A
Manual
For
GMRT Pulsar Timing Analysis
Pipeline.

By

Sapan Kumar Sahoo

Guided By

Bhaswati Bhattacharyya

NCRA • TIFR

Pulsar Data Timing Analysis Pipeline Manual

Overview

Here is a comprehensive manual for the pulsar data timing analysis pipeline. Below is a detailed manual for the pulsar data timing analysis pipeline.

Requirements

Before utilizing the pipeline, ensure the following requirements are met:

- ❖ Installation of gptool [1] for RFI mitigation.
- ❖ Matplotlib for plotting the profile.
- ❖ PRESTO:- https://github.com/scottransom/presto
- ❖ TEMPIO2:- https://bitbucket.org/psrsoft/tempo2

Link for pipeline: - https://github.com/sapankumarsahoo/Pulsar-TIming-Pileline

Introduction

Welcome to the Pulsar Data Timing Analysis Pipeline manual. This manual will guide you through the process of analyzing pulsar data for timing study by using the provided Bash script pipeline. The pipeline includes multiple steps aimed at generating integrated profiles, plotting profiles and multiple Gaussian fits into it, calculating Time of Arrivals (ToAs), and handling in-band sub-banding and correction for Dispersion Measure (DM) variation across it.

Script Overview

The provided script offers several options for analyzing pulsar data. The script starts by sourcing necessary bashrc files to ensure that environment variables are properly set.

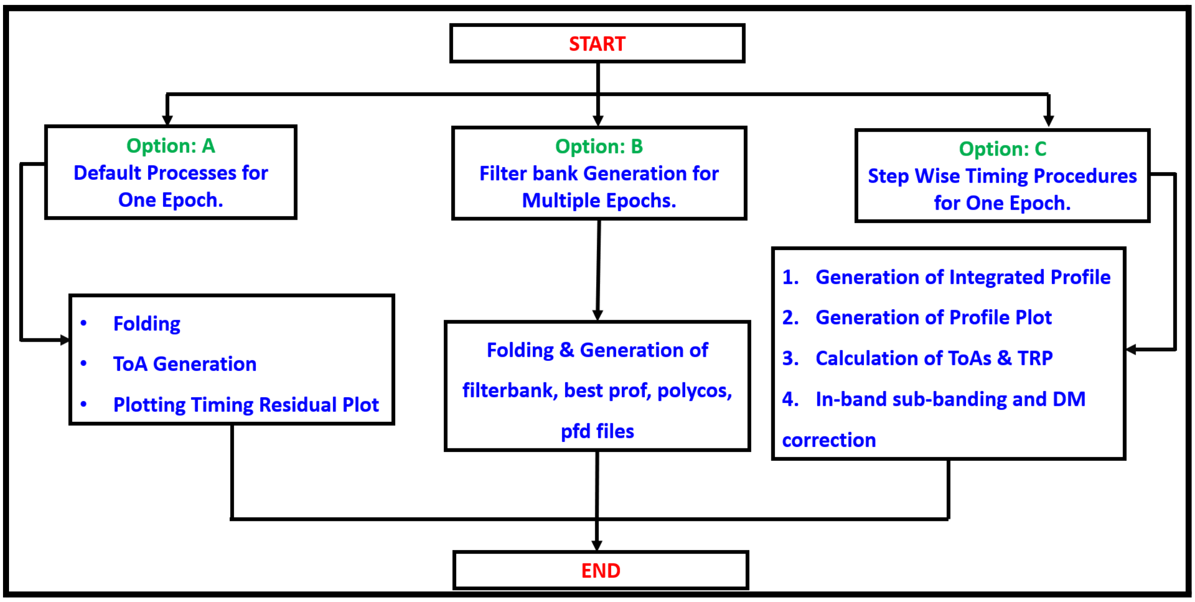
START
Option: A
Default Processes for One Epoch.
Option: B
Filter bank Generation for Multiple Epochs.
Option: C
Step Wise Timing Procedures for One Epoch.
• Folding
• ToA Generation
• Plotting Timing Residual Plot
Folding & Generation of filterbank, best prof, polycos, pfd files
1. Generation of Integrated Profile
2. Generation of Profile Plot
3. Calculation of ToAs & TRP
4. In-band sub-banding and DM correction
END

Figure 1:- Structure of Pipeline showing various options at different steps.

