The GMRT Pulsar Tool (gptool v4.3.5) Manual

Aditya Chowdhury & Yashwant Gupta

June 25, 2018

Contents

1 Introduction to gptool

The GMRT Pulsar Tool (gptool) is a real time package for pulsar data that has a comprehensive Radio Frequency Interference (RFI) mitigation programme built in with the ability to incoherently dedisperse and fold pulsar data. The tool can also work in offline mode with standard gmrt data files. It processes data in chunks referred to as windows/blocks in this document. For each block it can (if the user selects) filter RFI in either time domain, frequency domain or both. The tool can currently work in real time with 2k channels, down to a sampling interval of 40 microseconds.

2 Installation

gptool depends on the following packages -

- pgplot (with libx11)
- openmp

and requires the following *optional* packages to perform polyco based folding:

- psrcat
- tempo/2

After the dependencies have been installed, the rest can be performed via the following steps:

1. Extract the tarball and cd to the directory.
2. Skip to step 5 if PGPLOT and OPENMP have been installed via default package managers (e.g apt-get /yum)
3. Edit lines 3 and 4 of the makefile to point to the directory that contains pgplot.h and libcpgplot.so respectively. Make sure to retain the -I and -L prefixes.
4. Edit lines 5 and 6 to point to the directory that contains omp.h and libomp.so respectively. This step would not be required on most standard linux flavours.
5. Run 'make' in the directory.
6. Copy the compiled executable 'gptool' to your favourite binary destination or set PATH to the current directory.

3 Running gptool

The directory in which gptool is run must contain a gptool.in file. This file contains all the parameters that the program uses for processing data. The various parameters are explained in details in section 4. In case gptool is run without a gptool.in file, it'll write out a sample in the same directory. This file can then be modified by the user. gptool can either read from a data file or attach itself to either GWB or GSB shared memory and then process data in real time.

- To run it in file mode the command is:

```
$ gptool -f [filename]
```

  In order to do polyco based folding of the data, there must be a [filename].hdr file with the necessary time stamps. There are also quite a few optional parameters.

```
$ gptool -f [filename] -s [start_time_in_sec] -o [
    output_2d_filtered_file] -m [mean_value_of_2d_op]
    -tempo2 -nodedisp  -zsub -gfilt
```

-s	option allows the user to skip an initial portion of the file and start processing from the specified time.
-o	option specifies the location where the filtered 2D file (if any) is to be written to. If this is not specified the output filtered file (if any) will be written out to the same directory as the input file.
-m	mean value of output filtered file. This is the value by which the gptool normalized data is scaled before writing out the filtered file. This is done to ensure minimal quantization loss when writing the output file in 2byte integer format.
-tempo2	will use tempo2 to generate tempo1 style polycos. If this flag is not set, gptool will by default use tempo1.
-nodedisp	will not perform dedispersion and folding. This is useful where one wants to generate the flag files and/or filtered 2D raw data replaced by median values. Note that this will by default turn off the display.
-zsub	will do an optimized zero DM subtraction. Do note that this is an experimental feature. Do use with caution.
-gfilt	turns off all filtering options and over-ride any relevent input from the gptool.in file.

- To attach to a shared memory and run real time, there must be a *-r* flag:

```
$ gptool -r -t [thread_multiplicity]
```

  Here t selects the number of cores to run the tool on. Valid numbers are 1 (default), 2 and 3 which will use 4 (default), 7 and 10 cores respectively. We recommend using -t 2 for most sampling intervals.

