

# pinta Version 6 User Guide

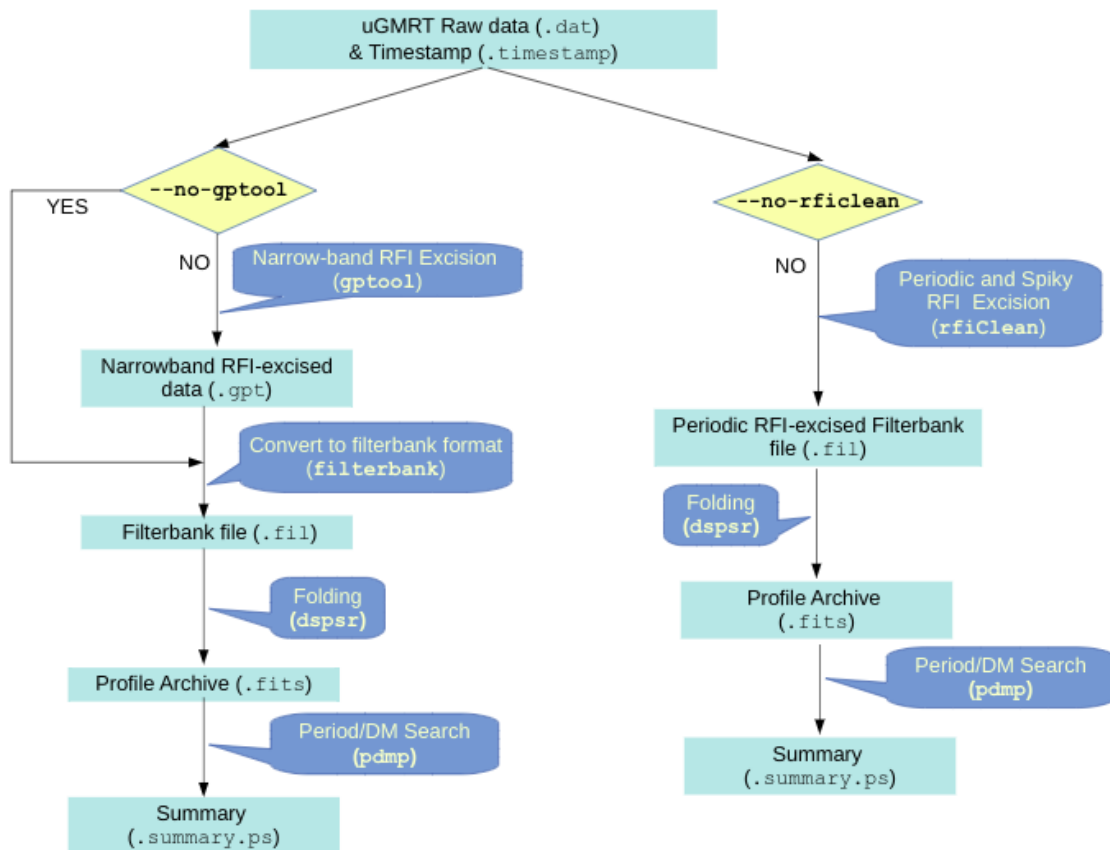
This pipeline is currently installed in the GMRT machine beamdata(fs4) and this is the user guide for the latest version (V6) of pinta.

More details can be found in the pinta paper: <https://doi.org/10.1017/pasa.2021.12>.

This SOP corresponds to commit [d7b57c301ec269ecf55ffa2de2fa55a7cf8d9a2f](https://github.com/beamdata/pinta/commit/d7b57c301ec269ecf55ffa2de2fa55a7cf8d9a2f).

## 1. Overview

The pipeline takes uGMRT raw data and creates folded profiles in the TIMER format and/OR PSRFITS format. The process is illustrated in the following flowchart.