Main Menu:

Users are presented with three options:

a. Option A: Default Procedure For One Epoch
b. Option B: Filterbank File Generation For Multiple Epochs
c. Option C: Step Wise Timing Procedure For One Epoch

Depending upon the users' input, it will select a specific task.

1. Option A: Default Procedure For One Epoch:
   a. Users are prompted to input details such as date, location of raw and timestamp files, frequency, number of channels, and mode (I or C) or details are extracted from the “file_details” file and output for analysis.
   b. The script then proceeds to execute a command (auto_fold_GWB_GPTOOL_FREQ_OFFSET.csh) with the given parameters.
2. Option B: Filterbank File Generation For Multiple Epochs:
   a. Users are prompted to enter details for multiple epochs of information, such as date, location, frequency, bandwidth, and mode.

    b. Details are written to a file named “file_list”.
    c. Users have the option to repeat this process for multiple epochs.
    d. After input, the script executes a command for processing the “file_list” to generate the filter bank file.
3. Option C: Step Wise Timing Procedure For One Epoch:
    a. Users are presented with multiple sub-options for different processes related to timing analysis.
        i. Generation of Integrated Profiles
        ii. Generation of Profile Plot
        iii. Calculation of Time of Arrivals (ToAs) & producing Timing Residual Plot (TRP)
        iv. In-band Sub-banding and DM Correction
    b. Depending on the chosen sub-option, the script performs various tasks such as generating filterbank files, profile plots, and calculating TOAs (Time of Arrivals).
    c. Additionally, users are allowed to make adjustments and corrections during the process.

Based on the selected task, the script executes relevant commands to accomplish the required analysis.

Detailed Manual

Step 1: Generation of Integrated Profiles or Folding

This step involves generating integrated profiles from raw pulsar data.

The script prompts the user to input relevant parameters as listed below,

- Location of the raw file, parameter file, and Frequency (MHz), Bandwidth (MHz) of observation.
- For coherently de-dispersed data (CD data):-
    - Date input format: 25Apr2k24
    - Mode: C
- Incoherently de-dispersed data (PA data):-
    - Date input format: 25Apr2024
    - Mode: I

After taking the inputs, it will process RFI mitigation, bad channel removal, or noise removal by using gptool and it will do the de-dispersion process by using the PRESTO command line.

After producing the above file, it will allow the user to account for any corrections (if necessary).

In this process, filterbank files, Integrated profiles (best prof file, ps file), polycos, pfd file (contain information about the data cubes) are generated and saved for further analysis.

Step 2: Generation of Profile Plot

This step involves plotting the generated profiles to account for profile evolution.

The script processes the profile file (pfd file), averages frequency and time using Pam, and then plots the profiles using matplotlib.

The user can visualize the profile evolution for analysis, do apply science accordingly.

After the profile plot, it will do the multi-gaussian fit into it. It will use curve_fit, find_peaks to do the multi-gaussian fit, and do the plotting. Simultaneously, it will also print the fitting parameters like values of sigma, center position, and peak amplitude for various fitted peaks.

Step 3: Calculation of Time of Arrivals (ToAs) & Producing TRP

This step calculates the Time of Arrivals (ToAs) from the generated profiles.

ToAs are the time of arrival of the signals at the telescope (Further transfer to the Barycentric center of the Solar System Frame)

The script prompts the user to input relevant parameters as listed below

1. It will also ask for the location (including name) of the template profile.
2. The script prompts the user to input the epoch's location of necessary files (e.g., pfd file, best prof, polycos).

3. Parameter file location with location

Then the ToAs and their uncertainties are calculated using “get_TOAs.py” and automatically updated in the final timing file.