4 Understanding the gptool.in file

As mentioned in the previous section, gptool needs a .in file in the directory where it is executed. This .in file contains various parameters that'll be explained in this section. A sample of the file is shown in figure 1. Various recommended parameters can be found in section 7.

```
#*#*#gptool input file v2.0#*#*#
-------------------------------------------------
#****Mode of observation****#
PA                : Beam mode
0                 : Polarization mode (0-> intesity, 1-> stokes data)
2                 : Sample size of data (in bytes, usually 2)
-------------------------------------------------
#****Observation Paramaters****#
1030              : Frequency band (lowest value in Mhz)
200               : Bandwidth(in Mhz)
-1                : Sideband flag (-1-> decreasing +1-> increasing)
2048              : Number of channels
1.31072           : Sampling Interval (in ms)
-------------------------------------------------
#****Pulsar Parameters****#
J1807-0847        : Pulsar name
-1                : Pulsar period (in milliseconds)
-1                : DM (in pc/cc)
-------------------------------------------------
#****Dedispersion & Folding parameters****#
-1                : Number of bins in folded profile (-1 for native resolution)
0                 : Phase offset for folding
12                : Number of coefficients for each polyco span (nCoeff)
60                : Validity of each span (nSpan in mins)
12                : Maximum hour angle
-------------------------------------------------
#****Display Parameters****#
0                 : Polarization channel to display (0-3 or -1 for all four)
1                 : Display window size (seconds, 0-> pulsar period)
0                 : Update mode           (0-> automatic, 1-> manual)
0                 : Time delay between window updates (0-> no delay, 1-> emulate
     real time)
-------------------------------------------------
#****Spectral line RFI mitigation options****#
50                : Number of channels to flag at band beginning
50                : Number of channels to flag at band end
1                 : Frequency flagging options (0-> no flagging, 1-> real time
     calculation)
1                 : Bandshape to use for frequency flagging (1-> normalized
     bandshape, 2-> mean-to-rms bandshape, 3-> Both)
2.5               : Threshold for frequency flagging (in units of RMS deviation)
-------------------------------------------------
#****Time domain impulsive RFI mitigation options****#
1                 : Time flagging options         (0-> no flagging, 1-> real time
     calculation)
1                 : Data normalization before filtering (0-> no, 1-> yes)
1                 : Time flagging algorithm       (1-> histogram based, 2-> MAD
     based)
3                 : Threshold for time flagging (in units of RMS deviation)
-------------------------------------------------
#****Other options****#
20                : Smoothing window size for bandshape normalization (in number of
      channels)
1                 : Normalization procedure (1-> cumulative smooth bandshape, 2->
     externally supplied bandshape.dat)
0                 : Replace by median values (0-> Ignore flagged samples, 1->
     Replace flagged samples by window median, 2-> Replace by smooth bandshape)
-------------------------------------------------
#****I/O options****#
0                 : Write channel flag file (0-> no,1-> yes)
0                 : Write time flag file (0-> no, 1-> yes)
0                 : write out filtered 2D raw data (0-> no, 1-> yes)
0                 : Write out fullDM.raw  (0-> no, 1-> yes)
-------------------------------------------------
#****manual flagging options****#
0                 : Number of bad channel blocks
List              : #in next line, example: [200,400],[1200,1400]
```

Figure 1: The gptool.in file

4.1 Parameters relating to the mode of observation

Line 4; **Beam mode** :

Valid inputs: IA or PA

Description: Information about whether the data to be processed is from Incoherent (IA) or Phased Array (PA). Currently it has no impact on what the program does.

Line 5; **Polarization mode** :

Valid inputs: 0 or 1

Description:

- 0: Total intensity data
- 1: Stokes data. For stokes data it assumes the data to be in default gmrt format where the polarization data of each channel is recorded one after the other and there are four such polarization channels. In this mode, if filtering is turned on, gptool will filter each pol. chan. independently. It'll also write out relevant quantities for each pol. chan.

Line 6; **Sample size of data (in bytes)** :

Valid inputs: 2 or 4

Description:

- 2 byte short integer. GMRT data is usually sampled as 2 byte integers.
- 4 byte floating point number. Such data samples maybe encountered in single channel dedispersed data from other pipelines.

