

A complete reproducible ROOT environment in under 5 minutes

Henry Schreiner^{1,2} Chris Burr³ Enrico Guiraud⁴

March 21, 2019

University of Cincinnati¹

Princeton University²

University of Manchester³

CERN⁴

CONDA



Getting Conda

Download from anaconda.com/distribution, [your favorite package manager](#), or:

Download the Linux installer

```
wget -nv http://repo.continuum.io/miniconda/Miniconda3-latest-Linux-x86_64.sh -O miniconda.sh
```

Or download the macOS installer

```
wget -nv https://repo.anaconda.com/miniconda/Miniconda3-latest-MacOSX-x86_64.sh -O miniconda.sh
```

Install Conda (same for macOS and Linux)

```
bash miniconda.sh -b -p $HOME/miniconda
```

Activate, add to bashrc - similar files available for fish and csh

```
source $HOME/miniconda/etc/profile.d/conda.sh
```

No sudo required ■ Works in CI ■ Docker available

Install ROOT anywhere

Install like any conda-forge package:

New environment:

```
conda create -n myrootenv python=3.7 root -c conda-forge  
conda activate myrootenv  
conda config --env --add channels conda-forge
```

Now use `conda activate myrootenv` and `conda deactivate`.

Inside an environment:

```
conda config --env --add channels conda-forge  
conda install root
```

Please use very recent Conda for best results

Features

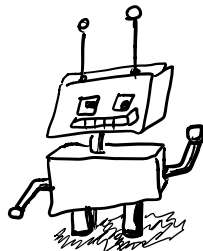
If you've heard of the NLeSC ROOT package, this is the successor:

- Full featured ROOT (not just PyROOT)
 - ▶ Compile your own code with C++17
 - ▶ Full graphics
 - ▶ JupyROOT
 - ▶ No environment setup needed
- Full support for Linux and macOS
 - ▶ Testing in ROOT nightlies planned
 - ▶ Tested on many Linux distributions
- `root -l -q $ROOTSYS/tutorials/dataframe/df013_InspectAnalysis.C`
- `root -b -l -q -e 'std::cout << (float) TPython::Eval("1+1") << endl;'`
- Supports Python 3.7, 3.6, and 2.7
- Uses external packages where possible
- Many optional features activated
 - ▶ Minuit2
 - ▶ RooFit
 - ▶ Pythia8
 - ▶ And more!

Note: macOS compiling requires 10.9 SDK for now - restriction may be lifted in the future.

Building

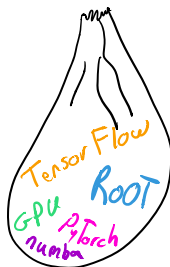
- Conda-forge automation: CI builds and bots help with updates
- Using new uniform compiler stack
 - ▶ ROOT became available on day 1 of the new compilers
- Recipe and patches are at [/conda-forge/root-feedstock](https://github.com/conda-forge/root-feedstock)



Development

- Many conda-forge formulas improved
 - ▶ Chris Burr is now part of staged-recipes
- Quite a few patches accepted into ROOT 6.16.00, 6.16.02, and 6.18.00
- User feedback in hours, fixes in 1-2 days

Usage examples: Reproducible and shareable environments



- Can mix ROOT with TensorFlow or PyTorch on GPU
- Can run multiple Python versions and ROOT

Run with:

```
conda env create -f environment.yml
```

```
name: pvfindergpu
```

```
channels:
```

- pytorch
- conda-forge

```
dependencies:
```

- python>=3.6
- root
- pythia8
- pandas
- jupyterlab
- uproot
- h5py
- numba
- scipy
- matplotlib
- pytorch
- cudatoolkit=9.0

Reproducible and shareable environments: continued

```
conda env export > environment.yml
```

```
name: pvfindergpu
```

```
channels:
```

- pytorch
- conda-forge
- defaults

```
dependencies:
```

- afterimage=1.21=hf014aae_1002
- asn1crypto=0.24.0=py37_1003
- ...
- root=6.16.00=py37_blas_openblash629dc00_5
- scipy=1.2.1=py37_blas_openblash1522bff_0
- ...
- zlib=1.2.11=h14c3975_1004

```
prefix: /home/schreihf/.conda/envs/pvfindergpu
```

