas reproducing an experiment is called replicating it.*
>
> *— Wikipedia Reproducibility*

https://en.wikipedia.org/wiki/Reproducibility, lastvisit:25.05.2017

Reproducibility :: A Definition
Reproducibility is the ability of an entire analysis of an experiment or study to be duplicated, either by the same researcher or by someone else working independently, whereas reproducing an experiment is called replicating it.
— Wikipedia Reproducibility

Reproducibility is the ability of an entire analysis of an experiment or study to be duplicated, either by the same researcher or by someone else working independently, whereas reproducing an experiment is called replicating it.

- Reproducibility verifies that an observation can be made repeatingly
- Reproducibility doesn't imply that the conclusions of a study is correct
- Reproducibility and replicability Often used interchangeably

Reproducibility :: A Definition

Reproducibility ≠ Replicability

[Dru09]

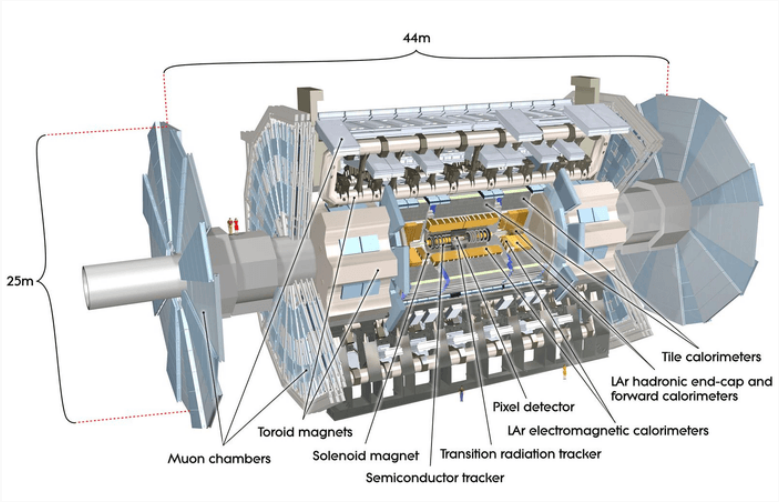
Reproducibility :: A Definition
Reproducibility ≠ Replicability
[Dru09]

1. Reproducibility:
   - 1.1 As defined in [Dru09]
   - 1.2 Outgoing from the data it is possible to get the same results as the original publisher, without using the same method
   - 1.3 Not restricted to using the same tools and instruments to measure the data
2. Replicability:
   - 2.1 As defined in [Dru09]
   - 2.2 Rebuild the exact project with its flaws
   - 2.3 Aims for the exact same data as the original work (even for wrong data)
   - 2.4 Leads to problems if the measuring was wrong or instruments where not calibrated correctly

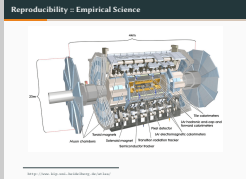
Reproducibility
in Empirical Science by Example

Reproducibility
in Empirical Science by Example

Reproducibility :: Empirical Science



http://www.kip.uni-heidelberg.de/atlas/

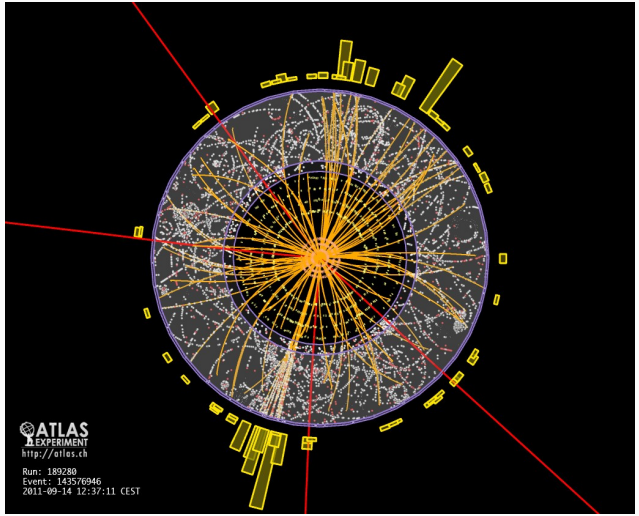


- Focus on gathering data
- Events aren't often deterministic
- High probability of measuring random data
- Empirical Sciences depends often on statistical methods.
  - → No exact solution possible
- Sometimes no second setup exist (e.g. Hubble)
- ATLAS / CMS:
  - Higgs detection
  - Huge range of particle detectors
  - Expensive to run 1.3TWh/a ≈ 60million euro

    `https://www.lhc-closer.es/taking_a_closer_look_at_lhc/0.energy_consumption`

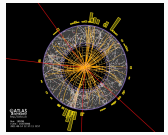
    `https://home.cern/about/engineering/powering-cern`
  - Same for HPC (DKRZ ≈ 1.4MW/a = $12.3 GWh/a$ (Mistral))

Reproducibility :: Empirical Science



http://www.ph.ed.ac.uk/sites/default/files/legacy/1112301_01-A4-at-144-dpi.jpg

Reproducibility :: Empirical Science



- Events are happening every 25ns or 40 million times per second
- Prefiltering of events necessary to store the data onto disk problematic for reproduciblility

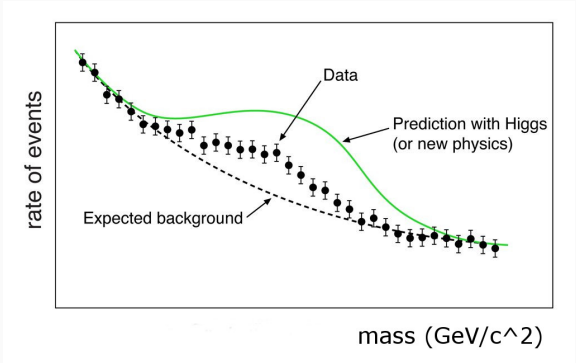
How did they make the Higgs experiment reproducible?

https://smolinacalvo.files.wordpress.com/2012/02/higgs_limits_simplified_5b.jpg

Reproducibility :: Empirical Science
How did they make the Higgs experiment reproducible?

