

Software Management for Open Science

Horea Christian

SSC TheAlternative | ETHZ and UZH

November 12, 2019

These Slides

Type one link, click all others:

- ▶ Download `thealternative.ch/ssm/slides.pdf`

SSH

Linux and MacOS:

- ▶ Check that you can run:

```
ssh YOURUSER@130.60.24.66
```

Windows:

- ▶ Download and launch “Git for Windows” from git-for-windows.github.io.
- ▶ Check that you can run:

```
ssh YOURUSER@130.60.24.66
```

Command Line Text Editor

Usable via SSH and ubiquitous. There are many alternatives, but here we use `nano`:

- ▶ Open file:

```
nano file
```

- ▶ Save via: `Ctrl` + `o`, `Enter`
- ▶ Exit via: `Ctrl` + `x`

Git and Social Coding

Git needs to know who you are.

- ▶ On the server, run:

```
git config --global user.name "Your Name"  
git config --global user.email yourname@example.com
```

GitHub is a **social coding platform** providing free accounts:

- ▶ Register under `github.com`.
- ▶ Use a password which you can remember.

The Package

Better organization for your research!

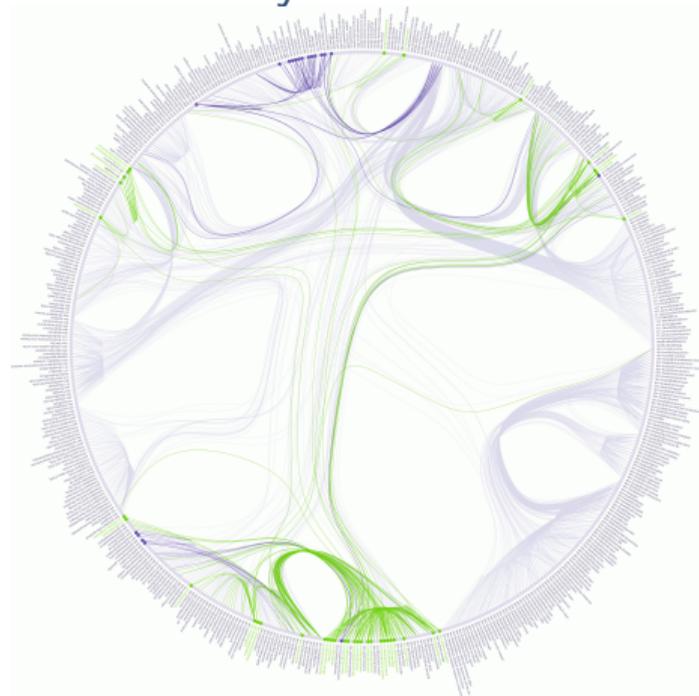
A package is a software format which is (easily):

- ▶ Distributable
- ▶ Integrated
- ▶ Testable
- ▶ Updateable
- ▶ Uninstallable
- ▶ Understandable

Package Management — best done automatically

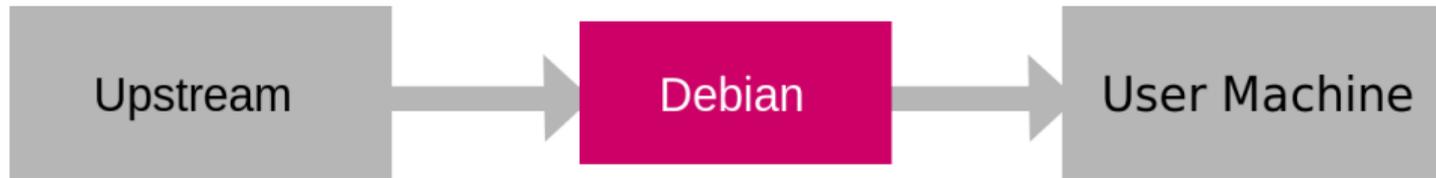
Packages interact in complex and non-trivial manners:

- ▶ Version-dependent behaviour
- ▶ Optional features
- ▶ Incompatibilities
- ▶ Static/dynamic linking



Minimal neuroscience package dependency stack [1]

Binary Packages



Rudimentary overview of binary package distribution.

Advantages:

- ▶ Faster installation
- ▶ Less variable installation

Disadvantages:

- ▶ No access to live software
- ▶ Man-in-the middle
- ▶ Limited support for rolling release

Source-Based Packages



Rudimentary overview of source-based package distribution.

