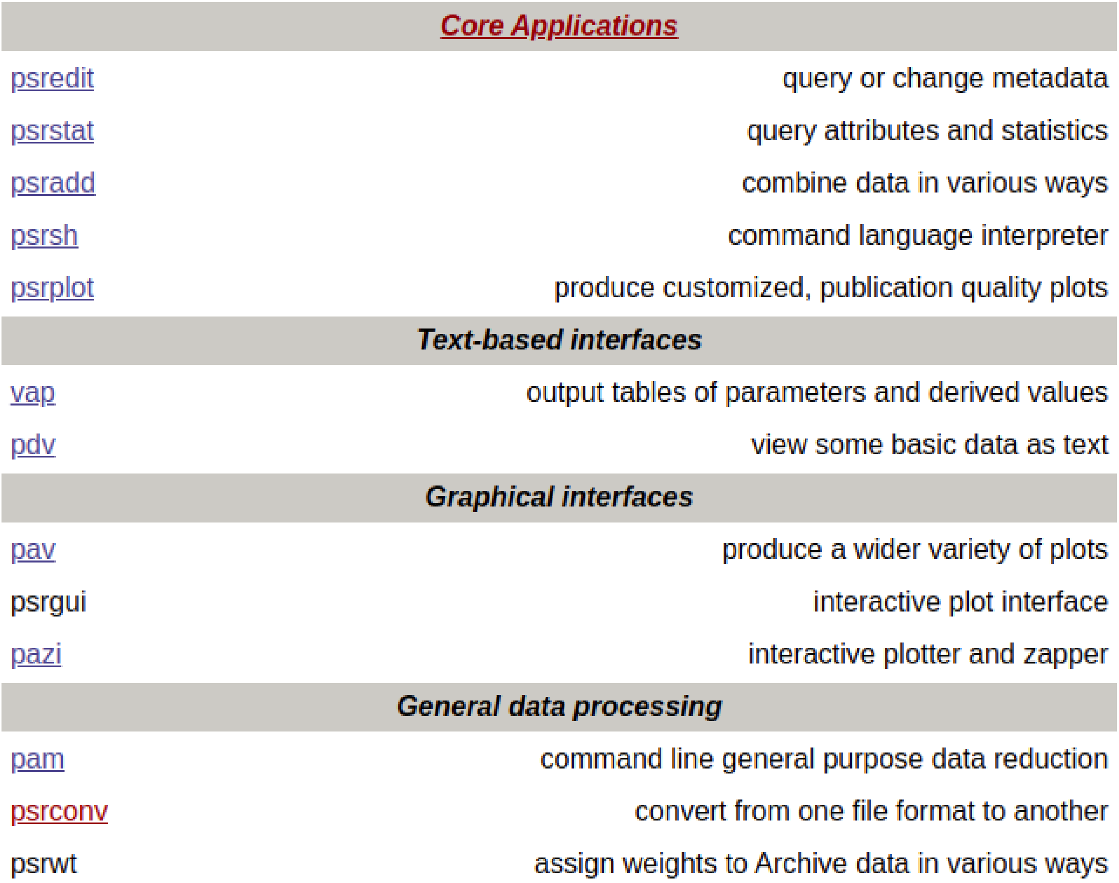
A brief Intro to PSRCHIVE

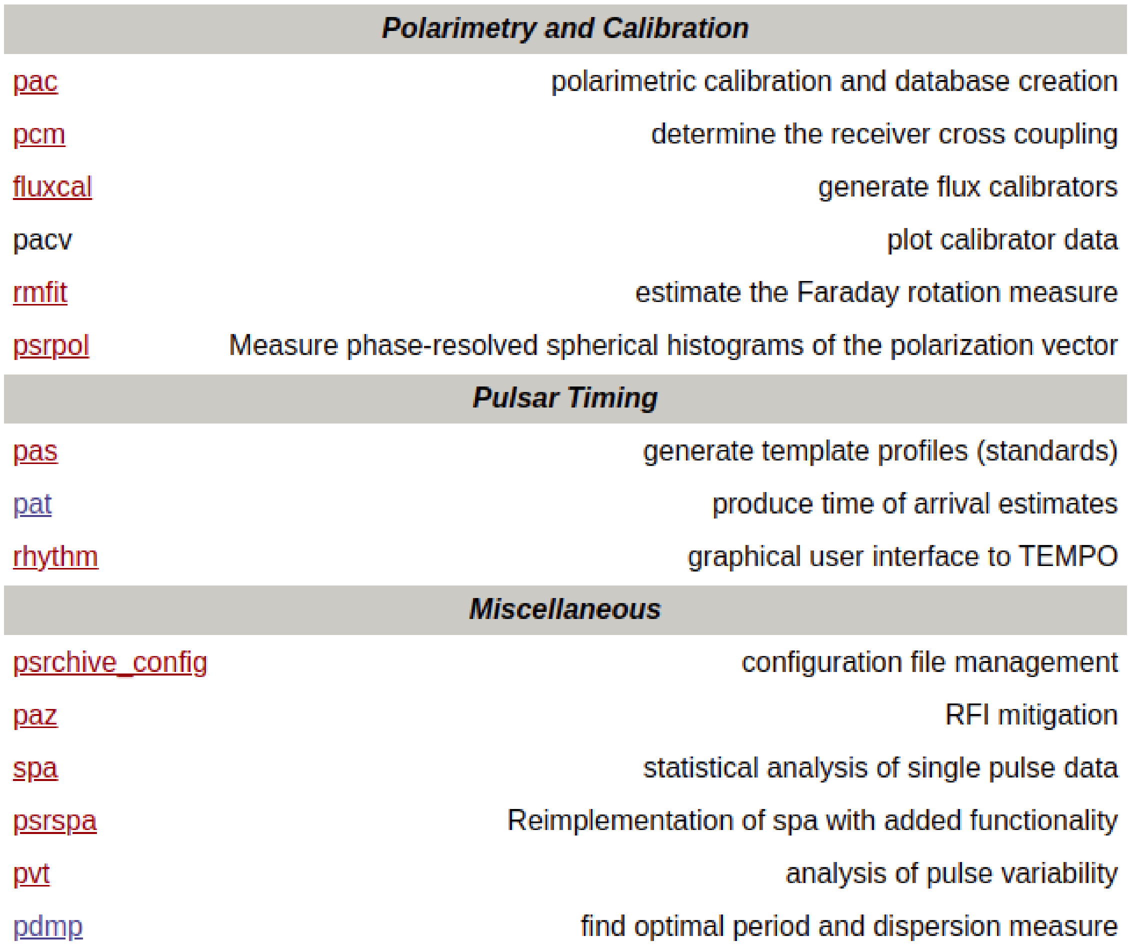
- PSRCHIVE is an open source C++ library for analysis of pulsar astronomical data.

- Detailed documentation: http://psrchive.sourceforge.net/manuals/, Willem van Straten et al. 2012 (https://arxiv.org/pdf/1205.6276.pdf)

- List of commands



Core Applications	
psredit	query or change metadata
psrstat	query attributes and statistics
psradd	combine data in various ways
psrsh	command language interpreter
psrplot	produce customized, publication quality plots
Text-based interfaces	
vap	output tables of parameters and derived values
pdv	view some basic data as text
Graphical interfaces	
pav	produce a wider variety of plots
psrgui	interactive plot interface
pazi	interactive plotter and zapper
General data processing	
pam	command line general purpose data reduction
