Introduction to PSRCHIVE

Willem van Straten & Paul Demorest

July 8, 2010

1 General introduction

The PSRCHIVE programs to be used as part of this tutorial print a brief help message when the `-h` command line option is used; e.g. `psredit -h`. Online reference manuals are also available at

`http://psrchive.sourceforge.net/manuals`

The PSRCHIVE software deals with observational data stored as a three-dimensional array of pulse profiles; the axes are time (sub-integration), frequency (channel), and polarization. The physical properties of the data are described by various attributes called meta-data. These characteristics are explored while introducing some basic features of PSRCHIVE that are of general use in pulsar data analysis, ending with a focus on those tools specific to pulsar timing. To begin the tutorial, log in to orion as the user pulsar, enabling trusted X11 forwarding; e.g.

```
ssh -Y pulsar@192.168.22.121
```

The data for this tutorial are located in `/home/pulsar/psrchive_data`. Please make your own copy of this sub-directory and set the `PSRDATA` environment variable to the full pathname to your copy. For example, if your name is Willem:

```
mkdir $HOME/willem
cp -r /home/pulsar/psrchive_data $HOME/willem
setenv PSRDATA $HOME/willem/psrchive_data
```

2 Viewing and editing meta-data with `psredit`

Please read **Section 2.0: Usage** at

`http://psrchive.sourceforge.net/manuals/psredit`

In the `$PSRDATA/0437/` directory, the receiver name is not set in any of the data files (`*.ar`). Set the receiver name to `MULT_1` using `psredit` and the *output option* to overwrite the original files.

```
cd $PSRDATA/0437
psredit -c rcvr:name=MULT_1 -m *.ar
```

The data files in `$PSRDATA/0437/` are a mixture of three different types. Create two sub-directories called `pulsar/` and `cal/` then use `psredit` to query the `type` attribute and use this to sort the files into the two sub-directories. (All three calibrator file types can go into the `cal/` sub-directory.)

```
cd $PSRDATA/0437
mkdir pulsar/
mkdir cal/
mv `psredit -c type *.ar | grep Pulsar | awk '{print $1}'` pulsar/
mv *.ar cal/
```

Note that the "back-tick" character ` is located on the keyboard on the same key as the "tilde" character ~. The expression inside the back-ticks is evaluated first.

3 Evaluating data with `psrstat`

Please read **Section 2.0: Usage** at

`http://psrchive.sourceforge.net/manuals/psrstat`

Use `psrstat` to print the signal-to-noise ratio S/N of each of the pulsar observations.

```
psrstat -c snr pulsar/*.ar
```

Note that, by default, `psrstat` computes the S/N of the profile in the first sub-integration, frequency channel and polarization (indexed by `subint`, `chan` and `pol`, respectively). Take one file and print the S/N in each frequency channel using the loop option. The number of frequency channels in the file can be queried with the `nchan` attribute.

```
psredit -c nchan pulsar/n2003200180804.ar
psrstat -l chan=0-127 -c snr pulsar/n2003200180804.ar
```

`psrstat` can be used in combination with other common tools to investigate problems and/or verify the quality of data. For example, use `psrstat` to print the S/N and the total power (`all:sum` attribute) as a function of frequency channel. (Hint: use the `-Q` command line option to print only the `value`, instead of `key=value`.) Redirect the output to a text file named `test.dat`

```
psrstat -Q -l chan=0-127 -c snr,all:sum pulsar/n2003200180804.ar > test.dat
```

and use `gnuplot` to plot these data as follows:

```
gnuplot> set xlabel "channel index"
gnuplot> set ylabel "total power"
gnuplot> plot "test.dat" using 4
```

and

```
gnuplot> set xlabel "channel index"
gnuplot> set ylabel "signal-to-noise"
gnuplot> plot "test.dat" using 3
```

and, finally

```
gnuplot> set xlabel "total power"
gnuplot> set ylabel "signal-to-noise"
gnuplot> plot "test.dat" using 4:3
```

Is the relationship between total power and S/N as might be expected?

4 Pre-processing data with the `psrsh` interpreter

Please read **Section 2.3 Job preprocessor** at

`http://psrchive.sourceforge.net/manuals/psrsh`

Note that the output of `psrsh -H` prints three effective columns. The first column is the command name, the second column is a single-letter short-cut key in square brackets, the third column is a short description of each command.