Usage examples: MyBinder

Aghast: Demo of histogram with ROOT comparison.



```
environment.yml
```

```
channels:
```

- conda-forge

```
dependencies:
```

- root
- pandas
- numpy
- # for flatc code generator*
- flatbuffers>=1.8.0
- pip:
 - # for Python flatbuffers runtime*
 - flatbuffers>=1.8.0

tcanvasmagic: Demo of new proposed notebook magic for ROOT.



```
environment.yml
```

```
channels:
```

- conda-forge

```
dependencies:
```

- root

Other

- `conda config --add channels conda-forge;`
- `conda create -q -n testenv python=${PYVER} root;`

CI

- Uproot now tests against ROOT on Travis CI using Conda.
- Many SciKit-HEP packages now can/are adding support for ROOT in testing.
- Simple to add to any CI system

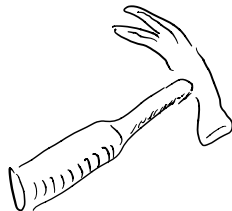
```
docker run --rm -it continuumio/miniconda3  
conda config --add channels conda-forge  
conda install -y python=3.7 root
```

Docker

- You can use the docker image for Conda to setup ROOT!
- Recipe for pre-configured ROOT on Conda docker [available](#)

More resources

- Release blogpost on [iSciNumPy.gitlab.io](https://iscinumpy.gitlab.io)
- Recipe at [/conda-forge/root-feedstock](https://github.com/conda-forge/root-feedstock)
- Join the [ROOT Conda mattermost channel](#)
- See [Chris Burr's slides](#) for the HSF packaging working group



The root-feedstock was created by Chris Burr and Henry Schreiner, with assistance from Enrico Guiraud (ROOT) and Patrick Bos (NLeSC). Guilherme Amadio (ROOT), Axel Naumann (ROOT), and Danilo Piparo (ROOT) provided valuable advice and support. Special thanks to the original NLeSC formula by Daniela Remenska.

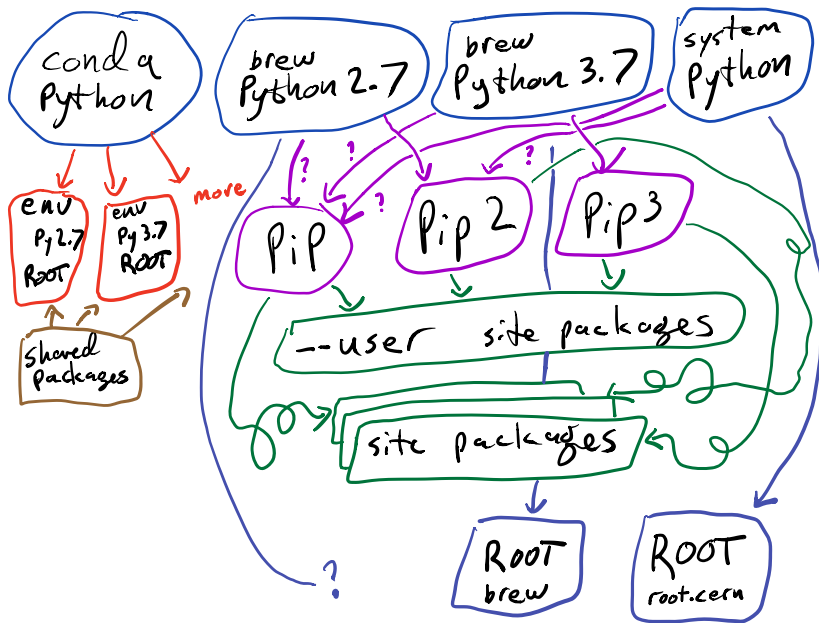
Future ideas

- Make root a metapackage (allows a smart user to avoid some modifications using root-base)
- Make a root-minimal package with a small set of features
 - ▶ The package is only 50-100 MB, deps are another 400 MB though
 - ▶ Won't be fully useful until ROOT modules are finished
- Remove current caveat on macOS
 - ▶ Currently you need to download and prepare the OSX 10.9 SDK
- Expand official ROOT testing to cover Conda-Forge
 - ▶ Might add nightlies in HEP channel or similar
- Add more packages to Conda-Forge for HEP
 - ▶ Already added XRootD, Pythia8, ...
 - ▶ Planned: EvtGen, Geant4, HepMC3, CLHEP, ...