psrconv	convert from one file format to another
psrwt	assign weights to Archive data in various ways



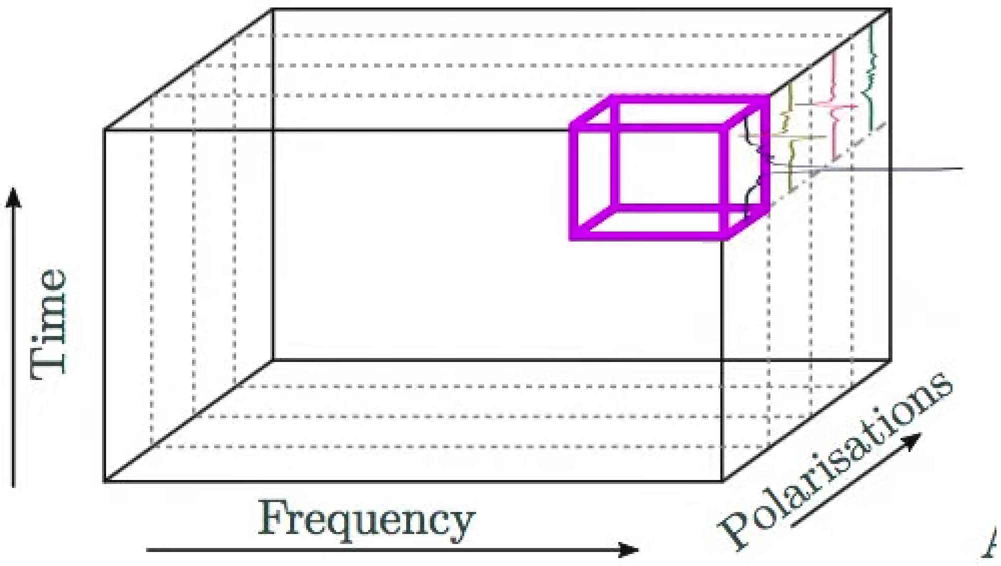
Polarimetry and Calibration	
pac	polarimetric calibration and database creation
pcm	determine the receiver cross coupling
fluxcal	generate flux calibrators
pacv	plot calibrator data
rmfit	estimate the Faraday rotation measure
psrpol	Measure phase-resolved spherical histograms of the polarization vector
Pulsar Timing	
pas	generate template profiles (standards)
pat	produce time of arrival estimates
rhythm	graphical user interface to TEMPO
Miscellaneous	
psrchive_config	configuration file management
paz	RFI mitigation
spa	statistical analysis of single pulse data
psrspa	Reimplementation of spa with added functionality
pvt	analysis of pulse variability
pdmp	find optimal period and dispersion measure

