Student Week - Aug 06, 2022

DMCalc Tutorial

Pratik Tarafdar

Materials

Available at
https://github.com/inpta/InPTA-student-week-2022/tree/main/DMCalc_Tutorial

CONTENTS

- Data files
- DMCalc script

Prerequisites - I

1. Activate the pulsar environment: $ conda activate pulsar

2. Crosscheck for python3 or python3.* executables inside
   $HOME/anaconda3/envs/pulsar/bin/
   This path should match with the path in the first line of
   /DMCalc_Tutorial/scripts/dmcalc_dr1.py, and
   /DMCalc_Tutorial/scripts/run_dmcalc_dr1.py

3. If all is well, you should be able to see the help menu on running the following commands from inside the scripts directory -
   $ python3.9 dmcalc_dr1.py -h
   $ python3.9 run_dmcalc_dr1.py -h

Prerequisites - II

1. DMCalc makes use of various PSRCHIVE functions. We shall need to use some of these functions during the template preparation phase even before invoking DMCalc. PSRCHIVE comes with some default configurations which require certain changes before application to InPTA data.

2. Check if you have a file named .psrchive.cfg in your home area.
   (Just do a $ ls ~/.psrchive.cfg in your terminal if it is there)

3. If not there, type the following in your terminal -
   $ psrchive_config > $HOME/.psrchive.cfg
   This will create the configuration file inside your home area.

Prerequisites - III

1. Open the .psrchive.cfg file using your favourite text editor (vim, nano, gedit, notepad, atom..) and uncomment (remove ‘#’) the following lines and make the following changes -

   Archive::unload_class = PSRFITS
   ArrivalTime::default_format = Tempo2
   Dispersion::barycentric_correction = 1
   FrequencyAppend::force_new_predictor = 1
   Predictor::default = tempo2
   WeightedFrequency::round_to_kHz = 0

Cross-checking the data files

1. Navigate to the DMCalc_Tutorial directory

2. Run psredit on the data files (*.fits) inside data/BAND* directories -
   $ psredit -c name,nbin,nchan,nsubint,freq,bw data/BAND*/*.fits

3. This should display the following information contained in the data file headers -
   Name of the pulsar, #bins in pulse phase, #subbands in freq., #subintegrations in time, central freq. of the band and the bandwidth

4. Ensure that the name and central frequencies are correct (~399 MHz for Band 3 300-500 MHz and ~1360 MHz for Band 5 1260-1460 MHz). Check that the no. of subintegrations is 1 (the data is integrated in time, or ‘time-collapsed’) and that the bandwidth is +200 MHz.

Identifying the template epoch

1. Templates - standard, clean pulsar profiles curated for cross-correlation with other profiles to measure pulse arrival times, dispersion measures and other quantities.

2. We prepare templates from the observed data by selecting a high S/N epoch (in all observed frequency bands). The profile should be a standard one - visually check for any loss or artefacts across time, frequency or phase, or scintillations across frequency etc. before finalizing the epoch for template preparation.

3. One way (not the best though) to estimate the S/N is using ‘psrstat’ package from PSRCHIVE. Navigate to the ‘data’ folder and run the following command -
   $ psrstat -jDF -c “snr=pdmp” -c snr BAND*/*.fits

4. Note the epochs with the highest S/N in BOTH bands. If the highest S/N in both bands belong to different epochs, come to a logical compromise based on the other checks.

Identifying the template epoch

1. For visual checks, try out to following commands -

   $ psrplot -jDF -p flux <filename>.fits (for the integrated profiles)
   $ psrplot -jDF -p time <filename>.fits (for intensities across time)
   $ psrplot -jD -p freq+ <filename>.fits (for intensities across frequency)

2. Once the template epoch is identified, we proceed to prepare the template.

Template generation - I

1. Extract the pulsar ephemeris from the BAND 3 template epoch data (inside the ‘std’ folder) by
   $ vap -E *_500.*.fits > J1909-3744.par

2. Reverse the order of frequency channels in the template epoch data files using -
   $ pam -m --reverse_freqs *.fits

3. Create time-collapsed versions of the fits files in both bands -
   $ pam -e T.fits -T *rfiClean.fits

4. Create versions with 3 sub-integrations in both bands -
   $ pam -e 3.fits --setnsub 3 *rfiClean.fits

5. Estimate an initial DM using ‘pdmp’ on the BAND 3 time-collapsed data from the template epoch -
   $ pdmp -g /NULL *_500.*T.fits

6. Look at the row corresponding to “Best DM” in pdmp output
   If DM correction > DM error, replace the DM inside the par file with the Best DM and DMEPOCH with the template epoch.
   If DM correction < DM error, replace only the DMEPOCH inside the par file with your template epoch.

Template generation - II

1. Apply this par file to both the *T.fits and *3.fits data files -
   $ pam -m -E J1909-3744.par --update_dm *T.fits *3.fits

2. Dedisperse the *T.fits files using this updated DM in their headers -
   $ pam -m -D *T.fits

3. Smoothen the *T.fits files using ‘psrsmooth’ -
   $ psrsmooth -W -t UB103 *_500.*T.fits
   $ psrsmooth -W -t UB103 *_1460.*T.fits
   It will create a smoothed profile with extension *.sm
   ‘UB103’ is a wavelet transform for smoothing profiles. It works best for this template. But other wavelet transforms are also available and should be considered.
   The full list can be obtained using $ psrsmooth -h=t
   The default transform (only -W option) is UD8. We can also try UB301, UB303 etc.
   The optimum transform is selected by checking the increase in S/N (remember the ‘psrstat’ command?) and also a visual check of the smoothed profiles - such that it doesn’t get oversmoothed thus changing the profile significantly - using ‘psrplot’ commands.
   $ psrplot -jDF -p flux -N 1x2 *_500.*T* for BAND 3 or *_1460.*T* for BAND 5