Windows

- Windows will not be supported until ROOT supports 64 bit builds on Windows

Python connections



Conda in 30 seconds

- Language agnostic (Python, R, C++, Julia, ...)
- Supports all major OS's and distributions
- Changed a lot in the last two years
- Can install graphically from anaconda.com or on command-line for CI
- Doesn't require administrator privileges
- Packages are stored online in channels
- Strong support from industry (and HPC centers)
 - ▶ Intel: providing higher performance MKL Numpy
 - ▶ Nvidia: CUDA

Channels

- Defaults: the main channel that holds conda itself
- Anaconda: A set of 1,400+ packages, officially supported
- Conda-forge: A community run channel with 6,000+ packages
 - ▶ Powerful CI infrastructure builds “recipes”
 - ▶ Bots help manage updates
 - ▶ Officially recognized by Anaconda, unlimited storage
- User channels: For example “HEP”, with 3-10 GB limits
- Hosted channels: You can host your own channel anywhere
- Note: other channels include bioconda (6,000+ packages) and AstroConda (1,500+ packages)

Packages

- Binaries for each arch/Python version or universal
- Users do not build from source
- Can specify complex dependencies within a channel
- <IMPROVED> Can specify setup/teardown
- <IMPROVED> Consistent build platform
 - ▶ Linux: GCC 7.3
 - ▶ macOS: Clang 4.0.1
 - ▶ Windows: Why would I care?

Environments

- <NEW> Environments are activated with `conda activate` on all platforms
- You start with a “base” environment
- You can specify Python version, package version(s)
- Can be saved and reproduced from a file, exact or loose requirements

Package details

- A package consists of a yml recipe
 - ▶ Defines build and runtime dependencies
 - ▶ Usually includes build scripts and patches too
- Packages *must* be relocatable
 - ▶ Lots of tricks employed to help with this by conda-build
 - ▶ Should have no dependencies on the host system
- Packages should define all build and runtime dependencies

ROOT

- Lots of people in HEP already use Conda
 - ▶ Multiple versions of Python
 - ▶ Avoid breaking a system Python
 - ▶ Reproducible environments
 - ▶ Great way to get ML tools, even GPU versions (PyTorch, TensorFlow, ...)
 - ▶ Even some experiments have been using it: XENON1T, NEXT
- NLeSC provided a ROOT package for a while: “the most popular thing we ever produced”
 - ▶ ROOT 6.05 and Python 2.7 and 3.4 is latest version
 - ▶ Only worked properly on some OS's
 - ▶ The focus was on providing IO to get data in/out of ROOT files

Conda-forge ROOT

- Discussions began at the ROOT users workshop in Sarajevo, 2018
- Chris Burr built the first Linux ROOT 6.14.06 package for Conda-forge
 - ▶ He now has merge rights in staged-recipes, and is involved in many packages
- macOS was added soon after
 - ▶ Large number of improvements and fixes based on user feedback
- Release of 6.16.00 was only a few days after official release
 - ▶ Release day updates possible in the future
- Many patches accepted into the ROOT source
 - ▶ In 6.16.00, and will be in 6.16.02 and 6.18.00

Features

- Full ROOT support (Not just PyROOT and JupyROOT!)
 - ▶ Use the interpreter, compile, hadd, rootbrowse, everything present.
 - ▶ Most options turned on (Minuit2, RooFit, Pythia8, ...)
 - ▶ C++17 support
 - ▶ As many packages separated as ROOT allows (Clang, ...)
- You can have ROOT, Tensorflow GPU, and Python 3.7 in one environment, and ROOT, PyTorch, and Python 2.7 in another!
- Many Unix OS's supported (CentOS6+,)
- The ROOT team is working on adding nightly tests for the conda packages (might be available in the HEP channel)

Perfect reproduction

The normal method is probably fine, but if you really want to store the exact packages:

```
conda list --explicit > spec-file.txt
```

followed by

```
conda create --name myenv --file spec-file.txt
```

will perfectly recreate an environment on the same OS.