## 2. Input Files

This process requires the following input files:

1. Raw data file (.dat or .raw) -- uGMRT raw data. This is a binary non-human readable file with its size in tens of GB. The size of this file is given by  
$$\text{Size} = 2 \text{ bytes} * \text{npol} * \text{nchan} * \text{Tobs} / \text{Tsample}$$
2. Timestamp file (.timestamp or .hdr) -- Contains the timestamp of the start of observation. Each raw data file will have a corresponding timestamp file. This is a small human-readable ASCII file.
3. gptool configuration file (gptool.in) -- Contains configuration information for gptool. Each uGMRT band will have its own gptool.in file. Ideally, this file should be optimized for each observation cycle. **Using the wrong gptool.in file gives wrong results with gptool. USING THE DEFAULT VALUE IS NOT RECOMMENDED. ALWAYS EXPLICITLY SPECIFY THIS OPTION. NOTE THAT THESE ARE NEW SETTINGS for some GMRT project and may not apply to ALL GMRT data.**
4. Pulsar ephemeris (.par) -- This file contains the pulsar ephemeris in TEMPO2 format. It is used by dspr to fold the data. This file can be generated from pulsar catalog using ***psrcat -e J0534+2200 > J0534+2200.par***  
These files should be updated if user has better timing solutions.
5. rfiClean Configuration file -- Contains flags and command-line options required to run rfiClean for pinta.
6. Pipeline input and configuration file (pipeline.in) -- This file contains all the metadata required for processing.
7. pinta configuration file (pinta.yaml) -- Contains the settings required by pinta.

## 3. pinta Configuration File

This file contains the following settings:

1. **Pulsar ephemeris directory** -- All the par files are stored in this directory. Each pulsar will have one par file. These files can be generated using the following command or users can put their own updated ephemeris  
***psrcat -e Jmmmm+dddd > Jmmmm+nnnn.par***  
Where Jmmmm+nnnn is the Jname of a pulsar
2. **gptool.in configuration directory** -- The gptool.in file for each frequency is stored here. By default this directory is  
**Using the default is now deprecated. The user must explicitly give the gptool.in conf directory using the --gptdir option.**
3. **rfiClean configuration file** -- Described above.

## 4. Directories

Here we define the various directories relevant for running pinta.

1. **Input directory** -- The raw data and timestamp files are stored here. An input directory can contain data from multiple days, multiple bands and multiple pulsars. Each raw data file must be accompanied by a corresponding timestamp file. Requires read permission.
2. **Working directory** -- The pipeline output (namely, the PSRFITS profile) and intermediate files (gpt, fil) are written into this directory. This directory must contain the pipeline.in file for the pipeline to start. Requires read and write permissions.
3. **Current directory** -- This is the directory from which the pipeline is invoked. Requires write permission.
4. **Script directory** -- The directory where the pinta script is stored. The pinta configuration file is also stored in the same directory.
5. **gptool Configuration directory** -- Described above. Requires read permission.
6. **Pulsar ephemeris directory** -- Described above. Requires read permission.

## 5. Instructions for running the pipeline

### 5.1. Step-by-step

1. Log in to beamdata/fs4.gmrt.ncra.tifr.res.in using ssh -XC from ncra or through wm.
2. Change the group.
  - a. **The pipeline will not work if the group is not pulsarg.**  
`newgrp pulsarg`
3. Source the bashrc files.
  - a. This should be done **after** changing the group  
`bash`  
`source /home/bcj/PSRSOFT/pulsarV6rc`
4. Change directory to your scratch data directory  
`cd /Data/jsingha (example)`
5. Identify the input files (should all be in the same directory) and the working directory.
  - a. **It is recommended that the input and working directories be different.**
6. Ensure that the working directory has enough free disk space available.
  - a. **Otherwise, the pipeline will crash.**
  - b. Typically ~3 times the raw data size per observation.
7. It is now **required** that the user not use the default gptool.in files.
  - a. **The gptool.in directory should be explicitly given using the --gptdir option.**
  - b. **The parfile should be explicitly specified using the --pardir option.**
  - c. **The gptool.in files for bands 5, 4 and 3 should be named gptool.in.1459, gptool.in.749, and gptool.in.499.** This is a historical idiosyncrasy and will be corrected in the future.
8. \*\* If you wish to use your own par files, copy all of them to a directory with appropriate file names (JNAME.par).
9. Create the pipeline.in file in your working directory.