Advantages:

- ▶ Live software is a first-class citizen
- ▶ Thin wrapper for upstream
- ▶ Acutely version and linking aware

Disadvantages:

- ▶ Slower installation
- ▶ More variable installation

Quality

- ▶ Make development more transparent.
- ▶ Get **constructive** feedback.
- ▶ Ask for help with concrete reproducible examples.
- ▶ Easily manage `bugs/issues` and `contributions`.
- ▶ Implement proper version tracking.

Impact

- ▶ Reach more potential users.
- ▶ Communicate with users to improve your software's usability.
- ▶ Retain more users.

Recognition

- ▶ Establish proof of authorship.
- ▶ Publicize your innovative workflows, solutions, data structures.
- ▶ Create a handle for attribution (including DOI), e.g:
 - ▶ BehavioPy: [10.5281/zenodo.188169](https://doi.org/10.5281/zenodo.188169)
 - ▶ Nipype: [10.5281/zenodo.50186](https://doi.org/10.5281/zenodo.50186)

Sustainability

A sustainable project **cannot** depend on environments remaining unchanged.

- ▶ Ensure long-term viability of your software.
- ▶ Avoid death-by-PhD.
- ▶ Give your funders their money's worth.
- ▶ Develop a lean start-up.
- ▶ Maintain a reliable and affordable infrastructure for your work.

Why Not?

- ▶ Don't be afraid of your software not being “good/unique enough”!
- ▶ Don't wait until your software is “ready”!
- ▶ A lot of research software you are already using is not written by “professional” programmers.

How do I package my software?

Choose Appropriate Technologies



Gentoo Linux Logo by Gentoo Foundation and Lennart Andre Rolland - CC BY-SA/2.5.



Python Logo by Python Software Foundation.

Python Package Distribution

You can package your python software by writing **one short** file.

- ▶ Python provides its own limited package management, e.g. via `setuptools`.
- ▶ Package metadata saved in `setup.py`, e.g. `SAMRI/setup.py`.

```
from setuptools import setup, find_packages

packages = find_packages(exclude=('samri.tests*', 'samri.*.tests*'))

setup(
    name="SAMRI",
    version="9999",
    description = "Small animal magnetic resonance imaging via Python.",
    author = "Horea Christian",
    author_email = "chr@chymera.eu",
    url = "https://github.com/IBT-FMI/SAMRI",
    keywords = ["fMRI", "pipelines", "data analysis", "bruker"],
    classifiers = [],
    install_requires = [],
    provides = ["samri"],
    packages = packages,
    include_package_data=True,
    extras_require = {
    },
    entry_points = {'console_scripts' : \
        ['SAMRI = samri.cli:main']}
    },
)
```

Gentoo Packages

A Gentoo package is **one short** file.

- ▶ Regardless of the programming language
- ▶ Can automatically interpret information contained in the package, e.g. in `setup.py`

```
# Copyright 1999-2019 Gentoo Authors
# Distributed under the terms of the GNU General Public License v2

EAPI=7

PYTHON_COMPAT=( python{3,5,3,6} )

inherit distutils-r1

DESCRIPTION="Small Animal Magnetic Resonance Imaging"
HOMEPAGE="https://github.com/IBT-FMI/SAMRI"
SRC_URI="https://github.com/IBT-FMI/SAMRI/archive/${PV}.tar.gz -> ${P}.tar.gz"

LICENSE="GPL-3"
SLOT="0"
IUSE="test"
KEYWORDS="-amd64 -x86"

DEPEND="
    test? (
        dev-python/pytest[${PYTHON_USEDEP}]
        sci-biology/samri_bidsdata
        sci-biology/samri_bindata
    )
"
RDEPEND="
    dev-python/argh[${PYTHON_USEDEP}]
    dev-python/joblib[${PYTHON_USEDEP}]
    >=dev-python/matplotlib-2.0.2[${PYTHON_USEDEP}]
    >=dev-python/numpy-1.13.3[${PYTHON_USEDEP}]
    dev-python/pandas[${PYTHON_USEDEP}]
    dev-python/seaborn[${PYTHON_USEDEP}]
    dev-python/statsmodels[${PYTHON_USEDEP}]
    media-gfx/blender
    >=sci-biology/fsl-5.0.9
    sci-biology/brun2nii
"

S="${WORKDIR}/SAMRI-${PV}"
```

Reposit Your Software



Git Logo by Jason Long (CC-BY-3.0)

You can self-host, but hosting also available via social coding platforms:

▶ GitLab

▶ GitHub

▶ Bitbucket

Put what you have learned into practice, and start typing...

A Few Basic Gentoo Commands

- ▶ Check available package names, versions, and details.

```
eix -v nibabel
```

- ▶ See package dependencies.

```
equery g nibabel
```

- ▶ See what packages depend on a said package.

```
equery d nibabel
```

- ▶ See files installed by package.

```
equery f nibabel
```

- ▶ Try to install a new package.

```
emerge -p psychopy
```

Put what you have learned into practice, and start typing...