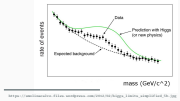
- $7\sigma \mathrel{\hat{=}} 10^{-10}\%$ of a random measured result
  (https://phys.org/news/2012-12-cms-atlas-higgs-like-particle-sigma.html)
- Through measuring the significance
- Small mistakes can make such a experiment irreproducible
  like Faster than Light Neutrinos
  (https://en.wikipedia.org/wiki/Faster-than-light_neutrino_anomaly#Measurement_errors)
  - A loose link from a GPS receiver which increased the delay through the fiber
  - A clock on an electronic board ticked faster than its expected 10 MHz frequency
- Cheating makes it irreproducible!
  - Creative data (adaption or invention of datasets)

How did they make the Higgs experiment reproducible?



https://smolinacalvo.files.wordpress.com/2012/02/higgs_limits_simplified_5b.jpg

Reproducibility :: Empirical Science
How did they make the Higgs experiment reproducible?

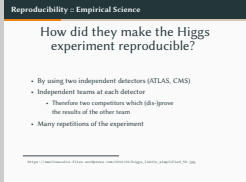


- $7\sigma \mathrel{\widehat{=}} 10^{-10}\%$ of a random measured result
  (https://phys.org/news/2012-12-cms-atlas-higgs-like-particle-sigma.html)
- Through measuring the significance
- Small mistakes can make such a experiment irreproducible
  like Faster than Light Neutrinos
  (https://en.wikipedia.org/wiki/Faster-than-light_neutrino_anomaly#Measurement_errors)
  - A loose link from a GPS receiver which increased the delay through the fiber
  - A clock on an electronic board ticked faster than its expected 10 MHz frequency
- Cheating makes it irreproducible!
  - Creative data (adaption or invention of datasets)

How did they make the Higgs experiment reproducible?

- By using two independent detectors (ATLAS, CMS)
- Independent teams at each detector
  - Therefore two competitors which (dis-)prove the results of the other team
- Many repetitions of the experiment

https://smolinacalvo.files.wordpress.com/2012/02/higgs_limits_simplified_5b.jpg



- $7\sigma \mathrel{\widehat{=}} 10^{-10}\%$ of a random measured result
  (`https://phys.org/news/2012-12-cms-atlas-higgs-like-particle-sigma.html`)
- Through measuring the significance
- Small mistakes can make such a experiment irreproducible
  like Faster than Light Neutrinos
  (`https://en.wikipedia.org/wiki/Faster-than-light_neutrino_anomaly#Measurement_errors`)
  - A loose link from a GPS receiver which increased the delay through the fiber
  - A clock on an electronic board ticked faster than its expected 10 MHz frequency
- Cheating makes it irreproducible!
  - Creative data (adaption or invention of datasets)

Reproducibility in Simulations

Reproducibility in Simulations

Reproducibilty :: Simulations

ICON-Model

https://www.cleanenergywire.org/sites/default/files/styles/lightbox_image/public/images/factsheet/dwd-icon-welt-bild1-1_0.jpg?itok=mIdcxQQX

- Used for worldwide weather forecasting / climate prediction
- Support nested grids for finer detail.
- Constructed through a icosahedral grid
  - Avoid polar singularities
  - Calculates on a finer mesh than the previous model
  - Higher surface (75km)
    - Low-level airflow
    - Rivers
    - Islands

Reproducibilty :: Simulations

- 2,949,120 Triangles
- $173 km^2$ on average per triangle
- Total of 265 million grid points
  - Prognostic variables are located in the circumcenter
  - Wind directions at the midpoint of the edges
  - Smaller models at specific areas (e.g. COSMO-EU, COSMO-DE)
    - Got replaced at 30.11.2016 by ICON-NEST [Hel16]
- 900 GB of data for a 7-day forecast

[DWDoj]

Reproducibilty :: Simulations
- 2,949,120 Triangles
- $173 km^2$ on average per triangle
- Total of 265 million grid points
  - Prognostic variables are located in the circumcenter
  - Wind directions at the midpoint of the edges
  - Smaller models at specific areas (e.g. COSMO-EU, COSMO-DE)
    - Got replaced at 30.11.2016 by ICON-NEST [Hel16]
- 900 GB of data for a 7-day forecast

[DWDoj]

- A about a quarter the size of Hamburg $\frac{755.22 km^2}{173 km^2} \approx 4.37$
- Replacement happened because the new ICON model supports nested areas with a smaller resolution.
- Also expensive to run
- Not available for everyone (hard to check reproducibility)

Is it not equal to an empirical experiment?

Reproducibilty :: Simulations

Is it not equal to an empirical experiment?