4.2 Observation Parameters

Line 9; **Frequency of observation (lowest value in MHz)** :

Valid inputs: Any positve definite real number

Description: Please note that this is independent of sideband flag. It is NOT channel#1 frequency. For example, if the frequency is ordered from 500-300MHz, even then this value should be 300MHz.

Line 10; **Bandwidth(in Mhz)** :
Valid inputs: Any positve definite real number
Description: The bandwidth of observation in MHz.

Line 11; **Sideband flag** :
Valid inputs: -1 or 1
Description:

- -1: Channel #1 frequency corresponds to the highest frequency, i.e lowest frequency (value in line 7) + bandwidth (value in line 8)
- 1: Channel #1 frequency corresponds to the lowest frequency (value in line 7)

Line 12; **Number of channels** :
Valid inputs: Any positive definite integer (usually multiple of 2)
Description: The number of spectral channels in the data.

Line 13; **Sampling Interval (in ms)** :
Valid inputs: Any positve definite real number.
Description: The sampling time for data.

4.3 Pulsar Parameters

Line 16; **Pulsar Name** :
Valid inputs: A string.
Description: The J or B name of the pulsar. This is used to determine the period and dispersion measure (DM) of the pulsar when it is not specified. This field is ignored if both the period and DM is entered manually by the user in the next two lines.

Line 17; **Pulsar period (in milliseconds)** :
Valid inputs: -1 or any positive definite real number.
Description: The period of pulsar in ms. If the value of this field is -1 then gptool fetches the pulsar period from the ATNF pulsar catalogue (psrcat). And then it fires tempo using the parameters specified in lines 20 to 22 to generate polyco coefficients and performs folding based on that. In case of user entered period, gptool will perform a fixed period folding.

Line 18; **DM (in pc/cc)** :

Valid inputs: -1 or any positive definite real number.

Description: The dispersion measure of pulsar in pc/cc. If the value of this field is -1 then gptool fetches the DM value from the ATNF pulsar catalogue (psrcat).

4.4 Folding parameters

Line 21; **Number of bins in folded profile :**

Valid inputs: -1 or any positive real integer

Description: The number of bins in the folded profile. If -1 it'll use the native resolution (period of pulsar divided by sampling interval).

Line 22; **Phase offset for folding :**

Valid inputs: A real number between 0 and 1

Description: This puts a phase offset in the profile of the pulsar. For example if the on pulse peak was roughly at 0.25 of the period length, with a 0.5 phase offset it'll now appear at 0.75 of the period length.

Line 23; **Number of coefficients for each polyco span (nCoeff) :**

Valid inputs: A positive definite integer.

Description: This field is ignored if the user manually gives a pulsar period in line 15. Otherwise this is passed on to tempo as a parameter that determines the degree of polynomial to which the pulsar phase is expanded as a function of time.

Line 24; **Validity of each span (nSpan in mins) :**

Valid inputs: A positive definite real number.

Description: This field is ignored if the user manually gives a pulsar period in line 15. Otherwise this is passed on to tempo as a parameter that determines the validity of each polynomial expansion of the pulsar phase as a function of time.

Line 25; **Maximum hour angle :**

Valid inputs: A positive definite real number.

Description: This field is ignored if the user manually gives a pulsar period in line 15. Otherwise this is passed on to tempo as a parameter that determines the maximum hour angle of observation.

4.5 Display parameters

Line 28; **Polarization channel to display :**

Valid inputs: -1, 1, 2, 3, 4.

Description: This field is used only when processing stokes data (Line 3).

- -1: gptool displays the time series, dedispersed time series, bandshape & folded profile for all four polarization channels. It is unable to display the flag plots due to a lack of display space.
- 1, 2, 3, 4: The number specifies the polarization channel to display. Please note even when only one polarization channel is being displayed, gptool processes all four polarization channels in the background and writes relevant outputs for each of the channels.