10. Run the pipeline from a location where you have write permission (preferably the working directory).
  - a. You can use nohup to avoid the pipeline being killed if the ssh session is disconnected.
  - b. Check the logs to ensure that the process has finished successfully.
  - c. The logs from the component commands are written to the log/ directory inside the working directory. The logs from pinta are written to stdout.
11. Do sanity checks on the output products (see the sanity check SOP)

**Details on these steps are given in the following sections.**

## 5.2. Logging In

The pipeline can be run as follows after logging in to Tapti and then to fs4:

```
$ newgrp pulsarg
$ bash
$ source /home1/bcj/PSRSOFT/pulsarV6rc
```

Install the following once after logging in :

**(THIS MAY BE NEEDED TO BE DONE ONLY THE FIRST TIME BUT IF THESE PACKAGES ARE ALREADY INSTALLED IN YOUR HOME DIRECTORY, THIS STEP MAY NOT BE NEEDED )**

```
$ pip3.7 install numpy --user
$ pip3.7 install parse --user
$ pip3.7 install astropy --user
$ pip3.7 install pyyaml --user
```

## 5.3. Running the Pipeline

**pipeline.in file should be created in the working directory. The instructions for creating pipeline.in are given in the next section.**

**After you have pipeline.in, run the pipeline script. The syntax for running the script is as follows. Square brackets indicate optional arguments.**

```
$ pintav6 [--help] [--test] [--no-gptool] [--no-rficlean] [--nodel]
[--retain-aux] [--log-to-file] [--gptdir <...>] [--pardir <...>]
[--rficconf <...>] [--xnbin <...>] <input_dir> <working_dir>
```

**Run pinta as an executable. Don't explicitly invoke the python interpreter.**

The various options and command line parameters are described below.

1. **--help** : Prints out the help message.
2. **--test** : Commands are only displayed and not executed. All the checks are performed.
3. **--no-gptool** : Does not run gptool.

4. **--no-rficlean** : Does not run RFIClean.
5. **--nodel** : Intermediate files (such as .gpt, .fil etc) are not deleted.
6. **--retain-aux** : Auxiliary output files are not deleted and are moved to `working_directory/aux/`.
7. **--log-to-file** : The logs are written to a file called `pinta.log` in the current directory.
8. **--gptdir <...>** : Specify custom directory for `gptool.in` files.
9. **--pardir <...>** : Specify custom directory for `par` files.
10. **--rficconf <...>** : Specify custom RFIClean configuration file.
11. **--xnbin <x>** : Fold the time series to `x*nbin` number of bins in addition to the `nbin` given in the `pipeline.in` file.
12. Positional arguments:
  - a. **<input\_dir>** : Directory containing raw data and timestamp files.
  - b. **<working\_dir>** : Directory containing `pipeline.in`. Output files are written into this directory. This can be the same as `<input_dir>`. If you are already in the working directory you can use `."` to denote the working directory.

## 5.4. Regarding permissions

The user must belong to the “pulsarg” group. (This is the first command.)

The “pulsarg” group must have read permissions for all files within the `pulsar ephemeris` directory, `gptool.in` directory and input directory. It is advisable that the files within these folders be read-only.

The “pulsarg” group must have read and write permissions for all files within the working directory.

The “pulsarg” group must have read and execute permissions for the pipeline script.

## 6. Instructions for creating `pipeline.in`

Each row in this file corresponds to one raw data (“`.dat`”) file. Each raw data file has a corresponding unique timestamp (“`.timestamp` or `.hdr`”) file.

