Interacting with Pulsar Data

Overview

Teaching: 60 min
Exercises: 30 min
Questions

- How do I inspect my pulsar data
- How can I remove RFI from my observations

Objectives

- Learn the software used to process pulsars

Introduction

In this lesson we will go over how to interact with pulsar data. We will go over the different types of data, how to plot them and how to clean them.

The data we will use for this lesson can be downloaded with the following command:

Bash

```
wget  "ftp://elwood.ru.ac.za/pub/geyer/NWU_pulsartiming/data/Session1_pulsar_data.tar.gz"
```

Then untar it with

Bash

```
tar -xvf Session1_pulsar_data.tar.gz
```

And move into the directory it creates with

Bash

```
cd pulsar_data_lesson
```

Pulsar data types

Pulsar data are typically stored as a three-dimensional array of pulse profiles the axes being time (sub-integrations), observing frequency (channels) and polarization. Each data file (typically termed as archives) have attributes (metadata) that describe the pulsar observation. Pulsar data can be broadly sorted into two types, "raw voltage" which is the time series data as it comes off the telescope and "folded" which has been folded and dedisperesed using a timing ephemeris.

Some formats for raw voltage files include `PSRFITS`, `PSRDADA` and `VDIF`. Before analyzing the data you often have to fold and dedisperse the data which is often done with `dspsr` (explained in the next section).

The folded data are often called archives and have `.ar` at the end of their filename. Many telescopes (such as MeerKAT) have a pulsar timing backend that will automatically produce a folded archive to save processing and data size.

DSPSR

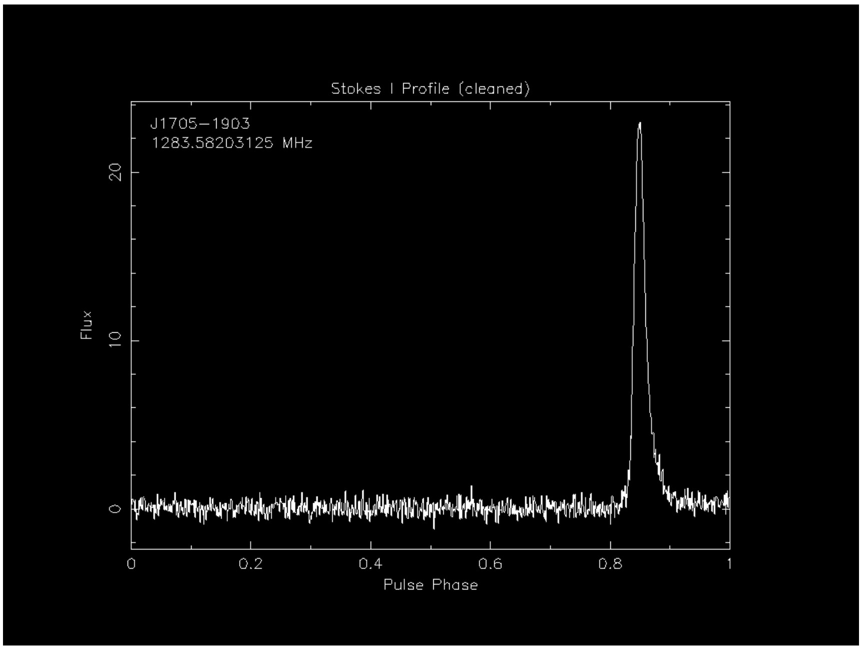
DSPSR (https://dspsr.sourceforge.net/manuals/dspsr/) is not installed on the VM or need for MeerKAT data but it is still useful to know how to use to process other data and to understand MeerKAT data. `dspsr` can perform phase-coherent dispersion removal while folding the data based on the input ephemeris. The output data may be divided into sub-integrations of arbitrary length, including single pulses. THe MeerKAT PTUSE outputs the archives in ~8 second sub-integrations so the equivalent `dspsr` command may look like this:

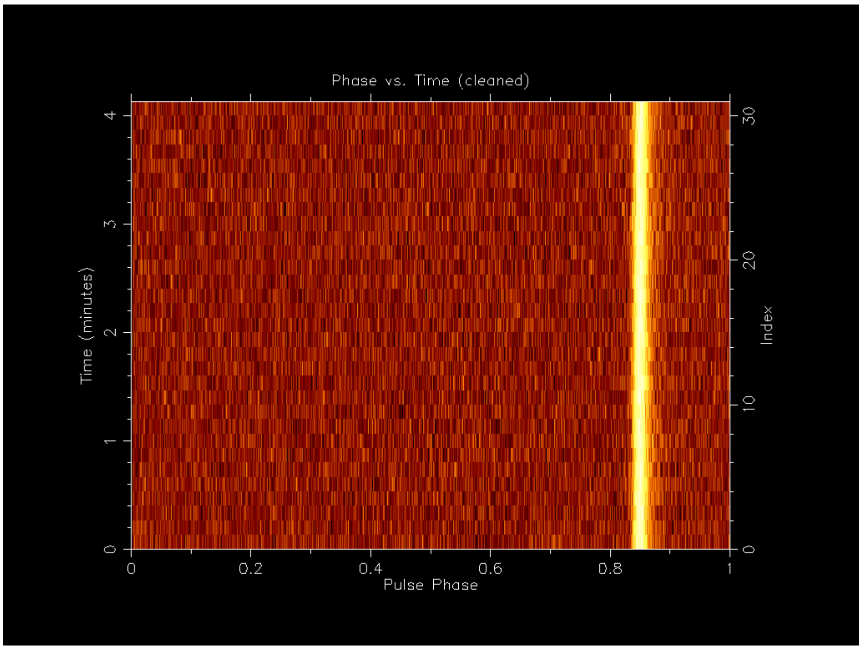
Code

```
dspsr -F 128 -E 1644-4559.eph -L 8 raw_observation.data
```

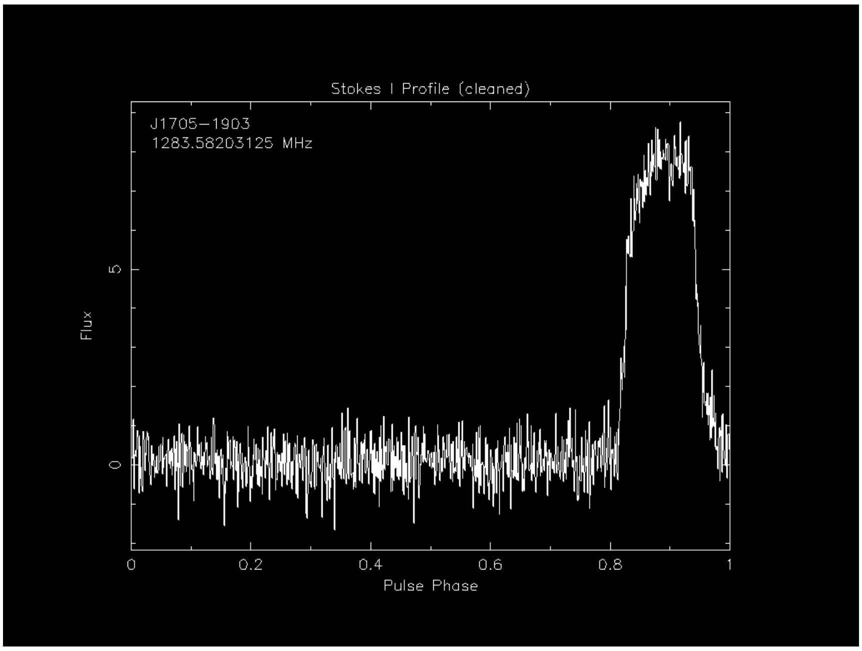
where `-F 128` outputs a 128 frequency channel archive, `-E 1644-4559.eph` is the ephemeris file, `-L 8` outputs an archive for every 8 seconds of data and `raw_observation.data` is whatever the input raw voltage file is called (if there is one).