The S/N values output by `psrstat` in the previous section are for only one polarization, not the total intensity. This is because the four polarization parameters stored in these files describe the elements of the coherency matrix: AA, BB, R$e[AB]$, and I$m[AB]$. The total intensity, $I = AA+BB$ is formed by the `pscrunch` command. Use the preprocessing capability of `psrstat` to print the S/N of the total intensity as a function of frequency for one file.

```
psrstat -j pscrunch -l chan=0-127 -c snr pulsar/n2003200180804.ar
```

Add the `fscrunch` command to print the S/N of the total intensity integrated across the entire observing bandwidth for each file.

```
psrstat -j pscrunch,fscrunch -c snr pulsar/*.ar
```

Using single-letter short-cut keys, the above line is equivalent to

```
psrstat -j pF -c snr pulsar/*.ar
```

Redirect the output to a file (remember to add the `-Q` option) and plot the S/N as a function of time using `gnuplot`. Why does it vary?

5 Plotting data with `psrplot`

Please read **Section 2.0 Usage** at

`http://psrchive.sourceforge.net/manuals/psrplot`

Use `psrplot` to plot the phase-vs-frequency image of the total intensity of the pulsar signal in each file. (Remember: to plot total intensity, use `pscrunch` or its short-cut `p`).

```
psrplot -P
psrplot -p freq -j p pulsar/*.ar
```

Plot the phase-vs-frequency image of Re$[AB]$ (`pol=2`), which roughly corresponds to Stokes U, one of the two numbers that describe the linear polarization.

```
psrplot -p freq -c pol=2 pulsar/*.ar
```

Why might this quantity vary with frequency?

Use the loop over range option to plot the total intensity profile (the plot type named `flux` or its short-cut `D`) as a function of frequency in one pulsar data file.

```
psrplot -pD -l chan=0- -jp pulsar/n2003200180804.ar
```

Note that `chan=0-` specifies the entire range without having to know the index of the last frequency channel. Why does the pulse profile drift to the right (later pulse phases) as a function of frequency channel?

The pulse profile is significantly distorted at the edges of the band. This distortion is due to quantization error (also called "scattered power") that arises during analog-to-digital conversion using 2 bits/sample. The most-affected channels will be excised in Section 6.

Plot the Stokes parameters integrated over all frequency channels for each file.

```
psrplot -p stokes -jF pulsar/*.ar
```

Why do the Stokes parameters vary with time?

5.1 Customized plots (*optional*)

`psrplot` supports a wide range of options that enable convenient customization and production of publication-quality plots. For example,

```
psrplot -c set=pub
```

will set a number of attributes such as character font and size to values that are better suited to publication. It is also possible to control the labels that are printed above and below the plot. Please read

`http://psrchive.sourceforge.net/manuals/psrplot/label.shtml`

and produce a publication-quality plot with the S/N printed in the top-right corner of the plot (inside the plot frame).

```
psrplot pulsar/n2003200180804.ar -pD -jFp -c below:r='SNR: $snr' -c set=pub
```

Note that filename(s) need not necessarily be the last argument(s) on the command line.

6 Excising invalid data with `paz`

Please read **Section 2.0 Usage** at

`http://psrchive.sourceforge.net/manuals/paz`

then use `paz` to assign zero weight to those frequency channels most affected by 2-bit quantization distortions. Use the *output option* to write output data files with a new extension; e.g.

```
paz -Z 0-19 -Z 108-127 -e zz pulsar/*.ar cal/*.ar
```

7 Interactive excision with psrzap

`psrzap` is an interactive tool for excising data corrupted by radio frequency interference (RFI). Run `psrzap -h` for a list of the keyboard and mouse interactive commands, then

```
cd $PSRDATA/1909
psrzap guppi_55245_1909-3744_0033_0001.rf
```

After a short wait, the dynamic noise spectrum will be plotted, given by the variance as a function of time sub-integration (x-axis) and frequency channel (y-axis). There are three modes of selecting ranges of data to view or excise:

1. time (`t` on keyboard) selects an entire column
2. frequency (`f` on keyboard) selects an entire row
3. both (`b` on keyboard) selects a rectangular region

The line(s) passing through the cursor indicate the current selection mode. To zoom in on a desired range, click the left mouse button at the start and end positions of the range. To excise a desired range, click the left mouse button at the start, and the right mouse button at the end of the range. Simply right clicking will excise the single column, row, or pixel under the mouse (depending on the selection mode).