- Additional layer of uncertainty
- Additionally, need to deal with:
  - Float-point errors
  - Nondeterministic (partial-) results
  - Errors from input data
  - Hardware effects
    E.g. ECC or Processor errors (Ryzen FMA3 or the Pentium-FDIV-Bug)
  - Unknown internals of the Hardware

Reproducibilty :: Simulations

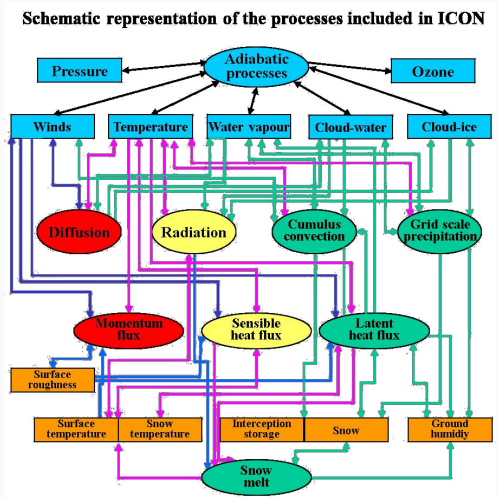
Is it not equal to an empirical experiment?

Not Quite!

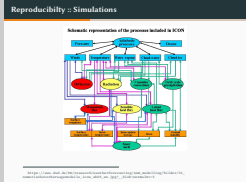
Reproducibilty :: Simulations
Is it not equal to an empirical experiment?
Not Quite!

- Additional layer of uncertainty
- Additionally, need to deal with:
  - Float-point errors
  - Nondeterministic (partial-) results
  - Errors from input data
  - Hardware effects
    E.g. ECC or Processor errors (Ryzen FMA3 or the Pentium-FDIV-Bug)
  - Unknown internals of the Hardware

Reproducibilty :: Simulations



https://www.dwd.de/EN/research/weatherforecasting/num_modelling/bilder/01_numerischevorhersagemodelle_icon_abb3_en.jpg?__blob=normal&v=3

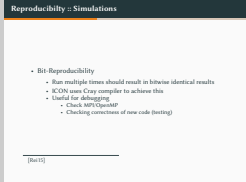


- No possibility to change real world variables (only observation)
  - Hard to predict what went wrong
  - Hard to test reproducibility and perform general tests
- Huge amount of interactions between the subsystems
- Multiple scalar / vector values on each cell

Reproducibilty :: Simulations

- Bit-Reproducibility
  - Run multiple times should result in bitwise identical results
  - ICON uses Cray compiler to achieve this
  - Useful for debugging
    - Check MPI/OpenMP
    - Checking correctness of new code (testing)

[Rei15]



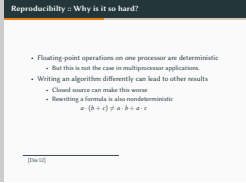
- Bit-Reproducibility results often in performance loss
- OpenMP checking is done by running the configuration multiple times and check if the results are bitwise identical (hint for OpenMP race condition) ([Rei15])
- MPI checking is done by changing the processor configuration and check if the results are bitwise identical (hint for MPI parallelization bug) ([Rei15])

Reproducibilty :: Why is it so hard?

- Floating-point operations on one processor are deterministic
  - But this is not the case in multiprocessor applications.
- Writing an algorithm differently can lead to other results
  - Closed source can make this worse
  - Rewriting a formula is also nondeterministic

$$a \cdot (b + c) \neq a \cdot b + a \cdot c$$

[Die12]



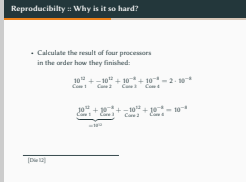
Reproducibilty :: Why is it so hard?

- Calculate the result of four processors
  in the order how they finished:

$$\underset{\text{Core 1}}{10^{12}} + \underset{\text{Core 2}}{-10^{12}} + \underset{\text{Core 3}}{10^{-8}} + \underset{\text{Core 4}}{10^{-8}} = 2 \cdot 10^{-8}$$

$$\underbrace{\underset{\text{Core 1}}{10^{12}} + \underset{\text{Core 3}}{10^{-8}}}_{=10^{12}} + \underset{\text{Core 2}}{-10^{12}} + \underset{\text{Core 4}}{10^{-8}} = 10^{-8}$$

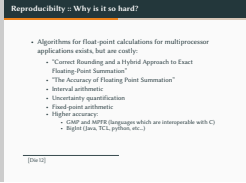
[Die12]



Reproducibilty :: Why is it so hard?

- Algorithms for float-point calculations for multiprocessor applications exists, but are costly:
  - “Correct Rounding and a Hybrid Approach to Exact Floating-Point Summation”
  - “The Accuracy of Floating Point Summation”
  - Interval arithmetic
  - Uncertainty quantification
  - Fixed-point arithmetic
  - Higher accuracy:
    - GMP and MPFR (languages which are interoperable with C)
    - BigInt (Java, TCL, python, etc...)

[Die12]

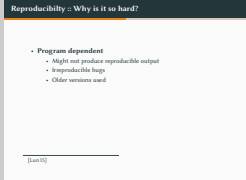


- The need an memory / processing overhead
- Have an upper- / lower- bound

Reproducibilty :: Why is it so hard?