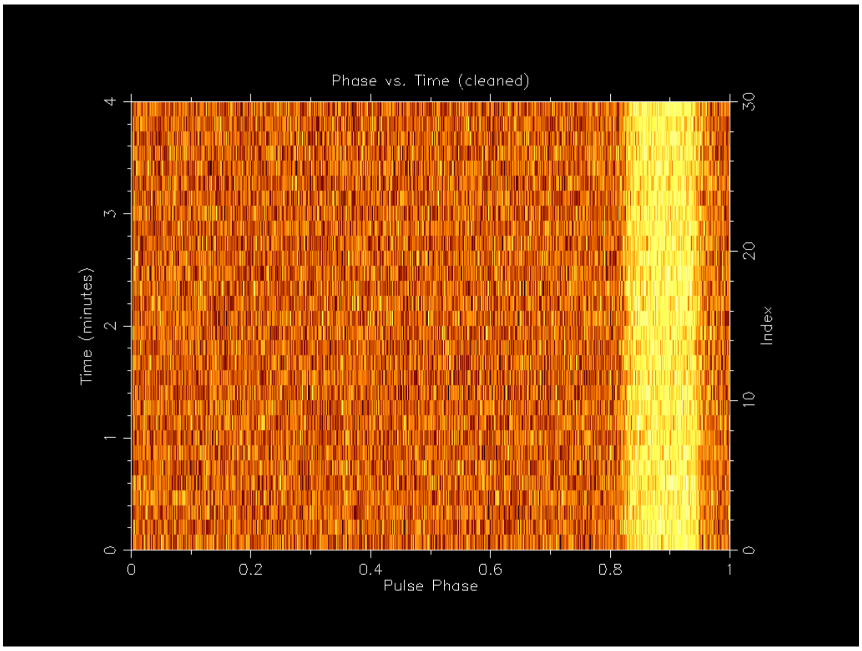
Because these archives have already been folded with an ephemeris, there is only so much you can do to improve your observation if the ephemeris is inaccurate. For example below is a recent observation (2023-04-21-23:58:20) of PSR J1705-1903's profile and phase against time plots:





You can see that pulse is thin as expected. If we then compare this to an early observation (2019-04-23-05:44:10) we can see that the pulse profile is much wider even though we have a applied a more recent ephemeris to it.





This is due to the PTUSE outputting the 8 second sub-integrations which have some timing smear in them due to the initial inaccurate ephemeris used during the observation. If you ever want to inspect the ephemeris used to create an archive file you can use the following command:

Bash

```
vap -E <archive_file>
```

PSRCHIVE

Now that we have folded and dedisperesed archives, we can analysis them and one of the best packages to do that is PSRCHIVE. PSRCHIVE is an Open Source C++ development library for the analysis of pulsar astronomical data. The software is described online (https://psrchive.sourceforge.net/index.shtml) and in Hotan, van Straten & Manchester (2004) (http://www.publish.csiro.au/?paper=AS04022) and Straten, Demorest & Osłowski 2012 (https://arxiv.org/abs/1205.6276).

There is a full list (https://psrchive.sourceforge.net/manuals/) of commands online and below are the most common ones:

- `psrstat` query attributes and statistics
- `psradd` combine data in various ways
- `psrplot` produce customized, publication quality plots
- `vap` output tables of parameters and derived values
- `pav` produce a wider variety of plots
- `pam` command line general purpose data reduction
- `rmfit` estimate the Faraday rotation measure
- `pas` generate template profiles (standards)
- `pat` produce time of arrival estimates
- `paz` RFI mitigation
- `pdmp` find optimal period and dispersion measure

Investigating files (`vap` and `psrstat`)

Archive files have a lot of metadata associated with them and you can use the `vap` and `psrstat` command to view them. The main difference between the two is that `vap` outputs the metadata in a table format and `psrstat` outputs it in a more human readable format and includes more statistics.

Lets look at the metadata of one of the archives we will be using for this lesson. First lets look at the `vap` output:

Bash

```
vap -c nchan,nsub,length J1903-7051_2022-07-17-22:44:02_*.ar
```

Output

```
filename   nchan   nsub   length
J1903-7051_2022-07-17-22:44:02_raw.ar   1024   31   248.000
J1903-7051_2022-07-17-22:44:02_zap.ar   1024   31   248.000
```

Bash

```
psrstat -c nchan,nsubint,length J1903-7051_2022-07-17-22:44:02_*.ar
```

Output

```
J1903-7051_2022-07-17-22:44:02_raw.ar nchan=1024 nsubint=31 length=248
J1903-7051_2022-07-17-22:44:02_zap.ar nchan=1024 nsubint=31 length=248
```