RFI typically appears as bright spots in the dynamic noise spectrum. Excise data until you are satisfied and save the result by pressing `s` on the keyboard. Use `psrstat` to compare the S/N of the data before and after RFI excision. (Integrate over all of the sub-integrations using the `tscrunch` pre-processing directive.)

```
psrstat -jTFp -c snr guppi_55245_1909-3744_0033_0001.rf*
```

8 Calibrating data with pac

A first-order approximation to calibration can be attempted using the calibrator observations of type `PolnCal` to derive the complex gains of the instrumental response as a function of frequency. In this approximation to calibration, the Jones matrix in each frequency channel has the form

$$\mathbf{J} = \begin{pmatrix} z_0 & 0 \\ 0 & z_1 \end{pmatrix} \tag{1}$$

where z_0 and z_1 are the complex gains. The absolute phase of the Jones matrix is lost during detection, and the matrix may be parameterized using a polar decomposition described by the absolute gain G, differential gain γ, and differential phase ϕ.

8.1 Plotting calibrator data with `pacv`

The polar decomposition parameters can be plotted with the program `pacv` (no online manual available). Simply type `pacv` *filename* to see the parameters derived from any of the observations in the `cal/` sub-directory.

```
cd $PSRDATA/0437/cal
pacv *.zz
```

Why does differential phase vary approximately linearly with frequency?

8.2 Preparing flux calibrator data with `fluxcal`

Although not immediately necessary for pulsar timing, it is also useful to perform absolute flux calibration, which may be later useful in other studies that could have an impact on long-term timing analyses; e.g. studies of refractive scintillation.

Absolute flux calibration information is derived from observations made while pointing at or near a standard candle such as Hydra A; these observations have a `type` attribute equal to `FluxCalOn` or `FluxCalOff`. Start by creating a calibrator database. In the `cal/` sub-directory, run

```
pac -w -u zz
```

to create a file called `database.txt`. Then run

```
fluxcal -f -d database.txt
```

This will produce a file named `n2003201035947.fluxcal`, which can be viewed with `pacv`. It will also have updated `database.txt` with a new entry for this file.

8.3 First-order calibration

To perform the first-order approximation to calibration, run

```
cd $PSRDATA/0437
pac -d ../cal/database.txt *.zz
```

For each input file, a new output file will be created with a new extension, `.calib`. If the receiver were ideal, the first-order approximation to calibration would have eliminated the variation of the Stokes parameters as a function of parallactic angle. Use `psrplot` to test this expectation.

```
psrplot -ps -jF *.calib
```

Why do the Stokes parameters still vary as a function of time?

Create a sub-directory, e.g. `ideal/` and move the newly calibrated data to this sub-directory (otherwise, they will be over-written in the next step).

```
mkdir ideal
mv *.calib ideal/
```

8.4 Measuring the cross-coupling parameters with `pcm`

To properly calibrate these data, the cross-coupling terms (off-diagonal components of the Jones matrix) must be estimated, which can be done by modeling the Stokes parameters as a function of time. The process of performing the least-squares fit is called Measurement Equation Modeling (MEM) and the PSRCHIVE implementation is described in van Straten (2004, ApJS 152:129).

Please read **Section 2.1 Before running `pcm`**, **Section 2.2 Running `pcm`**, and **Section 4.1 MEM Example** at

`http://psrchive.sourceforge.net/manuals/pcm`

then use `psradd` to combine the first-order calibrated archives into a single archive (as described in **Section 2.1**)

```
psradd -T -o calib.TT ideal/*.calib
```

and run `pcm` (as described in **Section 4.1**) to derive the cross-coupling parameters, which are output in a file named `pcm.fits`.

```
pcm -d ../cal/database.txt -s -c calib.TT *.zz
```

What is the `-s` option and why is it used?

It will take about ten minutes to perform the least-squares fit in each frequency channel. If you are running on `lclinux`, it is best to stop `pcm` (with Ctrl-C) when it slows down and starts printing messages about the goodness of fit; e.g.

```
  reduced chisq 1.26665282249451
  reduced chisq 1.24342453479767
```

If you are running on another machine and would like to run `pcm` to completion, please ignore error messages like

```
Failed to add data 7 out of 12288 times for phase bin 1013

Calibration::StandardModel::solve failure
        Singular Matrix.  irow=79 nrow=79
```

The latter error message should occur only if the invalid data in frequency channel 0 have not been previously excised.

Finally, if you are running on a machine with more than one CPU/core, you can speed up `pcm` by running on more than one thread using the `-t <nthread>` option, where `<nthread>` is the number of cores to be used.