Template generation - III

1. The smoothed profiles (*.sm) are our initial templates. Let us rename them for convenience, say -
   $ mv *_500.*.sm J1909-3744_b3_200.std
   $ mv *_1460.*.sm J1909-3744_b5_200.std

2. We shall now refine our DM estimate for the template with a little bit of pulsar timing! (remember last week's tutorial by Manjari and Jaikhomba?)
   We generate TOAs (Times Of Arrival) from the template epoch data itself with 3 sub-integrations by comparing with our initial templates (the *.std files created above)

3. At first, only BAND 3 TOAs for a first order estimation of DM -
   $ pat -j"F 64" -A FDM:mcmc=1 -f tempo2 -P -s *_b3*.std *_500.*3.fits > J1909-3744.tim

4. Plot the residuals using 'tempo2' -
   $ tempo2 -us -gr plk -f J1909-3744.par J1909-3744.tim -nofit

5. Note the pre-fit reduced chi-sqr and Wrms values of the residuals. We would like to bring the value of reduced chi-sqr closer to '1' and decrease the value of Wrms by fitting for some basic pulsar parameters

Template generation - IV

1. Save the fitted model to new par file (J1909-3744.fit.par)
   Apply this fitted par file to the *3.fits files in both bands
   $ pam -m -E J1909-3744.fit.par --update_dm *3.fits

2. Apply the fitted par file to the templates in both bands
   $ pam -m --DD -E J1909-3744.fit.par --update_dm *.std
   (Note the --DD option. It stands for ‘de-dedisperse’)

3. Dedisperse the templates again, but w.r.t. the fitted DM
   $ pam -m -D *.std

4. A little more timing! - Use the slightly improved templates to generate TOAs again but for both bands
   $ pat -j”F 64” -A FDM:mcmc=1 -f tempo2 -P -s *_b3*.std *_500.*3.fits > J1909-3744.tim
   $ pat -j”F 16” -A FDM:mcmc=1 -f tempo2 -P -s *_b5*.std *_1460.*3.fits >> J1909-3744.tim
   (Note the ‘>>’ operator in the second command. It appends BAND 5 TOAs to the same file. Otherwise the BAND 3 TOAs will be overwritten)

5. Plot the residuals and do some fitting again with ‘tempo2’ using the fitted par file
   $ tempo2 -us -gr plk -f J1909-3744.fit.par J1909-3744.tim -nofit

Template generation - V

1. Save the fitted model again to a new par file (J1909-3744.fit.par can be overwritten)

2. Apply the fitted par file to the templates in both bands
   $ pam -m --DD -E J1909-3744.fit.par --update_dm *.std

3. Dedisperse the templates again, but w.r.t. the fitted DM
   $ pam -m -D *.std

4. These new *.std files are our final templates that will be used by DMCalc for the DM estimation of any other epoch

DM estimation with DMCalc - I

1. Replace the DM value in your previous J1909-3744.par file with the final fitted DM value inside J1909-3744.fit.par

2. Come out of the ‘std’ directory and apply this par file to all the data files whose DMs we would like to measure -
   $ pam -m -E std/J1909-3744.par --update_dm data/BAND*/*.fits

3. Create a working directory for the DMCalc run (say ‘dmcalc’) and navigate to that folder

4. Run DMCalc on both bands of the epoch whose DM we would like to measure
   $ python3.9 ../scripts/dmcalc_dr1.py -E ../std/J1909-3744.par -M ../std/J1909-3744_b3_200.std ../std/J1909-3744_b5_200.std -b3n 8 -b5n 4 ../data/BAND3/J1909-3744_59590.303334_500.rfiClean.fits ../data/BAND5/J1909-3744_59590.303184_1460.rfiClean.fit

5. The TOAs are written to J1909-3744_allToAs.tim and DMs from only BAND3 TOAs, only BAND5 TOAs and BAND3+5 combined TOAs are written to J1909-3744_DM_timeseries.txt

DM estimation with DMCalc - II

1. Data columns in the DM timeseries file represent the following quantities -
   MJD, DM, DM_error, Reduced_chisq, prefit_rms, postfit_rms, med_toaE, centre_freqs, bandwidth, telescope, band_flag

2. Suppose you want to estimate DMs for a large number of epochs. Running DMCalc separately for each epoch would be a real horror!

3. To make life simpler, you can use a python wrapper called ‘run_dmcalc_dr1.py’ also available inside the ‘scripts’ directory

4. Remember that fresh DMCalc runs should always be initiated in fresh empty directories. So either remove previous files from the ‘dmcalc’ directory or create a new working directory.

5. Run DMCalc wrapper on both bands of all the epochs whose DMs you would like to measure
   $ python3.9 ../scripts/run_dmcalc_dr1.py ../std/J1909-3744.par ../std/J1909-3744_b3_200.std ../std/J1909-3744_b5_200.std 8 4 ../data/BAND3/ ../data/BAND5/

6. The TOAs are again written to J1909-3744_allToAs.tim and DMs from only BAND3 TOAs, only BAND5 TOAs and BAND3+5 combined TOAs for all epochs are written to J1909-3744_DM_timeseries.txt

Try the analysis on your own.

Perform each step, let it sink in, think and explore the various options for each package being used.

Get in touch for any queries.

Happy Data Crunching!