- To know how to use a command go to the PSRCHIVE manual page or just run
  - **$ command -h**

The pulsar archive:

- The PSRCHIVE commands are run on pulsar archives. For InPTA, these archives are the *.fits files produced by **pinta**.

- Pulsars archives contain astronomical data recorded during pulsar observations and are typically stored as a three-dimensional array of pulse profiles, the axes being time (sub-integration), frequency (channel) and polarization.



- It further contains the metadata that tell us about various settings of the observation.

psredit:

- The Pulsar Archive Editor, psredit, enables one to query or change the parameters in a Pulsar Archive.

- First run **$psredit -h** to look at the help documents. Look at these useful options:
  - **-m, -e, -c**

- Just running **$psredit pulsar.fits** will list all the metadata for the archive. Try it on one of the archives. Few important metadata includes (but not limited to)
  - **nbin, nchan, npol, nsubint, freq, bw, dm, dmc, length, rcvr:name, be:name, be:config, ….**

- To get a particular quantity, center frequency of observation for example, run **$psredit -c freq pulsar.fits**

- To change a quantity, e.g. dm, run **$psredit -c dm=10.4 -e dmchanged J1909-3744_59348.989243_500.rfiClean.fits**
  Check the dm again with **$psredit -c dm J1909-3744_59348.989243_500.rfiClean.dmchanged**

Don’t use psredit to change any metadata unless you know for sure what you are doing.

psrstat:

- The pulsar data statistical query interface, psrstat, provides access to various quantities that can be derived from the data stored in a Pulsar Archive.

- Run **$psrstat -h**

- Run **$psrstat pulsar.fits** and check the output

- Calculate the SNR of the profile: **$psrstat -c snr J1909-3744_59348.989243_500.rfiClean.fits**
  What do you see? Why is that?

- Calculate SNR of the time and frequency scrunched profile: **$psrstat -c snr -j tscrunch,fscrunch J1909-3744_59348.989243_500.rfiClean.fits**
  Or,
  $psrstat -c snr -j TF J1909-3744_59348.989243_500.rfiClean.fits

- We can use pdmp method to calculate the SNR: **$psrstat -c “snr=pdmp” -c snr -j TF J1909-3744_59348.989243_500.rfiClean.fits**

  If you are copy pasting from here, make sure to change the quotation marks.

vap:

- The Viewer of Archive Parameters, vap, is designed to interrogate headers and glean simple numerical quantities. Feel free to try it out.

- **$vap -E pulsar.fits** prints out the par file used to fold the data.

psrplot:

- The Pulsar Archive Plotter, psrplot, is a highly configurable plotting program that may be used to create customized plots of pulsar data.

- Run **$psrplot -h**

- Run **$psrplot -P**. Useful plot types include (but not limited to):
  - **flux, freq, freq+, time, bpcw, ….**

- **$psrplot -p flux -j FT J1909-3744_59348.989243_500.rfiClean.fits** and then **/xs**

  Or

  $psrplot -p flux -j FT J1909-3744_59348.989243_500.rfiClean.fits -D /xs

- **$psrplot -p freq -j T J1909-3744_59348.989243_500.rfiClean.fits -D /xs**

  What do you see? Why?

- **$psrplot -p freq -j T,dedisperse J1909-3744_59348.989243_500.rfiClean.fits -D /xs**

  Or

  $psrplot -p freq -j TD J1909-3744_59348.989243_500.rfiClean.fits -D /xs

- Try **$psrplot -p freq -j TD J1909-3744_59348.989243_500.rfiClean.dmchanged -D /xs**

  What do you see? Explain.