You'll notice that they output the values in different formats and that they label the time sub-integrations differently (`nsub` vs `nsubint`). The `psrstat` command also has access to more statistics and you can output all of the by not including the `-c` option:

Code

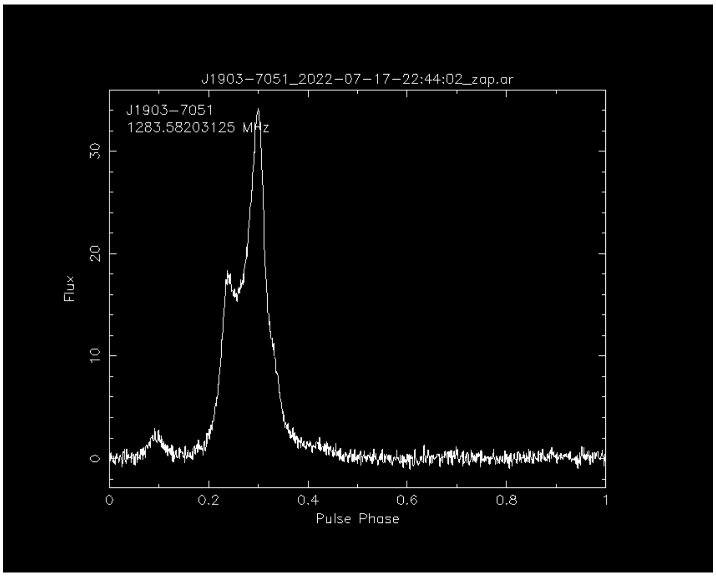
```
psrstat J1903-7051_2022-07-17-22:44:02_zap.ar
```

Pulse profile

Lets give it a go and try to make a pulse profile plot with the command:

Bash

```
psrplot -p flux -jFTDp -D J1903-7051_profile_fts.png/png J1903-7051_2022-07-17-22:44:02_zap.ar
```

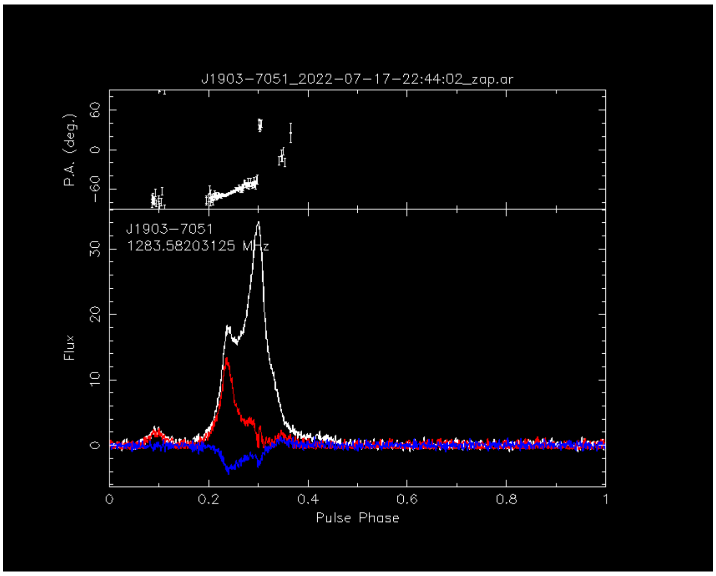


You can also remove the `-D J1903-7051_profile_fts.png/png` part and it will open the plot in a window. Now lets go through all the common plot type commands