Reproduce a Scientific Article

Novel frameworks, such as RepSeP [2] permit articles to be written as software.

- ▶ Get the source code for brand-new articles:

- ▶ Work-in-progress (reexecution time ≈ 2 min)

```
git clone https://gitlab.com/Chymera/nvcz.git
```

- ▶ Preprint (reexecution time ≈ 11 min)

```
git clone https://bitbucket.org/TheChymera/irsabi.git
```

- ▶ Switch to article directory.

```
cd nvcz
```

- ▶ Attempt to reexecute.

```
./compile.sh
```

Put what you have learned into practice, and start typing...

What happened? Dependency requirements happened.

But you can solve the issue yourself!

Write a new package atom for the package manager.

- ▶ Gentoo Linux makes this wholly autonomous.
- ▶ Solve one problem only once:
 - ▶ Installation will be automatic on all your further systems.
 - ▶ And on everybody else's systems!

Put what you have learned into practice, and start typing...

Write a Package Atom — The Overlay

- ▶ Fork an overlay on GitHub, e.g. from `github.com/TheChymera/overlay`



- ▶ Go back to your home directory.

```
cd
```

- ▶ Clone your fork of the overlay.

```
git clone https://github.com/YourName/overlay.git
```

- ▶ Make the ebuild directory, and navigate into it.

```
mkdir -p overlay/sci-biology/samri && cd $_
```

Put what you have learned into practice, and start typing...

Transparency means less work for you!

You could write the following files from scratch, but you can also reuse analogous files from existing packages.

- ▶ Copy a metadata file from a Python package.

```
cp /usr/portage/dev-python/astropy/metadata.xml .
```

- ▶ Copy an ebuild file from a Python package.

```
cp /usr/portage/dev-python/astropy/*2.0.1.ebuild samri-0.4.ebuild
```

Put what you have learned into practice, and start typing...

Write a Package Atom — The Metadata File

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE pkgmetadata SYSTEM "http://www.gentoo.org/dtd/metadata.dtd">
<pkgmetadata>
  <maintainer type="person">
    <email>chr@chymera.eu</email>
    <name>Horea Christian</name>
  </maintainer>
  <longdescription lang="en">
    SAMRI (Small Animal Magnetic Resonance Imaging) - pronounced "Sam-rye" - provides
    fMRI preprocessing, metadata parsing, and data analysis functions and workflows.
    SAMRI integrates functionalities from a number of other packages to create
    higher-level tools. The resulting interfaces aim to maximize reproducibility,
    simplify batch processing, and minimize the number of function calls required to
    generate figures and statistical summaries from the raw data.
  </longdescription>
  <upstream>
    <remote-id type="github">IBT-FMI/SAMRI</remote-id>
  </upstream>
</pkgmetadata>
```

Put what you have learned into practice, and start typing...

Write a Package Atom — The Ebuild (header excerpt)

```
# Copyright 1999-2019 Gentoo Authors
# Distributed under the terms of the GNU General Public License v2

EAPI=7

PYTHON_COMPAT=( python{3_5,3_6} )

inherit distutils-r1

DESCRIPTION="Small Animal Magnetic Resonance Imaging"
HOMEPAGE="https://github.com/IBT-FMI/SAMRI"
SRC_URI="https://github.com/IBT-FMI/SAMRI/archive/${PV}.tar.gz -> ${P}.tar.gz"

LICENSE="GPL-3"
SLOT="0"
IUSE="test"
KEYWORDS="~amd64 ~x86"
```

Put what you have learned into practice, and start typing...

Write a Package Atom — The Ebuild (dependency excerpts)

- ▶ Compile-time dependency example:

```
DEPEND="
  test? (
    dev-python/pytest[${PYTHON_USEDEP}]
    sci-biology/samri_bidsdata
    sci-biology/samri_bindata
  )
"
```

- ▶ Run-time dependency DIY (fill out, consulting github.com/IBT-FMI/SAMRI):

```
RDEPEND="
  dev-python/argh[${PYTHON_USEDEP}]
  dev-python/joblib[${PYTHON_USEDEP}]
  >=dev-python/matplotlib-2.0.2[${PYTHON_USEDEP}]
"
```

Put what you have learned into practice, and start typing...

Write a Package Atom — Finishing Touches

- ▶ Not all packages are perfect. Append the following to the ebuild:

```
S="${WORKDIR}/SAMRI-${PV}"
```

- ▶ Check your work. Minor formatting differences (e.g. indents) are not critical.