It will produce multiple numbers of ToAs for the same epoch like 1, 2, 3, 4, 5, 6, 10, 12, 15, 30, 60 (multiple of 60) number of ToAs and ask to choose one number to retain the ToAs information.

Then with the tempo2 command line, it will generate the TRP for different numbers of ToAs, and allow users to modify each step.

Step 4: In-band Sub-banding and DM Correction

This step involves sub-banding and correction for Dispersion Measure (DM) variation inside the bandwidth of the observation.

The script performs sub-banding and DM correction for each profile file.

The process involves the following steps:-

1. It will divide the present band into 10 or 12 sub-bands.
2. It will generate corresponding ToAs by following step 3
3. It will create a timing file that includes the above-produced information.
4. It will show the TRP with the tempo2 command line.
5. To account for the correct DM, you have to fit the DM manually.
6. It will take the corrected DM value as an input, it will repeat the above process by adding an extra column “ -dm DM_value “ in the timing file.
7. Again, it will produce the TRP with a modified timing file.

It generates a final timing file with updated DM values for further analysis.

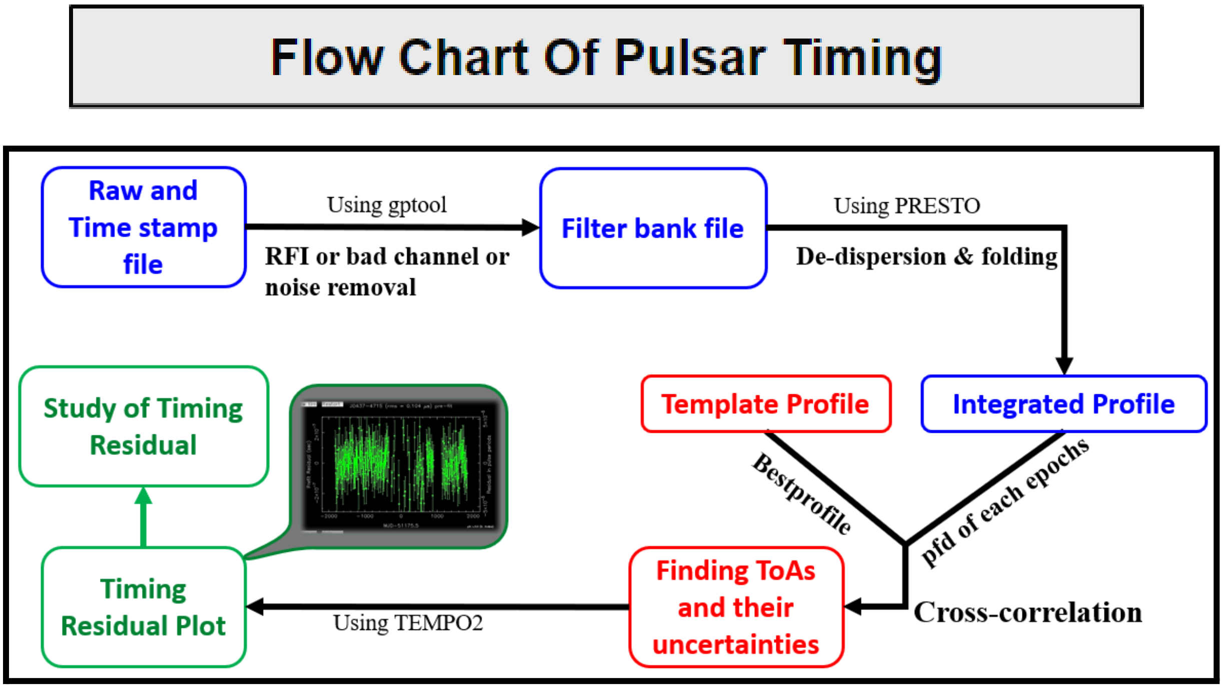

Figure 2:- Showing the overall procedure to produce a Timing Residual Plot.