Line 29; **Display window length :**

Valid inputs: 0 or a positive definite real number

Description: In terms of what the user sees, gptool reads and processes data one block at a time. This field specifies (in seconds) the length of each block. If the input is 0, then it sets the length equal to the period of the pulsar.

Line 30; **Update mode :**

Valid inputs: 0 or 1

Description: When this field is 1, the user has to press return for each new plot update. if 0, it automatically moves to process and plot the next window.

Line 31; **Time delay between window updates :**

Valid inputs: 0 or 1

Description: This field is ignored when user selects manual update (1 in line 27).

- 0: gptool will update the plot the moment it has finished processing the next block. At times, this may be too fast to monitor what's happening and in such a case the user can give 1 in this field.
- 1: The program spends atleast the amount of time specified in display window length (line 26). For example, if the window length is 0.5 sec, then gptool will spend atleast 0.5 seconds between successive plot updates. Do note that this field is ignored in real time processing.

4.6 Spectral line RFI mitigation parameters

Line 34; **Number of channels to flag at band beginning :**

Valid inputs: Any positive integer less than the number of channels

Description: gptool by default flags these many channels at band beginning. For example, if the number of channels are 2048 and this field is 50 then gptool will flag the first 50 channels and start processing from Chan # 50 onwards.

Line 35; **Number of channels to flag at band end :**

Valid inputs: Any positive integer less than the number of channels

Description: gptool by default flags these many channels at band end. For example, if the number of channels are 2048 and this field is 50 then gptool will flag the last 50 channels and stop processing at Chan #1998.

Line 36; **Frequency flagging options :**
Valid inputs: 0 or 1
Description:

- 0: gptool will not perform any spectral channel RFI mitigation. All fields from lines 34 to 36 will be ignored.
- 1: gptool will perform real time calculations to detect and filter spectral line RFI based on the parameters in lines 35 to 38. Note that this field does not affect the inputs in lines 32 & 33.

Line 37; **Bandshape to use for frequency flagging :**
Valid inputs: 1 or 2
Description: This field determines the type of bandshape from which outliers are detected. The possible options are:

- 1: Normalized bandshape. The bandshape is normalized using a cumulative smoothed bandshape (using a moving median technique with a window length specified in line 46) or an externally supplied bandshape depending on the input in line 47
- 2: Mean to rms bandshape. The mean and rms for each spectral channel across time is calculated and divided to get the mean by rms bandshape.
- 3: Flag channels using both the bandshapes - a channel is not used if it is flagged by EITHER of the two algorithms.

Either of these are done for every block of data processed.

Line 38; **Threshold for frequency flagging :**
Valid inputs: Any positive definite real number.
Description: This field determines the cutoff rms level (w.r.t to the predicted mean and in units of predicted std. dev. of the underlying Gaussian) above which gptool will treat points as outliers and flag them.

4.7 Time domain impulsive RFI parameters

Line 41; **Time flagging options :**
Valid inputs: 0 or 1
Description:

- 0: gptool will not perform any time domain RFI mitigation. Lines 40 to 44 will be ignored.
- 1: gptool will perform real time calculations to detect and filter time domain RFI based on the parameters in lines 40-42.

Line 42; **Data normalization before filtering :**
Valid inputs: 0 or 1
Description: If 1, the full time-frequency data for each window is normalized using a cumulative smoothed bandshape (using a moving median technique with a window length specified in line 46) or an externally supplied bandshape depending on the input in line 47

Line 43; **Time flagging algorithm :**

Valid inputs: 0 or 1

Description: This field determines the algorithm that is used to calculate the mean and rms of the (zero DM) time series. The options are:

- 1: A histogram based algorithm that generates a histogram from the un-dedispersed (zero DM) time series and then uses the modal point as a measure of the mean and the FWHM as a measure of the standard deviation of the underlying distribution.
- 2: A median absolute deviation (MAD) based algorithm that uses the median as a measure of the mean and the median absolute deviation as a measure of the standard deviation of the underlying distribution.