```
wget https://thealternative.ch/ssm/samri/samri-0.4.ebuild -P ~  
colordiff ~/samri-0.4.ebuild samri-0.4.ebuild  
wget https://thealternative.ch/ssm/samri/metadata.xml -P ~  
colordiff ~/metadata.xml metadata.xml
```

Put what you have learned into practice, and start typing...

Social Coding — Upload Your Package for Reuse

- ▶ Download the data and make git aware of your files.

```
ebuild samri-0.4.ebuild manifest && git add .
```

- ▶ Run a quality check.

```
repoman full
```

- ▶ Record and publish your work in version control.

```
git commit -a && git push origin master
```

- ▶ Include your work in widely used overlay: visit github.com/YourName/overlay.

Branch: master ▾

New pull request

Create new file

Upload files

Find file

Clone or download ▾

This branch is 1 commit ahead of TheChymera:master.

 Pull request

 Compare

Put what you have learned into practice, and start typing...

Use Your Work

- ▶ Update the package index (as superuser).

```
eix-sync
```

- ▶ Try out the install command yourself.

```
emerge -pv samri
```

- ▶ Install (as superuser).

```
emerge -v samri
```

Put what you have learned into practice, and start typing...

The Article Environment is Now Reproducible

- ▶ Navigate back to the article directory.

```
cd ~/nvcz
```

- ▶ Compile.

```
./compile.sh
```

- ▶ Log out from SSH: `Ctrl` + `d`

- ▶ Get the document locally.

```
scp YOURUSER@130.60.24.66:nvcz/article.pdf .
```

Put what you have learned into practice, and start typing...

And the Article is now Automated

- ▶ Log back in and navigate to article directory.

```
ssh YOURUSER@130.60.24.66  
cd nvcz
```

- ▶ Automatically adjust the t-statistic threshold for the entire document.

```
grep -rI 3\.5 | xargs sed -i -e "s/3.5/3.0/g"
```

- ▶ Clean up trace files and visualize what you have changed.

```
./cleanup.sh && git diff
```

- ▶ Compile, log out.
- ▶ Get the document locally.

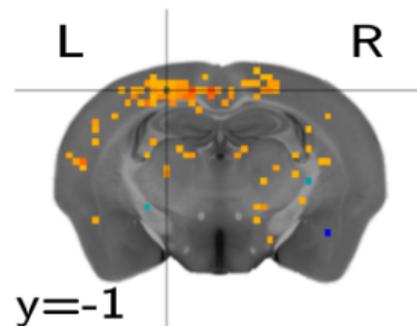
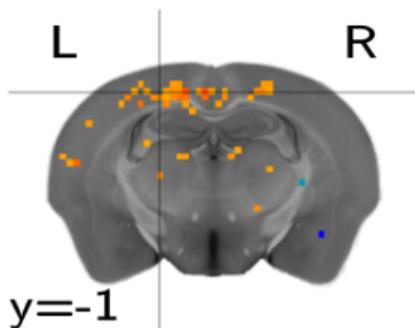
```
scp YOURUSER@130.60.24.66:nvcz/article.pdf newarticle.pdf
```

Put what you have learned into practice, and start typing...

Results

You have:

- ▶ Packaged a new piece of scientific software, now automatically installable:
 - ▶ by anybody else,
 - ▶ by you on any machine.
- ▶ Updated data analysis visualizations in a reproducible article.
 - ▶ It's that easy to contribute to well-organized research!



What now?

- ▶ Q&A round
in a few seconds
- ▶ Get help packaging your own Free and Open Source Scientific Software
in a few minutes
- ▶ Get help with running your own Gentoo Linux data analysis server
in a few hours
- ▶ Spread package management in your field
tomorrow at work

These Slides

- ▶ Latest Slides:
`thealternative.ch/ssm/slides.pdf`
- ▶ Source:
`gitlab.ethz.ch/thealternative/courses/tree/master/scientific_software_management`
- ▶ License: CC BY-SA 3.0

References

- [1] H.-I. Ioanas, B. Saab, and M. Rudin, “Gentoo linux for neuroscience - a replicable, flexible, scalable, rolling-release environment that provides direct access to development software,” *Research Ideas and Outcomes*, vol. 3, p. e12095, 2017. [Online]. Available: <https://doi.org/10.3897/rio.3.e12095>
- [2] H.-I. Ioanas and M. Rudin, “Reproducible self-publishing for Python-based research.” *EuroSciPy*, Aug. 2018. [Online]. Available: https://figshare.com/articles/Reproducible_Self-Publishing_for_Python-Based_Research/7247339