- **Program dependent**
  - Might not produce reproducible output
  - Irreproducible bugs
  - Older versions used

[Lun15]

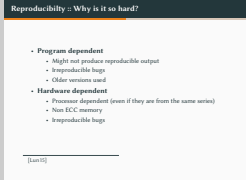


- **Program dependent**
  - E.g. LAPACK / Gnuplot / numpy have errors or aren't reproducible without knowing it
  - irreproducible bugs
- **Hardware dependent**
  - Chiplottery
  - Quantum sized effects
  - Error correction codes not available (e.g consumer hardware)

Reproducibilty :: Why is it so hard?

- **Program dependent**
  - Might not produce reproducible output
  - Irreproducible bugs
  - Older versions used
- **Hardware dependent**
  - Processor dependent (even if they are from the same series)
  - Non ECC memory
  - Irreproducible bugs

[Lun15]

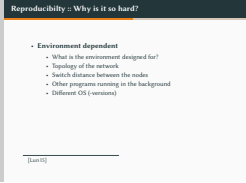


- **Program dependent**
  - E.g. LAPACK / Gnuplot / numpy have errors or aren't reproducible without knowing it
  - irreproducible bugs
- **Hardware dependent**
  - Chiplottery
  - Quantum sized effects
  - Error correction codes not available (e.g consumer hardware)

Reproducibilty :: Why is it so hard?

- **Environment dependent**
  - What is the environment designed for?
  - Topology of the network
  - Switch distance between the nodes
  - Other programs running in the background
  - Different OS (-versions)

[Lun15]

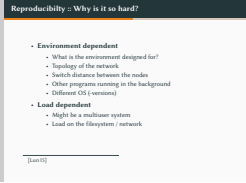


- **Environment dependent**
  - E.g. mobile vs desktop
  - Could have non stable connection (Amazon AWS / SETI@home)
- **Load dependent**
  - How many switches are between the nodes
  - Is the processor shared with something else
  - What else run on the computer

Reproducibilty :: Why is it so hard?

- **Environment dependent**
  - What is the environment designed for?
  - Topology of the network
  - Switch distance between the nodes
  - Other programs running in the background
  - Different OS (-versions)
- **Load dependent**
  - Might be a multiuser system
  - Load on the filesystem / network

[Lun15]

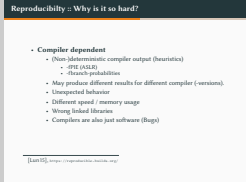


- **Environment dependent**
  - E.g. mobile vs desktop
  - Could have non stable connection (Amazon AWS / SETI@home)
- **Load dependent**
  - How many switches are between the nodes
  - Is the processor shared with something else
  - What else run on the computer

Reproducibilty :: Why is it so hard?

- **Compiler dependent**
  - (Non-)deterministic compiler output (heuristics)
    - -fPIE (ASLR)
    - -fbranch-probabilities
  - May produce different results for different compiler (-versions).
  - Unexpected behavior
  - Different speed / memory usage
  - Wrong linked libraries
  - Compilers are also just software (Bugs)

[Lun15], https://reproducible-builds.org/



- Even for the same compiler the output might not be the same
  - ASLR = address space layout randomization, randomly arrange address space positions (like stack, heap, libraries)
    `https://en.wikipedia.org/wiki/Address_space_layout_randomization`
  - -fbranch-probabilities guess the branching based on a random probability, only on old gcc versions
    `https://gcc.gnu.org/onlinedocs/gcc-3.3.1/gcc/Optimize-Options.html`
  - Compilers are software
  - Bugs in the compiler
  - Some compilers insert time stamps
  - Other machine was used to compile
  - Can result in different run-time or memory usage

Reproducibilty :: Why Reproducible Builds?

- For a deterministic build system
  - Improve debugging

[Lun15]

Reproducibilty :: Why Reproducible Builds?
- For a deterministic build system
  - Improve debugging
[Lun15]

- For debugging software
- For sharing software without rebuilding it

Reproducibilty :: Why Reproducible Builds?

- For a deterministic build system
  - Improve debugging
- Reproducible environment
  - Verification through hash
  - Rebuild software from source with the same hash
  - Deterministic output on the same hardware

[Lun15]

Reproducibilty :: Why Reproducible Builds?
- For a deterministic build system
  - Improve debugging
- Reproducible environment
  - Verification through hash
  - Rebuild software from source with the same hash
  - Deterministic output on the same hardware
[Lun15]

- For debugging software
- For sharing software without rebuilding it

Reproducibilty :: Common Problems

- Timestamps / Timezone

[Lun15]

- Timestamps for info texts / DRM or verification
- Local may be generated before compilation
- File order / paths may lead to other optimizations or code generation

Reproducibilty :: Common Problems

- Timestamps / Timezone
- Local

[Lun15]

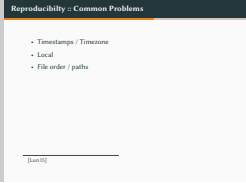
Reproducibilty :: Common Problems
- Timestamps / Timezone
- Local
[Lun15]

- Timestamps for info texts / DRM or verification
- Local may be generated before compilation
- File order / paths may lead to other optimizations or code generation

Reproducibilty :: Common Problems

- Timestamps / Timezone
- Local
- File order / paths

[Lun15]



- Timestamps for info texts / DRM or verification
- Local may be generated before compilation
- File order / paths may lead to other optimizations or code generation

Reproducibilty :: Common Problems

- Timestamps / Timezone
- Local
- File order / paths
- UUID (Universally Unique Identifier)

[Lun15]

Reproducibilty :: Common Problems
- Timestamps / Timezone
- Local
- File order / paths
- UUID (Universally Unique Identifier)
[Lun15]