Line 44; **Threshold for time flagging :**

Valid inputs: Any positive definite real number.

Description: This field determines the cutoff rms level (w.r.t to the predicted mean and in units of predicted std. dev. of the underlying Gaussian) above which gptool will treat points as outliers and flag them.

4.8 Other parameters

Line 47; **Smoothing window size for bandshape normalization :**

Valid inputs: Any positive definite integer less than the number of channels

Description: This defines the window size over which a moving median is calculated to smooth the bandshape. This option is ignored if both line 34 is 2 and line 40 is 0 is OR line 47 is 2.

Line 48; **Bandshape to use to normalize data :**

Valid inputs: 1 or 2
Description:

- 1: Data and/or mean bandshape (line 34 & 40) is normalized using a cumulative smoothed bandshape (using a moving median technique with a window length specified in line 46)
- 2: An externally supplied bandshape is used to normalize the data. This bandshape must be in a file named "bandshape.dat". The file should have two columns separated by spaces, the 1st column being channel number and the second column being the corresponding intensity of that channel.

Line 49; **Replace by median values :**
Valid inputs: 0 or 1
Description:

- 0: Flagged samples are ignored in subsequent processing
- 1: Flagged samples are replaced by the median value of the zero DM time series in subsequent processing. Note that for this to work normalization must be turned out (1 in line 42). Along with that time filtering should also be on (1 in field 43).

4.9 I/O related parameters

Line 52; **Write channel flag file :**
Valid inputs: 0 or 1.
Description: If this is 1 then gptool writes out a chanflag.dat file that contains the channel flags calculated for each window. The file is written out in binary format. The file contains, for each window, N bytes in succession with either 0 for unflagged and 1 for flagged channels, where N is the number of channels.

Line 53; **Write time flag file :**

Valid inputs: 0 or 1.

Description: If this field is 1 then gptool writes out a time flag.dat file that contains the time flags calculated for each window. The file is written out in binary format. The file contains, for each window, N bytes in succession with either 0 for unflagged and 1 for flagged time samples, where N is the number of time samples in that window.

Line 54; **Write out filtered 2D raw data:**

Valid inputs: 0 or 1.

Description: If this field is 1 then gptool writes out a file similar to the input 2D file. It will have the same name as the input file with a .gpt added to the filename and the flagged samples would be replaced by median or zero depending on the input of line 49. Note that this file is written to the same directory as the input file unless a different location is specified by the -o option.

Line 55; **Write out fullDM.raw :**

Valid inputs: 0 or 1

Description: For 1, gptool will write out the calculated dedispersed series (both RFI filtered and unfiltered). This file is in binary format with each time sample being a 4 byte floating point number.

4.10 Manual flagging options

Line 58; **Number of bad channel blocks :**

Valid inputs: Any positive integer.

Description: Number of bad channel blocks that'll be by default flagged, independent of frequency domain flagging options selected. The list must be provided in line 60.

Line 60; **List of bad blocks :**

Valid inputs: Comma separated list of start and end channel for each block. e.g - [1200,1400],[400,600]

Description: The number of blocks specified must exactly match with the input in line 58. Note that the start channel is inclusive and the end channel is exclusive.

5 Plot Window

Figure 2 shows the plot layout of gptool. Various plots are:

