PSRCHIVE PYTHON BINDINGS

Golam Shaifullah

with Sarah Buchner, Renée Spiewak & Avishek Basu

i do not know what it is about you that closes
and opens; only something in me understands
the voice of your eyes is deeper than all roses

– E. E. Cummings

ASTRON
Netherlands Institute for Radio Astronomy

PSRCHIVE Python bindings

The python bindings expose a subset of the PSRCHIVE (i.e., C++) classes as python objects.

- The classes are not programs like pam, pat, psredit, etc but rather (data) objects that you can manipulate as numpy arrays or matplotlib plot objects.
- See more at http://psrchive.sourceforge.net/manuals/python

DISCLAIMER!

These slides borrow very heavily from Paul Demorest's talk at the IPTA 2018 in New Mexico.

PSRCHIVE Python bindings : Why is this useful?

- Python (plug your favourite reason here)
- Direct access to data values for exploration, debugging, etc.
- Prototyping or implementation of new analysis routines (often easier in Python than C++).
- Greater scripting complexity (while avoiding the subprocess overload).

However, not straightforward to reproduce complex PSRCHIVE applications (pac, pat).

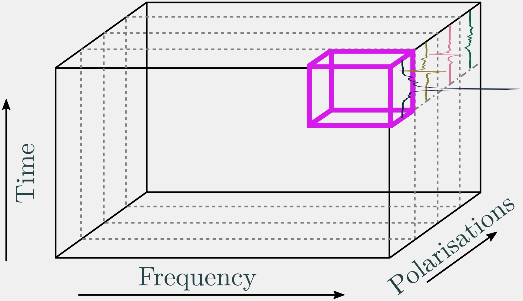
The beating heart

- Essentially, autogenerated wrapper code in python that allows direct calls to C++ code
- “Simplified Wrapper and Interface Generator” aka SWIG.
- Can in principle generate bindings for languages besides Python (project for the brave!).
- http://swig.org

Installating PSRCHIVE Python bindings

- the Python wrapper come swith the standard package.
- Dependencies (beyond those of standard PSRCHIVE):
  - Python, with development headers ("apt install python-dev" or similar pacakge).
  - SWIG
  - NumPy
  - Very useful but not required: SciPy, matplotlib, ipython/jupyter
- When building PSRchive, do '$ ./configure –enable-shared' (this compiles shared libraries)
- You can also supply the path to preferred flavour of python (i.e., python3 is possible)

PSRCHIVE Python bindings



A super-simple example:The three fundamental PSRCHIVE data classes

- Profile is a single pulse profile – data as a function of pulse phase only.
- Integration is a set of pulse profiles recorded simultaneously – usually profiles as a function of frequency channel and/or polarization.
- Archive is a set of Integration as a function of time. Represents a single data file.

PSRCHIVE Python bindings

Accessing data in Python using these classes

- Archive:
  - use archive.get_Integration(isub)
  - or archive[isub] to retrieve an Integration
- Integration:
  - use integration.get_Profile(ipol,ichan) to retrieve a single Profile
- Profile:
  - use profile.get_amps() to return data as a NumPy array

PSRCHIVE Python bindings

Shortcut to get all data:

- Use archive.get_data() to return entire (N sub , N pol , N chan , N bin) data cube as a NumPy array.Plotting example:Plotting example:Plotting example:Plotting example:Accessing data using these classes

PSRCHIVE Python bindings

- ‘view’ the original data with profile.get_amps()
- work with a copy - archive.get_data()
- save to disk with archive.unload(“new_filename”)

PSRCHIVE Python bindings

Archive has a large number of methods (functions) for performing common data processing steps.

- Common examples: dedisperse(), remove_baseline(), fscrunch(), tscrunch(), pscrunch(), convert_state(), ...
- archive.execute("[psrsh code...]") will run any psrsh command on the archive.

How to learn what else is available?

- Browse the PSRCHIVE class documentation at http://psrchive.sourceforge.net/classes/psrchive
- Tab completion in ipython is very useful!

PSRCHIVE Python bindings

Most use cases only need Archive, Integration, and Profile classes. But some of the PSRCHIVE algorithm classes are also included in the Python interface. For example, ProfileShiftFit for doing template-matching. No comprehesive list of these unfortunately. Browse the C++ class docs, contact Paul Demorest, Maciej Serylak or raise a ticket at the PSRchive site if you want something added. Advanced PSRCHIVE classes:

- Data profile
- Standard (aka template)
- profileSummary

Thank You!

References & Literature

Hotan A. W., van Straten W., Manchester R. N., 2004, PASA, 21, 302

van Straten W., 2006, ApJ, 642, 1004

van Straten W., 2013, ApJS, 204, 13

van Straten W., Manchester R. N., Johnston S., Reynolds J. E., 2010, PASA, 27, 104

van Straten W., Demorest P., Khoo J., Keith M., Hotan A., et al. 2011, PSRCHIVE: Development Library for the Analysis of Pulsar Astronomical Data (ascl:1105.014)

van Straten W., Demorest P., Oslowski S., 2012, Astron. Res. & Tech., 9, 237