- Timestamps for info texts / DRM or verification
- Local may be generated before compilation
- File order / paths may lead to other optimizations or code generation

Reproducibilty :: Common Problems

- Timestamps / Timezone
- Local
- File order / paths
- UUID (Universally Unique Identifier)
- CPU-specific

[Lun15]

Reproducibilty :: Common Problems
- Timestamps / Timezone
- Local
- File order / paths
- UUID (Universally Unique Identifier)
- CPU-specific
[Lun15]

- Timestamps for info texts / DRM or verification
- Local may be generated before compilation
- File order / paths may lead to other optimizations or code generation

Reproducibilty :: Common Problems

- Timestamps / Timezone
- Local
- File order / paths
- UUID (Universally Unique Identifier)
- CPU-specific
- Version information

[Lun15]

Reproducibilty :: Common Problems
- Timestamps / Timezone
- Local
- File order / paths
- UUID (Universally Unique Identifier)
- CPU-specific
- Version information
[Lun15]

- Timestamps for info texts / DRM or verification
- Local may be generated before compilation
- File order / paths may lead to other optimizations or code generation

Reproducibilty :: Common Problems

- Timestamps / Timezone
- Local
- File order / paths
- UUID (Universally Unique Identifier)
- CPU-specific
- Version information
- Environment variables

[Lun15]

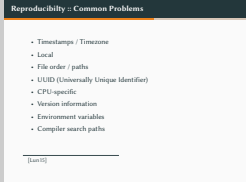
Reproducibilty :: Common Problems
- Timestamps / Timezone
- Local
- File order / paths
- UUID (Universally Unique Identifier)
- CPU-specific
- Version information
- Environment variables
[Lun15]

- Timestamps for info texts / DRM or verification
- Local may be generated before compilation
- File order / paths may lead to other optimizations or code generation

Reproducibilty :: Common Problems

- Timestamps / Timezone
- Local
- File order / paths
- UUID (Universally Unique Identifier)
- CPU-specific
- Version information
- Environment variables
- Compiler search paths

[Lun15]

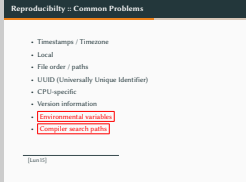


- Timestamps for info texts / DRM or verification
- Local may be generated before compilation
- File order / paths may lead to other optimizations or code generation

Reproducibilty :: Common Problems

- Timestamps / Timezone
- Local
- File order / paths
- UUID (Universally Unique Identifier)
- CPU-specific
- Version information
- Environmental variables
- Compiler search paths

[Lun15]



- Timestamps for info texts / DRM or verification
- Local may be generated before compilation
- File order / paths may lead to other optimizations or code generation

Reproducibilty :: HPC Project Approach

https://raw.githubusercontent.com/LLNL/spack/develop/share/spack/logo/spack-logo-text-64.png

Reproducibilty :: HPC Project Approach

- http://www.hpcadvisorycouncil.com/events/2017/swiss-workshop/pdf/Weds12April/MCulpo_Spack_SWInstallation_Packaging_Wed04122017.pdf
- Sets environment variables
- Searches for the compiler
- Download necessary dependencies
- gcc /ld still checks for libraries in other paths
  - Solution: Jail the environment
- Package manager for supercomputers on Linux / MacOS
- Multilanguage support (C, C++, python, R, Go, Fortran)
- Supports for different software package versions
- Custom package support

Reproducibilty :: HPC Project Approach

- Sets environment variables

Reproducibilty :: HPC Project Approach
Spack
- Sets environment variables

Reproducibilty :: HPC Project Approach

- Sets environment variables
- Searches for the compiler

Reproducibilty :: HPC Project Approach
Spack
- Sets environment variables
- Searches for the compiler

Reproducibilty :: HPC Project Approach

- Sets environment variables
- Searches for the compiler
- Download and compile necessary dependencies

Reproducibilty :: HPC Project Approach
Spack
- Sets environment variables
- Searches for the compiler
- Download and compile necessary dependencies

Reproducibilty :: HPC Project Approach

- Sets environment variables
- Searches for the compiler
- Download and compile necessary dependencies
- Patches packages

Reproducibilty :: HPC Project Approach
Spack
- Sets environment variables
- Searches for the compiler
- Download and compile necessary dependencies
- Patches packages

Reproducibilty :: HPC Project Approach

- Sets environment variables
- Searches for the compiler
- Download and compile necessary dependencies
- Patches packages
- Multilanguage support (C, C++, python, R, Go, Fortran)
- Supports for different software package versions
- Custom package support

Reproducibilty :: HPC Project Approach
Spack
- Sets environment variables
- Searches for the compiler
- Download and compile necessary dependencies
- Patches packages
- Multilanguage support (C, C++, python, R, Go, Fortran)
- Supports for different software package versions
- Custom package support

Reproducibilty :: HPC Project Approach

- spack install libelf

Reproducibilty :: HPC Project Approach
Spack
- spack install libelf

- Sets environment variables
- Searches for the compiler
- Download necessary dependencies
- gcc /ld still checks for libraries in other paths
  - Solution: Jail the environment
- Package manager for supercomputers on Linux / MacOS
- Multilanguage support (C, C++, python, R, Go, Fortran)
- Supports for different software package versions
- Custom package support

Reproducibilty :: HPC Project Approach

- spack install libelf
- spack install libelf@0.8.13 %gcc

Reproducibilty :: HPC Project Approach
Spack
- spack install libelf
- spack install libelf@0.8.13 %gcc