- **$psrplot -p time -j F J1909-3744_59348.989243_500.rfiClean.fits -D /xs**

- **$psrplot -p freq+ -j TD J1909-3744_59348.989243_500.rfiClean.fits -D /xs**

- Plot 2 profiles side by side:

  $psrplot -N 2x1 -p flux -j FT J1909-3744_59348.989243_500.rfiClean.fits J1909-3744_59380.815942_500.rfiClean.fits -D /xs

- Plot the t-scrunched profile of a particular chanel

  $psrplot -p flux -c chan=1 -j T J1909-3744_59348.989243_500.rfiClean.fits -D /xs

  $psrplot -p flux -c chan=20 -j T J1909-3744_59348.989243_500.rfiClean.fits -D /xs

- Plot the f-scrunched profile of a particular subint

  $psrplot -p flux -c subint=100 -j F J1909-3744_59348.989243_500.rfiClean.fits -D /xs

- Zoom into the profile
  ```
  psrplot -p flux -c 'x:range=(0.2,0.5)' -j FT
  J1909-3744_59348.989243_500.rfiClean.fits -D /xs
  ```

- To create a publication quality plot
  ```
  $psrplot -p flux -c 'set=pub, y:range=(0,1.2)' -j FT
  J1909-3744_59348.989243_500.rfiClean.fits -D /xs
  ```

- To save a plot in the machine
  ```
  $psrplot -p flux -c 'set=pub, y:range=(0,1.2)' -j FT
  J1909-3744_59348.989243_500.rfiClean.fits -D /cps
  ```
  It will create a postscript file named **pgplot.ps**. You can see it using **$evince pgplot.ps**. You can convert it to PDF using **$ps2pdf pgplot.ps**

- To plot two different type of plots of the same profile side by side
  ```
  $psrplot -N 2x1 -p time -p freq -jD -j:0:F -j:1:T
  J1909-3744_59348.989243_500.rfiClean.fits -D /xs
  ```

- **Feel free to explore other option..**

pav:

- The Pulsar Archive Viewer, pav, is designed to visually display the data stored in an archive.

- **$pav -h**

- Explore different options including: **-D, -G, -Y, -b, -r, -f, -t, -N, --cmap, ….**

pam:

- The Pulsar Archive Manipulator, pam, is designed to do the dirty work of messing around with archives on disk, processing them, compressing them, tweaking them... As such it is one of the most dangerous of the PSRCHIVE applications.

- **$pam -h**

- **-m** →modify original file, **-e ext** →write to new file with extension ext

- Try these following options out. Few of the examples are given below.
  - **-m, -e, -T, -F, -D, --DD, -t, -f, -b, --setnsub, --setnchn, --setnbin, -d, -E, --reverse_freqs, --update_dm, …**

- Tsrunch all profiles and add extension Tcoll: **$pam -T -e Tcoll J1909-3744_59348.989243_500.rfiClean.fits**

- Update the par file of the Tcoll profiles and update header dm: **$pam -m -E J1909-3744.par --update_dm *.Tcoll**

- De-disperse the Tcoll files with the header dms: **$pam -m -D *.Tcoll**

pazi:

- pazi allows user to visualize integrated, freq-resolved and time-resolved pulse profiles at one go, and also interactively zap freq channels or subint

- **$pazi -h**

- Interactive use of **f, t, zoom, reset zoom, zap, zap multiple, undo, save**

- **$pazi J1909-3744_59348.989243_500.rfiClean.fits**

psrsmooth:

- psrsmooth is used to create frequency resolved smoothened template of pulsar profiles

- **$psrsmooth -W -t <wavelet type> pulsar.fits**

- **$psrsmooth -help=t** shows all the wavelet types options

- Run **$psrsmooth -W -t UB103 J1909-3744_59469.499588_500.rfiClean.Tcoll**

- It will create a ***.sm** file. Run **$pazi *.sm** to visualize the smoothened profile

pat:

- pat is used to determine the time of arrival (TOA) of each integrated Profile in an Archive

- **$pat -h**

- **$pat -j”F 8” -A FDM:mcmc=1 -f tempo2 -P -X ‘-sys GM_GWB_500_200_b1’ -s J1909-3744_59469.499588_500.rfiClean.Tcoll.sm *.Tcoll > J1909-3744.tim**

  If you are copy pasting, make sure to replace quotation marks!

- Take a look at the tim file

Few other commands to look at:

- **psradd**: add many profiles in phase to create a single very high SNR profile. Generally used to make F-scrunched template.

- **pdv**: output the profile as an ascii file. Try out this command to get an ascii version of phase vs intensity of the profile with 64 bins **$pdv -t -j DFT -B 8 J1909-3744_59348.989243_500.rfiClean.fits**

- **pas, paas**: align and create frequency scrunched templates

- **pdmp**: find optimal period and dispersion measure

PSRCHIVE python interface:

```
$python3
>>> import psrchive
>>> import numpy as np
>>> import matplotlib.pyplot as plt
>>>
>>> arch = psrchive.Archive_load('J1909-3744_59348.989243_500.rfiClean.fits')
>>> arch.get_source()

>>> data = arch.get_data()
>>> data.shape
```

>>>