- *Waterfall plot (central greyscale):* Spectrogram of current block. Intensity for each channel and each time bin is displayed here.
- *Zero DM plot (bottom most):* Green curve is calculated after ignoring bad channels (if frequency domain filtering is on). The white curve is calculated by collapsing ALL channels (no filtering).
- *Zero DM flags (bar plot above the zeroDM plot):* Displays time domain flags (if time filtering is on). Black are the ones detected as RFI. Note that with too many samples in each window this plot may become unreliable due to fewer pixels available for rendering.
- *Bandshape (right central):* The white curve shows the mean bandshape for the window, the overlaid green is either the cumulative smooth bandshape or an externally supplied bandshape.
- *Frequency flags (bar plot right of bandshape):* Shows detected bad channels. Black ones are flagged.
- *Bandshape used to filter (right of frequency flags):* Plot of either the normalized bandshape or the mean by rms bandshape of the current window, depending on which is selected for filtering in the gptool.in file.
- *Full DM plot (top):* Green curve is the dedispersed time series with all the selected filtering options. The white curve is with NO filtering.
- *Full DM bin counts (bar plots top of full DM plot):* The first bar shows in greyscale the number of channels that went into each dedispersed bin. It is completely white if this number is equal to the total number of channels and completely black if it is zero. The plot above this indicates the dedispersed samples to be ignored if the bin count is less than 40 % of the total number of channels. It is black for bad samples and white otherwise.
- *Profiles:* Green curve on the bottom right is with all selected filtering options (folding of the green dedispersed series). White curve on the top right is with no filtering (folding of the white dedispersed series)

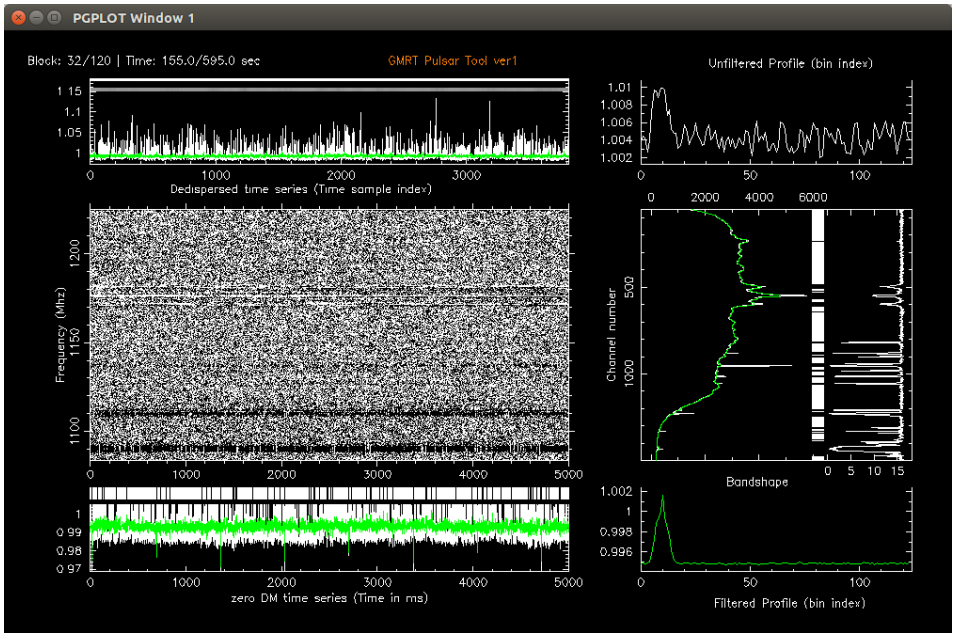

Figure 2: Plot window of gptool.

6 Outputs files

The following are files generated by gptool. Note that for stokes data there will be four copies of each file name [filename][1/2/3/4].[extension], where the number specifies the polarization channel.

6.1 Files written by default

- *profile_filtered.gpt* :
  Format: Two columns of real numbers. (ASCII encoded)
  Description: The files contains the folded profile. First column contains the phase (0-1) and second column contains the intensity at that phase. This is generated from a dedispersed series calculated using all the RFI mitigation options specified by the user

- *profile_unfiltered.gpt*:

Format: Two columns of real numbers. (ASCII encoded)
Description: The file contains the folded profile calculated without any RFI mitigation technique applied. First column contains the phase (0-1) and second column contains the intensity at that phase.