- Sets environment variables
- Searches for the compiler
- Download necessary dependencies
- gcc /ld still checks for libraries in other paths
  - Solution: Jail the environment
- Package manager for supercomputers on Linux / MacOS
- Multilanguage support (C, C++, python, R, Go, Fortran)
- Supports for different software package versions
- Custom package support

Reproducibilty :: HPC Project Approach

- spack install libelf
- spack install libelf@0.8.13 %gcc
- spack install libelf@0.8.12 %gcc cppflags=”-O3”

Reproducibilty :: HPC Project Approach
Spack
- spack install libelf
- spack install libelf@0.8.13 %gcc
- spack install libelf@0.8.12 %gcc cppflags=”-O3”

- Sets environment variables
- Searches for the compiler
- Download necessary dependencies
- gcc /ld still checks for libraries in other paths
  - Solution: Jail the environment
- Package manager for supercomputers on Linux / MacOS
- Multilanguage support (C, C++, python, R, Go, Fortran)
- Supports for different software package versions
- Custom package support

Reproducibilty :: HPC Project Approach

- spack install libelf
- spack install libelf@0.8.13 %gcc
- spack install libelf@0.8.12 %gcc cppflags="-O3"
- spack install hdf5@1.10.1 +szip

Reproducibilty :: HPC Project Approach
Spack
- spack install libelf
- spack install libelf@0.8.13 %gcc
- spack install libelf@0.8.12 %gcc cppflags="-O3"
- spack install hdf5@1.10.1 +szip

- Sets environment variables
- Searches for the compiler
- Download necessary dependencies
- gcc /ld still checks for libraries in other paths
  - Solution: Jail the environment
- Package manager for supercomputers on Linux / MacOS
- Multilanguage support (C, C++, python, R, Go, Fortran)
- Supports for different software package versions
- Custom package support

Reproducibilty :: HPC Project Approach

- A new version of a package doesn't break the existing one
  - Every package get its own prefix and hash
  - Prevent using wrong libraries / programs

Reproducibilty :: HPC Project Approach
Spack
- A new version of a package doesn't break the existing one
  - Every package get its own prefix and hash
  - Prevent using wrong libraries / programs

- Sets environment variables
- Searches for the compiler
- Download necessary dependencies
- gcc /ld still checks for libraries in other paths
  - Solution: Jail the environment
- Package manager for supercomputers on Linux / MacOS
- Multilanguage support (C, C++, python, R, Go, Fortran)
- Supports for different software package versions
- Custom package support

Reproducibilty :: HPC Project Approach

- A new version of a package doesn't break the existing one
  - Every package get its own prefix and hash
  - Prevent using wrong libraries / programs
- spack install libelf ⇒ c5ru3jw

Reproducibilty :: HPC Project Approach
Spack
- A new version of a package doesn't break the existing one
  - Every package get its own prefix and hash
  - Prevent using wrong libraries / programs
- spack install libelf ⇒ c5ru3jw

- Sets environment variables
- Searches for the compiler
- Download necessary dependencies
- gcc /ld still checks for libraries in other paths
  - Solution: Jail the environment
- Package manager for supercomputers on Linux / MacOS
- Multilanguage support (C, C++, python, R, Go, Fortran)
- Supports for different software package versions
- Custom package support

Reproducibilty :: HPC Project Approach

- A new version of a package doesn't break the existing one
  - Every package get its own prefix and hash
  - Prevent using wrong libraries / programs
- spack install libelf ⇒ c5ru3jw
- spack install libelf@0.8.13 %gcc ⇒ axygp5o

Reproducibilty :: HPC Project Approach
Spack
- A new version of a package doesn't break the existing one
  - Every package get its own prefix and hash
  - Prevent using wrong libraries / programs
- spack install libelf ⇒ c5ru3jw
- spack install libelf@0.8.13 %gcc ⇒ axygp5o

- Sets environment variables
- Searches for the compiler
- Download necessary dependencies
- gcc /ld still checks for libraries in other paths
  - Solution: Jail the environment
- Package manager for supercomputers on Linux / MacOS
- Multilanguage support (C, C++, python, R, Go, Fortran)
- Supports for different software package versions
- Custom package support

Reproducibilty :: HPC Project Approach

- A new version of a package doesn't break the existing one
  - Every package get its own prefix and hash
  - Prevent using wrong libraries / programs

- spack install libelf ⇒ c5ru3jw
- spack install libelf@0.8.13 %gcc ⇒ axygp5o
- spack install libelf@0.8.12 %gcc cppflags="-O3" ⇒ mmbjylc

Reproducibilty :: HPC Project Approach
Spack
- A new version of a package doesn't break the existing one
  - Every package get its own prefix and hash
  - Prevent using wrong libraries / programs
- spack install libelf ⇒ c5ru3jw
- spack install libelf@0.8.13 %gcc ⇒ axygp5o
- spack install libelf@0.8.12 %gcc cppflags="-O3" ⇒ mmbjylc

- Sets environment variables
- Searches for the compiler
- Download necessary dependencies
- gcc /ld still checks for libraries in other paths
  - Solution: Jail the environment
- Package manager for supercomputers on Linux / MacOS
- Multilanguage support (C, C++, python, R, Go, Fortran)
- Supports for different software package versions
- Custom package support

Best Practices

Best Practices

Reproducibility :: Best Practices

- What should be included ([Sto14] based on [Cec03]):

Reproducibility :: Best Practices
- What should be included ([Sto14] based on [Cec03]):

- Definition by Stodden et al. based on the definition for the Committee on Responsibilities of Authorship in the Biological Sciences
- Data:
  - If you got data from a different source or measured by your self it might be different and can include measuring errors
  - This also means that the author of the original paper needs to describe how the data was measured, to ensure reproducibility
- Algorithms:
  - Exact description of the algorithms are necessary to know what the algorithm should return for specific inputs
  - E.g: Stable vs. unstable sorting
- Others:
  - Which instruments where used and how where they calibrated
  - All parameters of a program should be listed.