Polarisation (Stokes) profile

Bash

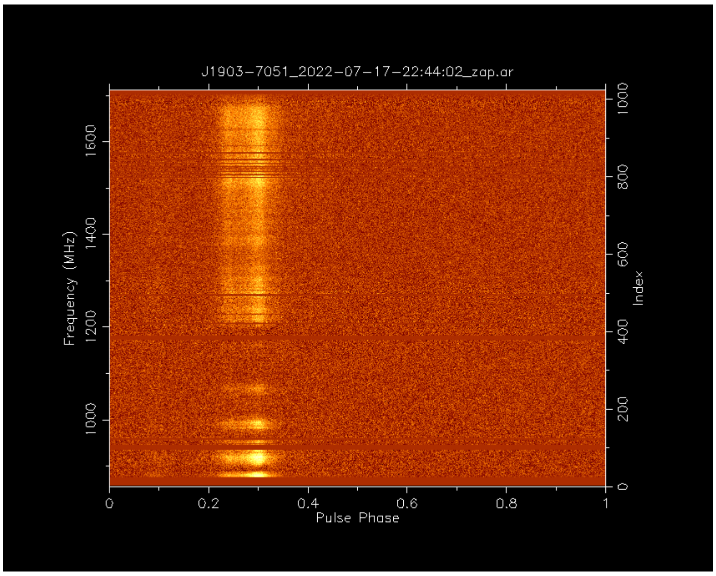
```
psrplot -p Scyl -jFTD -D \J1903-7051_profile_ftp.png/png J1903-7051_2022-07-17-22:44:02_zap.ar
```



Phase vs. Frequency

Bash

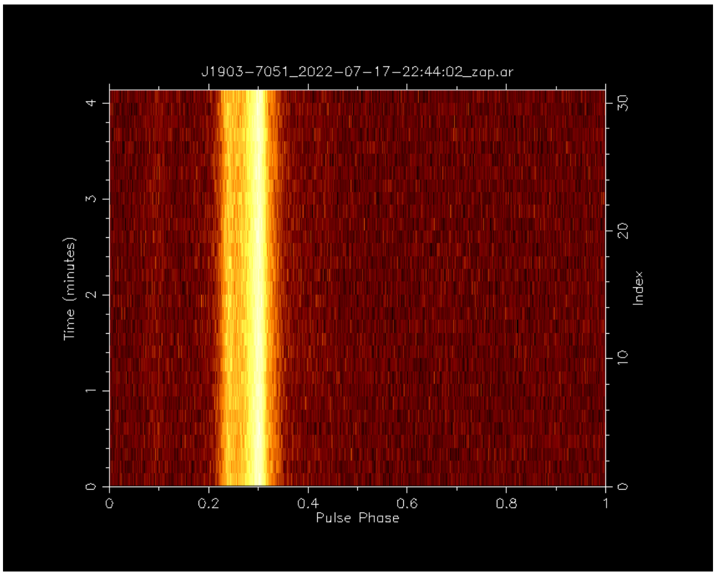
```
psrplot -p freq -jTDp -D J1903-7051_phase_freq.png/png J1903-7051_2022-07-17-22:44:02_zap.ar
```



Phase vs. Time

Bash

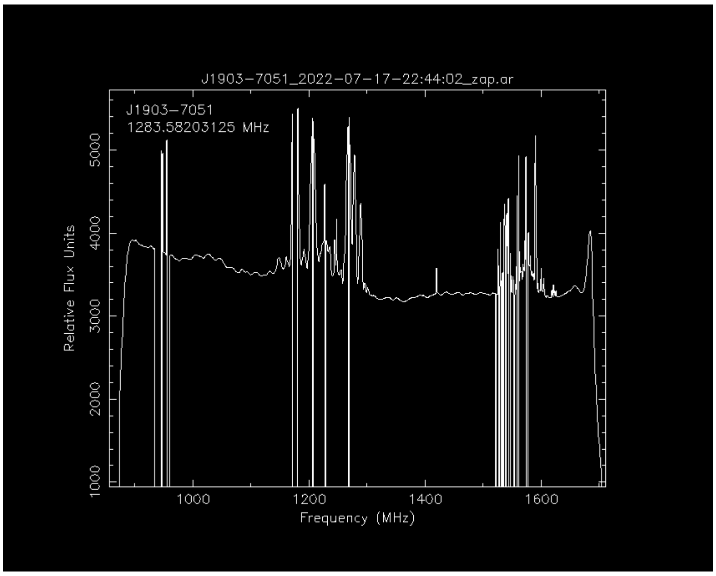
```
psrplot -p time -jFDp -D J1903-7051_phase_time.png/png J1903-7051_2022-07-17-22:44:02_zap.ar
```