- *bandshape.gpt*:
  Format: Four columns of real numbers. (ASCII encoded)
  Description: The file contains the following global quantities (generated over the entire run of the program) for each channel number (specified in column 1):
  - smooth mean bandshape (column 2)
  - mean bandshape (column 3)
  - rms bandshape (column 4)

- *stats.gpt*:
  Format: Ten columns of real numbers. (ASCII encoded)
  Description: The file contains a set of statistics on the pre and post filtered time series (with no dedispersion) for each block (index in column 1) that gptool processes. They are:
  - mean - predicted (column 2) pre-filter (column 3) post-filter(column 4)
  - rms - predicted (column 5) pre-filter (column 6) post-filter(column 7)
  - mean to rms ratio - predicted (column 8) pre-filter (column 9) post-filter(column 10)

  Here predicted is the statistics on time series predicted by the detection algorithm.

6.2 Files written when specified by the user

- *fullDM_filtered.gpt*:
  Format: 4 byte floating point binary data
  Description: The file contains the dedispersed time series calculated by ignoring all bad channels and bad time samples.

- *fullDM_unfiltered.gpt*:

Format: 4 byte floating point binary data
Description: The file contains the dedispersed time series calculated without any RFI mitigation technique.

- *[input-filename].gpt*:
  Format: 2 byte unsigned integer binary data
  Description: The file contains the full 2D time frequency data with the flagged samples replaced by zeros or the median value depending on the input in line 49 of gptool.in. The encoding of channel information is same as the input file.

- *fullDMCount.gpt*:
  Format: 4 byte integer binary data
  Description: The file contains the number of channels that went into each (filtered) dedispersed time series bin.

- *chanflag.gpt*:
  Format: 1 byte binary data
  Description: The file contains the channel flags calculated for each window. It holds, for each window, N bytes in succession with either 0 for unflagged or 1 for flagged channels, where N is the number of channels.

- *timeflag.gpt*:
  Format: 1 byte binary data
  Description: The file contains N bytes in succession with either 0 for unflagged or 1 for flagged time samples, where N is the total number of time samples.

7 Recommended RFI mitigation settings

Based on our experience with various datafiles and an analysis of the impact of each parameter we recommend these basic RFI mitigation settings.

- Spectral line RFI :
  - Both bandshapes.
  - Threshold of 2.5 sigma
- Time domain RFI :
  - Perform normalization before doing time domain filtering.
  - Histogram based algorithm
  - Threshold of 3.0 sigma (must be set higher for very strong/low DM pulsars)
- Other recommendation :
  - For normalizing bandshape, smoothing window size should be roughly 10% of the number of channels (20 for 2k channels).
  - Each display window (block) should atleast contain 3000 time samples to give the algorithms enough statistics for proper filtering.

8 Change log

New in ver 3

- Merging latest real-time version with latest offline version - streamlining code.
- Replacement with median option
- New parfile fetching policy - ATNF first then internal tempo database.
- Updated gptool.in - version 1.5

New in ver 3.5

- An i/o bug related to writing out the filtered 2D time-frequency data fixed.
- -nodedisp option in command line to skip dedispersion and folding.

New in ver 3.5.5

- Included option to manually flag blocks of channels.
- Updated gptool.in - version 2.0

New in ver 3.7

- Introduction -tempo2 flag to enable (tempo1 format) polycos generation using tempo2.

New in ver 3.8

- Uses signed integers to properly interpret full stokes data. Writes out 2D filtered file with mean of 16384.

New in ver 3.9

- A new spectral line filtering mode that uses both mean by rms AND normalized bandshape to detect RFI.
- Implementation of channel by channel zeroDM subtraction.
- Went back to unsigned integers and a 2D filtered mean of 32k.

New in ver 4.0

- Resolution of data type issue. Separate types for polar and intensity data.
- New summary plot output.
- Quick termination of gptool.

New in ver 4.3.5

- Complete refactoring of the code with some bug fixes.