The output file `pcm.fits` contains the solution output by `pcm`. (This file was prepared previously for those who stopped with `Ctrl-C`). Plot the Jones matrix parameters using

```
pacv -P pcm.fits
```

Note that only 3 new parameters have been added to the model: θ_1 describes the non-orthogonality of the feed receptors (the linearly polarized receptors should be oriented at 0 and 90 degrees); ϵ_k are the ellipticities of the receptors, which should be 0 in an ideal feed with linearly polarized receptors. The mean value of ~ 5 degrees corresponds to roughly 15% mixing between linear and circular polarizations (Stokes Q and V). Note that `pcm` cannot determine the absolute rotation of the receptors about the line of sight, θ_0, without an external reference; therefore, only the non-orthogonality is measured.

8.5 Full calibration

Move the file `pcm.fits` to the `cal/` sub-directory, change to this directory, and recreate the calibrator database

```
mv pcm.fits ../cal/
cd ../cal
pac -w -u zz -u fits -u fluxcal
```

Confirm that `pcm.fits` has been added to `database.txt`, then run

```
cd ../pulsar
pac -d ../cal/database.txt -S *.zz
```

Plot the Stokes polarization profile (integrated over the entire band) in each calibrated data file output by `pac` to confirm that they no longer vary with time. Choose one file and plot the phase-vs-frequency image of $\mathrm{Re}[AB]$ ... does this value still vary with frequency as before?

9 Producing arrival time estimates with `pat`

In this section, arrival time estimates are derived using a previously created standard (template) profile. This high S/N standard profile was formed by adding data from 5 different days (about 42 hours of integration).

9.1 Preparation of the standard profile

The standard profile is located in `$PSRDATA/0437/std/standard.ar`. Plot the phase-vs-frequency image of the total intensity and compare this image with that of `nonspc.ar` in the same directory. The file `nonspc.ar` was formed from data that were not corrected for scattered power. Although `standard.ar` was corrected, there still remain residual artifacts in the edges of the band. Use `paz` to give zero weight to the affected frequency channels. For best results, excise the same frequency channels that were excised from the data in Section 6.
Use `pam` to integrate over all frequency channels; e.g.

```
pam -F -e FF standard.zz
```

then use the wavelet smoothing algorithm implemented by `psrsmooth` to create a "noise-free" template profile

```
psrsmooth -W -t UD8 standard.FF
```

This will produce a file called `standard.FF.sm`. Use the `crop` attribute of the `flux` plot to zoom in on the low amplitude flux near the off-pulse baseline and compare the standard profile with its smoothed version; e.g.

```
psrplot -pD -jp -c crop=0.01 -N 1x2 standard.FF standard.FF.sm
```

9.2 Estimation of arrival times

In this tutorial, we will compare two different methods of arrival time estimation, both of which work in the frequency domain: scalar template matching using only the total intensity, and matrix template matching using the full-polarization profile.

We also have two standards with which to experiment: the smoothed and not smoothed versions of `standard.ar`.

Finally, there are three different data sets: the uncalibrated data, the data calibrated using the first-order approximation, and the data calibrated using the solution derived with `pcm`.

All together, there are 12 combinations of arrival time estimation algorithm, standard profile, and data. Experiment with these combinations to find the arrival times with the lowest residual standard deviation.

To experiment, you will need to run `pat` in either scalar template matching mode; e.g.

```
pat -F -s standard.FF *.zz > uncalibrated_unsmoothed_stm.tim
```

or matrix template matching mode

```
pat -F -p -c -s standard.FF.sm *.calib > calibrated_smoothed_mtm.tim
```

You will also need to run tempo2 to evaluate the arrival times; e.g.

```
tempo2 -f pulsar.par calibrated_unsmoothed_stm.tim
```

Search the output of `tempo2` for lines like

```
RMS pre-fit residual = 0.108 (us), RMS post-fit residual = 0.108 (us)
Fit Chisq = 359.7       Chisqr/nfree = 359.74/95 = 3.78671      pre/post = 1
```

and make note of both the **RMS post-fit residual** and **Chisqr/nfree** in each case tested.

9.3 Discussion

Form groups to compare the arrival time analyses and consider questions such as

- Why is the reduced χ^2 so far from unity in each case?
- What is the effect of using the smoothed standard profile?
- Does the smoothed standard profile have the same effect on both scalar and matrix template matching?
- What is the **last harmonic** reported by the matrix template matching algorithm?
  (Hint: plot the *Stokes; fluctuation power spectra* of the standard profile using `psrplot`)
- Is it necessary to calibrate the data before applying matrix template matching?
- Does calibration always improve the arrival time precision?