Cleaned bandpass

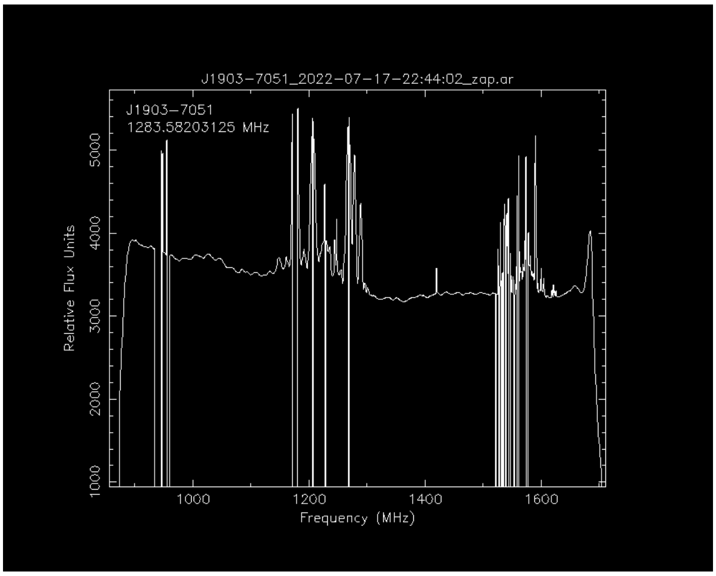
Bash

```
psrplot -p b -jT -D J1903-7051_bandpass.png/png J1903-7051_2022-07-17-22:44:02_zap.ar
```



RFI cleaning

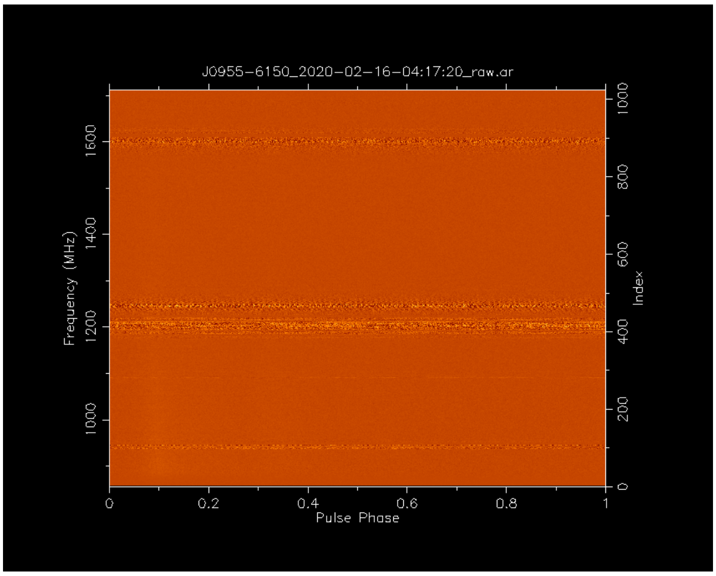
While most telescopes are situated in radio quiet zones, it is impossible to remove all sources of RFI. For this reason, it is important to remove RFI from your data before analysis. The `J1903-7051_2022-07-17-22:44:02_zap.ar` archive has already been RFI cleaned and we can see in the bandpass which frequencies were flagged:



We have a uncleaned file that we can use the different technquies on to do our best to clean the archive. We will be using `J0955-6150_2020-02-16-04:17:20_raw.ar` as our example file which has signifficant RFI in it which we can when we plot it with the following command:

Bash

```
psrplot -p freq -jTDp -D J0955-6150_phase_freq_raw.png/png J0955-6150_2020-02-16-04:17:20_raw.ar
```



Paz

`paz` (https://psrchive.sourceforge.net/manuals/paz/) is a psrchive command that can do some automated and manual RFI removal. You can use it manually or as a part of other commands.

What I mean by manually is you can use it to create a new archive with a command like so:

Bash

```
paz -r -e paz_median J0955-6150_2020-02-16-04\:17\:20_raw.ar
```

where `-r` zap channels using median smoothed difference and `-e paz_median` outputs a new file ending in `.paz_median`.