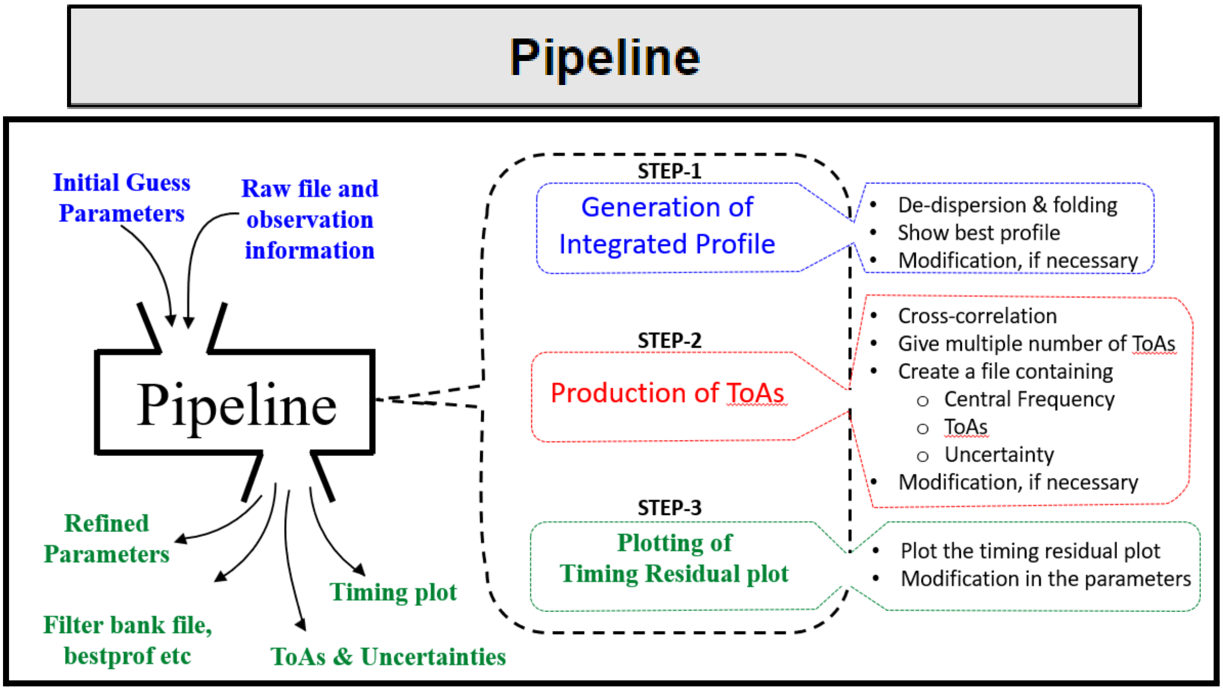
Pipeline
Initial Guess Parameters
Raw file and observation information
Pipeline
Refined Parameters
Filter bank file, bestprof etc
ToAs & Uncertainties
Timing plot
STEP-1
Generation of Integrated Profile
De-dispersion & folding
Show best profile
Modification, if necessary
STEP-2
Production of ToAs
Cross-correlation
Give multiple number of ToAs
Create a file containing
Central Frequency
ToAs
Uncertainty
Modification, if necessary
STEP-3
Plotting of Timing Residual plot
Plot the timing residual plot
Modification in the parameters

Figure 3:- Showing the overall steps included inside the pipeline.

Conclusion

Congratulations! You have completed the Pulsar Data Timing Analysis Pipeline using the provided Bash script. This manual should serve as a comprehensive guide for analyzing pulsar data and extracting relevant parameters for further study.

For further assistance or troubleshooting, please refer to the documentation of individual commands used in the script or consult with experts in the field of pulsar data analysis.

References

1. gptool (https://github.com/chowdhuryaditya/gptool)
2. PRESTO (https://github.com/scottransom/presto)
3. TEMPO2 (https://bitbucket.org/psrsoft/tempo2)
4. Reference for get_ToAs.py