Reproducibility :: Best Practices

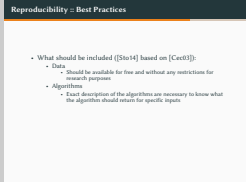
- What should be included ([Sto14] based on [Cec03]):
  - Data
    - Should be available for free and without any restrictions for research purposes

Reproducibility :: Best Practices
- What should be included ([Sto14] based on [Cec03]):
  - Data
    - Should be available for free and without any restrictions for research purposes

- Definition by Stodden et al. based on the definition for the Committee on Responsibilities of Authorship in the Biological Sciences
- Data:
  - If you got data from a different source or measured by your self it might be different and can include measuring errors
  - This also means that the author of the original paper needs to describe how the data was measured, to ensure reproducibility
- Algorithms:
  - Exact description of the algorithms are necessary to know what the algorithm should return for specific inputs
  - E.g: Stable vs. unstable sorting
- Others:
  - Which instruments where used and how where they calibrated
  - All parameters of a program should be listed.

Reproducibility :: Best Practices

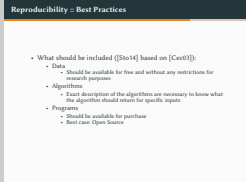
- What should be included ([Sto14] based on [Cec03]):
  - Data
    - Should be available for free and without any restrictions for research purposes
  - Algorithms
    - Exact description of the algorithms are necessary to know what the algorithm should return for specific inputs



- Definition by Stodden et al. based on the definition for the Committee on Responsibilities of Authorship in the Biological Sciences
- Data:
  - If you got data from a different source or measured by your self it might be different and can include measuring errors
  - This also means that the author of the original paper needs to describe how the data was measured, to ensure reproducibility
- Algorithms:
  - Exact description of the algorithms are necessary to know what the algorithm should return for specific inputs
  - E.g: Stable vs. unstable sorting
- Others:
  - Which instruments where used and how where they calibrated
  - All parameters of a program should be listed.

Reproducibility :: Best Practices

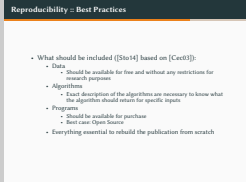
- What should be included ([Sto14] based on [Cec03]):
  - Data
    - Should be available for free and without any restrictions for research purposes
  - Algorithms
    - Exact description of the algorithms are necessary to know what the algorithm should return for specific inputs
  - Programs
    - Should be available for purchase
    - Best case: Open Source



- Definition by Stodden et al. based on the definition for the Committee on Responsibilities of Authorship in the Biological Sciences
- Data:
  - If you got data from a different source or measured by your self it might be different and can include measuring errors
  - This also means that the author of the original paper needs to describe how the data was measured, to ensure reproducibility
- Algorithms:
  - Exact description of the algorithms are necessary to know what the algorithm should return for specific inputs
  - E.g: Stable vs. unstable sorting
- Others:
  - Which instruments where used and how where they calibrated
  - All parameters of a program should be listed.

Reproducibility :: Best Practices

- What should be included ([Sto14] based on [Cec03]):
  - Data
    - Should be available for free and without any restrictions for research purposes
  - Algorithms
    - Exact description of the algorithms are necessary to know what the algorithm should return for specific inputs
  - Programs
    - Should be available for purchase
    - Best case: Open Source
  - Everything essential to rebuild the publication from scratch

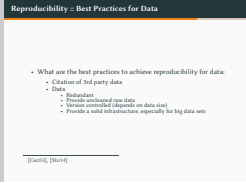


- Definition by Stodden et al. based on the definition for the Committee on Responsibilities of Authorship in the Biological Sciences
- Data:
  - If you got data from a different source or measured by your self it might be different and can include measuring errors
  - This also means that the author of the original paper needs to describe how the data was measured, to ensure reproducibility
- Algorithms:
  - Exact description of the algorithms are necessary to know what the algorithm should return for specific inputs
  - E.g: Stable vs. unstable sorting
- Others:
  - Which instruments where used and how where they calibrated
  - All parameters of a program should be listed.

Reproducibility :: Best Practices for Data

- What are the best practices to achieve reproducibility for data:
  - Citation of 3rd party data
  - Data
    - Redundant
    - Provide uncleaned raw data
    - Version controlled (depends on data size)
    - Provide a solid infrastructure, especially for big data sets

[Cec03], [Sto14]

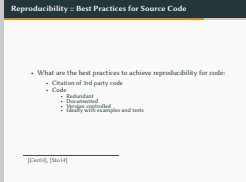


- Definition by Stodden et al.
- Data should be accessible to future researchers
- Uncleaned data should be used to avoid replicability
- Version controlled to avoid cheating but also to keep it apart from similar datasets.
- Should be freely available!