Each row corresponds to one observation and one frequency. Since we use three bands, one pulsar will have three such rows for each observation epoch.

Each row in `pipeline.in` has 12 columns. These are the following.

Column	Parameter	Description	Data Type	Unit
1	JName	The name of the pulsar in J2000 epoch.	String	
2	RawDataFile	Raw data file name. Only the file name is required and not the full path.	String	
3	TimestampFile	Timestamp file name. Only the file name is required and not the full path.	String	
4	Frequency ( $F_{LO}$ )	Local oscillator frequency of the observing band.	Float	MHz
5	NBins ( $N_{bin}$ )	Number of phase bins for the folded profile.	Integer	
6	NChans ( $N_{chan}$ )	Number of frequency channels.	Integer	
7	BandWidth ( $\Delta F$ )	Bandwidth of the observing band.	Float	MHz
8	TSample ( $T_{smp}$ )	The sampling time used for observation.	Float	s
9	SideBand	The sideband. This should be either LSB (lower sideband) or USB (upper sideband).	String	
10	NPol ( $N_{pol}$ )	Number of polarizations (1:=(I), 4:=(I,Q,U,V))	Integer	
11	TSubInt ( $T_{subint}$ )	The duration of individual sub-integrations within which the data will be folded over the pulsar period.	Float	s
12	Cohded	Whether the data has been coherently dedispersed (De & Gupta, 2016). 1 represents Yes and 0 represents No.	Boolean	

**Table 2** Description of various volumns in the `pipeline.in` file.

1. JName --- The name of the pulsar in J2000 epoch. I suggest not using the BName (in B1950 epoch) for the sake of standardization. For the pipeline to work, a par file `<parfile_dir>/<JNAME>.par` must exist.
2. RawDataFile --- The raw data file. Give only the filename and not the full path.
3. TimestampFile --- The timestamp file. Give only the filename and not the full path.
4. Frequency --- The local oscillator frequency of the observation. Depends on the band of observation and observation settings.
5. NBins --- The number of time bins. This should be a power of 2. This is mostly limited by the sampling time of the recorded data --- shorter the sampling time, higher the number of bins we can use.

Giving NBins as -1 in the `pipeline.in` file will result in the data being folded to “ideal” number of bins. This is the highest power of 2 less than  $(\text{pulsar\_period}/\text{sampling\_time})$ .

6. NChans --- The number of frequency channels. Depends on the observation settings. (The values are given in the spreadsheet)

7. BandWidth --- Bandwidth of the observing band. Depends on the observation settings.
8. TSample --- The sampling time. This is written in the setup file. (The values are given in the spreadsheet)
9. SideBand --- The sideband. This should either be LSB or USB. Depends on the observation settings.
10. NPol --- The number of polarization products. Its value should be 1 for our purpose. We record only “total intensity”, i.e., only one polarization parameter, that is why it should be 1.
11. TSubInt --- The duration of individual sub-integrations within which the data will be folded over the pulsar period by dspr. (Use 10 seconds)
12. Cohded -- Whether the data has been coherently dedispersed. This is 0 for PA and 1 for CDPA.

Lines starting with the '#' character are treated as comments and ignored.

The observation settings can be gleaned from the setup file we send to the operator and the observation log file sent by the operator.

**The changes from the previous versions are the following:**

1. LO Frequency is now provided in pipeline.in. The calculation of the first channel frequency is now encoded in pinta.
2. BandWidth instead of ChannelWidth.
3. New parameter Cohded.

The idea is to enable easier creation of pipeline.in from the observation settings without any calculations.

## 7. After the analysis

The summary pdf files can be viewed for basic sanity checks to ensure that the pulsar was detected with an estimate of optimised period and DM. The reduced PSRFITS files can be examined using PSRCHIVE tools. The reduced products can be copied to a place of choice.