You can then plot this with the following command:

Bash

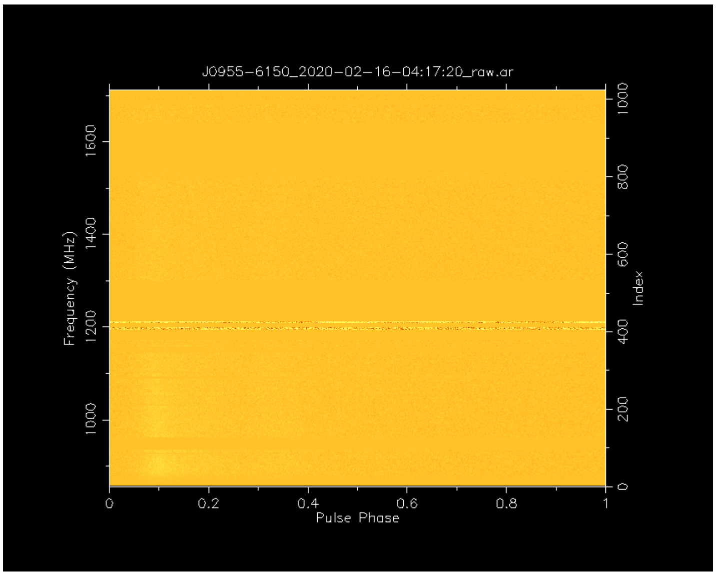
```
psrplot -p freq -jTDp -D J0955-6150_phase_freq_median.png/png J0955-6150_2020-02-16-04:17:20_raw.paz_median
```

Or I can produce the same result by addind the `,"zap median` to the psrplot command to do this in one go (note that I'm using the original `.ar` file)

Bash

```
psrplot -p freq -jTDp,"zap median" -D J0955-6150_phase_freq_median.png/png J0955-6150_2020-02-16-04:17:20_raw.ar
```

Both of these methods will create the following output:



As you can see it removed enough RFI that we can now see the pulsar but there is still some major RFI that was missed.

Pazi

For interactive RFI removal we can use `pazi` (https://psrchive.sourceforge.net/manuals/pazi/) which will create a window where we can select which frequency or time range we wish to flag.

If we run the `pazi` command with no options it will output the following help:

Output

```
pazi

A user-interactive program for zapping subints, channels and bins.
Usage: pazi [filename]

Options.
  -h                        This help page.

Mouse and keyboard commands.
  zoom:                     left click, then left click
  reset zoom:               'r'

Modes:
  phase-vs-frequency:       'f'
  phase-vs-time:            't'
  binzap-integration:       'b' (must be in phase-vs-time mode)

  center pulse:             'c'
  toggle dedispersion:      'd'

  zap:                      right click
  zap (multiple):           left click, then right click
  undo last:                'u'

  In binzap mode:
    -  zoom and reset zoom as above
    -  zap phase range:     left click, then right click
    -  mow the lawn:        'm'
    -  prune the hedge:     'x' to start box, 'x' to finish

  save (<filename>.pazi):   's'
  quit:                     'q'
  print paz command:        'p'
```

You'll want to run `pazi <file_name>` then click `f` to go to phase-vs-frequency mode then left click and the start of a frequency range then right click and the end of the range that you want to zap. When you are done press `s` and `q` and it will make a new file ending in `.pazi`.

✎ Remove as much RFI as you can

You can estimate the signal-to-noise ratio of an archive with the following command

Code

```
psrstat -j FTp -c snr=pdmp -c snr <archive_file_here>
```

Try using pazi to remove as much RFI as you can which keeping the S/N high. Whoever gets the highest S/N wins!