Reproducibility :: Best Practices for Source Code

- What are the best practices to achieve reproducibility for code:
  - Citation of 3rd party code
  - Code
    - Redundant
    - Documented
    - Version controlled
    - Ideally with examples and tests

[Cec03], [Sto14]

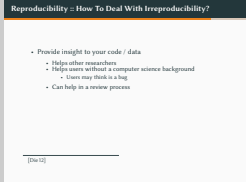


- Definition by Stodden et al.
- Data should be accessible to future researchers
- Documented (not in [Sto14]) To make sure that the program code does what it is supposed to and to help others to understand the code.
- Version controlled help seeing the progress of the researchers. Allow simple modification with the benefit to republish the code for others.
- Best case: Open Source!

Reproducibility :: How To Deal With Irreproducibility?

- Provide insight to your code / data
  - Helps other researchers
  - Helps users without a computer science background
    - Users may think is a bug
  - Can help in a review process

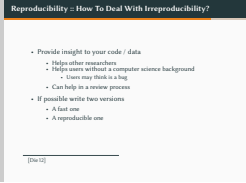
[Die12]



Reproducibility :: How To Deal With Irreproducibility?

- Provide insight to your code / data
  - Helps other researchers
  - Helps users without a computer science background
    - Users may think is a bug
  - Can help in a review process
- If possible write two versions
  - A fast one
  - A reproducible one

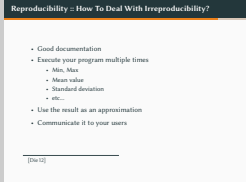
[Die12]



Reproducibility :: How To Deal With Irreproducibility?

- Good documentation
- Execute your program multiple times
  - Min, Max
  - Mean value
  - Standard deviation
  - etc...
- Use the result as an approximation
- Communicate it to your users

[Die12]



Reproducibility :: Summary

- Reproducibility $\neq$ Replicability
- Software introduces a layer of uncertainty
- Not always possible to achieve reproducibility
- Open data / source help others
- Reproducibility may be costly

Reproducibility :: Summary
- Reproducibility ≠ Replicability
- Software introduces a layer of uncertainty
- Not always possible to achieve reproducibility
- Open data / source help others
- Reproducibility may be costly

Literatur

C. Glenn Begley und Lee M. Ellis. “Drug development: Raise standards for preclinical cancer research”. eng. In: *Nature* 483 (7391 2012). Journal Article, S. 531–533. ISSN: 0028-0836. DOI: \url{10.1038/483531a}. eprint: 22460880 (siehe S. 5–7).

Thomas R. Cech. *Sharing Publication-Related Data and Materials*. Hrsg. von Sciences, National Research Council Committee on Responsibilities of Authorship in the Biological. 2003. DOI: \url{10.17226/10613} (siehe S. 58–64).

Kai Diethelm. “The Limits of Reproducibility in Numerical Simulation”. In: *Computing in Science & Engineering* 14 (1 2012), S. 64–72. ISSN: 1521-9615. DOI: \url{10.1109/MCSE.2011.21}. (Besucht am 07. 06. 2017) (siehe S. 24–26, 65–67).

Chris Drummond. *Replicability is not Reproducibility: Nor is it Good Science.* 2009. URL: \url{http://www.csi.uottawa.ca/åLijcdrummon/pubs/ICMLws09.pdf} (besucht am 22. 05. 2017) (siehe S. 10).

DWD. *ICON (Icosahedral Nonhydrostatic) Model.* Hrsg. von Deutscher Wetterdienst. o.J. URL: \url{https://www.dwd.de/EN/research/weatherforecasting/num_modelling/01_num_weather_prediction_modells/icon_description.html} (besucht am 18. 06. 2017) (siehe S. 19).

Frank Helmut. *COSMO-EU: Beendigung des operationellen Betriebs des Regionalmodells COSMO-EU.* 2016. URL: \url{http://www.dwd.de/DE/fachnutzer/forschung_lehre/numerische_wettervorhersage/nwv_aenderungen/_functions/DownloadBox_modellaenderungen/cosmo_eu/pdf_2016_2019/pdf_cosmo_eu_01_12_2016.pdf?__blob=publicationFile&v=2} (besucht am 18. 06. 2017) (siehe S. 19).

Lunar. *How to make your software build reproducibly. Provide a verifiable path from source to binary.* 2015. URL: \url{https://reproducible.alioth.debian.org/presentations/2015-08-13-CCCamp15-outline.pdf} (besucht am 09. 06. 2017) (siehe S. 27–42).

D. Reinert, G. Zängl, Prill F., A. Fernandez del Rio, R. Potthast, D. Rieger, Schröter, J., Förstner, J., C. Walter, R. Ruhnke und Vogel B. *Working with the ICON Model. Practical Exercises for NWP Model and ICON-ART*. 2015. URL: \url{https://www.earthsystemcog.org/site_media/projects/dcmip-2016/ICON_tutorial.pdf} (besucht am 18. 06. 2017) (siehe S. 23).

Victoria Stodden und Sheila Miguez. "Best Practices for Computational Science: Software Infrastructure and Environments for Reproducible and Extensible Research". en. In: *Journal of Open Research Software* 2 (1 2014). vcs@stodden.net Columbia University shekay@gmail.com. ISSN: 2049-9647. DOI: \url{10.5334/jors.ay}. URL: \url{/han/sfx/openresearchsoftware.metajnl.com/articles/10.5334/jors.ay/galley/64/download/} (siehe S. 58–64).