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
Why do we need Reproducibility
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 reproducible ([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?

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 reproducible ([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?

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 reproducible ([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?
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, last visit: 25.05.2017
Reproducibility ≠ Replicability

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

[Dru09]
Reproducibility in Empirical Science by Example
Focus on gathering data

Events aren’t often deterministic

High probability of measuring random data

Empirical Sciences depends often on statistical methods.

\[ \rightarrow \text{No exact solution possible} \]

Sometimes no second setup exist (e.g. Hubble)

ATLAS / CMS:

- Higgs detection
- Huge range of particle detectors
- Expensive to run 1.3TWh/a \(\approx\) 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 \(\approx\) 1.4MW/a \(\approx\) 12.3GWh/a (Mistral))
• Events are happening every 25ns or 40 million times per second

• Prefiltering of events necessary to store the data onto disk problematic for reproducibility
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

$\sigma \triangleq 10^{-10}\%$ of a random measured result

Through measuring the significance

Small mistakes can make such an experiment irreproducible like Faster than Light Neutrinos

- 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

7σ ≈ 10⁻¹⁰% of a random measured result

Through measuring the significance

Small mistakes can make such a experiment irreproducible like Faster than Light Neutrinos

- 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

7σ ≈ 10^{-10}\% of a random measured result

Through measuring the significance

Small mistakes can make such an experiment irreproducible like Faster than Light Neutrinos

- 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 :: Simulations

ICON-Model

- 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

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

• 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
• 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])
Reproducibility :: 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
    \[ a \cdot (b + c) \neq a \cdot b + a \cdot c \]
Reproducibility :: Why is it so hard?

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

\[
\begin{align*}
10^{12} \text{ Core 1} & + 10^{-8} \text{ Core 3} & = 10^{12} \\
-10^{12} \text{ Core 2} & + 10^{-8} \text{ Core 3} & = 10^{-8} \\
10^{-8} \text{ Core 3} & + 10^{-8} \text{ Core 4} & = 2 \cdot 10^{-8}
\end{align*}
\]

[Die12]
Reproducibility :: 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]

Reproducibility

• The need an memory / processing overhead
• Have an upper- / lower- bound
Reproducibility :: Why is it so hard?

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

- Hardware dependent
  - Chiplettery
  - Quantum sized effects
  - Error correction codes not available (e.g. consumer hardware)

[Lun15]
Reproducibility :: 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]
Reproducibility :: 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
  - 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

[Lun15]
Reproducibility :: 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]
Reproducibility :: 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
- 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
Reproducibility :: Why Reproducible Builds?

- For a deterministic build system
  - Improve debugging

- For debugging software
- For sharing software without rebuilding it

[Lun15]
Reproducibility: 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]
Reproducibility :: Common Problems

- Timestamps / Timezone

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

[Lun15]
Reproducibility :: Common Problems

- Timestamps / Timezone
- Local

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

[Lun15]
Reproducibility :: Common Problems

- Timestamps / Timezone
- Local
- File order / paths

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

• Timestamps for info texts / DRM or verification
• Local may be generated before compilation
• File order / paths may lead to other optimizations or code generation
Reproducibility :: Common Problems

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

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

[Lun15]
Reproducibility :: Common Problems

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

[2017-07-03]

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

[Lun15]
Reproducibility :: 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
Reproducibility :: HPC Project Approach

- 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

Reproducibility :: HPC Project Approach

- Sets environment variables

Reproducibility in Simulations

Supports for different software package versions

Custom package support
- Sets environment variables
- Searches for the compiler
- Sets environment variables
- Searches for the compiler
- Download and compile necessary dependencies
Reproducibility :: HPC Project Approach

- Sets environment variables
- Searches for the compiler
- Download and compile necessary dependencies
- Patches packages
Reproducibility :: 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
• 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
• 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
- spack install libelf
- spack install libelf@0.8.13 %gcc
- spack install libelf@0.8.12 %gcc cppflags="-O3"
• 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
Reproducibility :: 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

• 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
Reproducibility :: HPC Project Approach

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

- `spack install libelf` ⇒ c5ru3jw

Reproducibility

• 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
Reproducibility :: 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 ⇒ axyg5po

- 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
Reproducibility :: 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 ⇒ axyg5p0
• 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
• 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
  - E.g.: Stable vs. unstable sorting

• Programs
  - Should be available for purchase
  - Best case: Open Source

• Everything essential to rebuild the publication from scratch
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
• 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.
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
• 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 yourself 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 were used and how they were calibrated
  - All parameters of a program should be listed.
• 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!
• 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

• 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!

[Cec03], [Sto14]
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 ≠ Replicability
Software introduces a layer of uncertainty
Not always possible to achieve reproducibility
Open data / source help others
Reproducibility may be costly
Literatur