PDMP

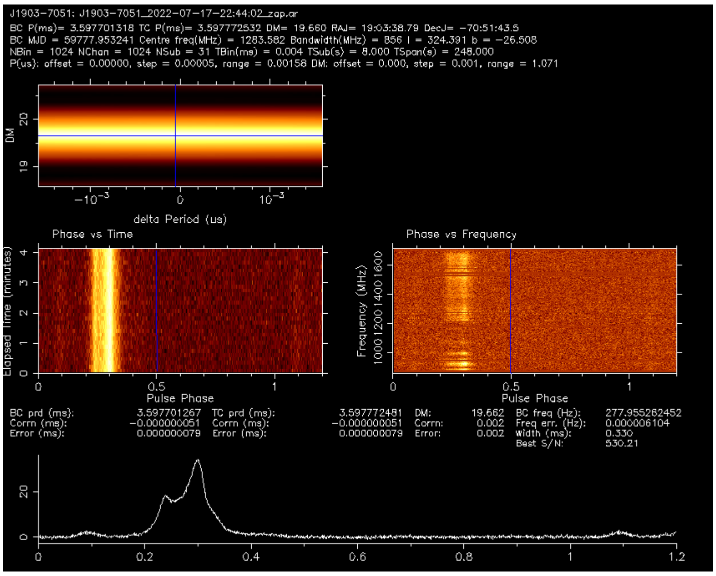
`pdmp` (https://psrchive.sourceforge.net/manuals/pdmp/) is a tool for find the optimal period and DM of a pulsar. This is useful when you have a pulsar that has not been accurately timed (perhaps a candidate or recent discovery). If the phase-vs-time or the phase-vs-freq plots don't show a vertical profile (you see a slope) then this may mean the period (if you see an angle in the phase-vs-time) or the DM (if you see a sweep in the phase-vs-freq).

It is a time consuming command so it may not finish during the work shop. You can run it like this

Code

```
pdmp -g J1903-7051_2022-07-17-22:44:02_zap_pdmp.png/png J1903-7051_2022-07-17-22:44:02_zap.ar
```

And here is what it will output:



You can see that the pulse profile looks vertical in both frequency and time (although they already were). It also outputs an estimated period and DM in the plot and in the `pdmp.posn` and `pdmp.per`.

BC is barycentric which is based on the centre of the solarsystem and what is used by pulsar ephemerises TC is topocentric which is based on the earth's position. Because the aparent/observed period is different depending on the position of the earth as it orbits the as the change in distance and relative speed causes an effect similar to the doppler effect. Pulsar software will convert barycentric period to topocentric period for you so always give it the barycentric period.

Decimating

You can decimate or scrunch your data along the time, frequency and polarisation axis. A major benfit of this is that it makes the files much smaller. Another benefit is that the files will be in an easier format to make ToAs later on (convered in future lessons).

The wording used commonly used to describe how the data has been decimated is "nchan" for number of frequency channels and "nsub" for number of time sub-integrations. The following command demonstrates how to use `pam` (https://psrchive.sourceforge.net/manuals/pam/)

Code

```
pam --setnchn <nchan> --setnsub <nsub> <-p> -e <nchan>ch<-p>p<nsub>t.ar <archive_file>
```

where you use `-p` if you want to polarisation scrunch and `-e <nchan>ch<-p>p<nsub>t.ar` is my personal preference of how to label the file to make it clear how it has been decimated. So for example if you wanted 16 frequency channels and 8 time sub-integrations:

Code

```
pam --setnchn 16 --setnsub 8 -e 16ch4p8t.ar J1903-7051_2022-07-17-22:44:02_zap.ar
```

Organisation

Dealing with data can be a bit confusing and you may get a bit lost if you don't keep your data organised. The following is a few tips that will hopefully make your life easier.

- Use good names for your files. Verbose long names are better than short confusing names. You should include the pulsar name and the date it was observed as well as other things like if you have removed RFI and if and how you have decimated it.
- Make a script to automate your work. Once you have worked out how to process your data, you can make a script that will run all those commands for you which will save you time and make sure you don't forget a step.
- Use a kanban board (to do list) to track your processing. If you have a lot of data to process it may start to ger hard to keep track of what you have and haven't done. One way to make this easier is to use a kanban board like Trello (https://trello.com/) (see this (https://adacs-australia.github.io/research_project_management_training/03-trello/index.html) lesson on how to use it)
- Take notes on the commands you used. Taking notes will make it clearer what steps you took and may help you if you forgot how to use a command. An easy way to do this is to dump the last 100 commands you ran with the following command `history | tail -n 100 > my_commands.txt`

Done!

Our data has been cleaned, checked and formated so we are ready to start making ToAs and do some timing!

Key Points

- There are different types of pulsar data and different ways to process them
- Removing RFI can be difficult but will significantly improve your data
- Keeping your data organised will make your work easier

(/2023-09-25_NWU_Pulsar_Timing_Workshop/Setup/index.html)

(/2023-09-25_NWU_Pu