
dran

Release v1.0

Pfesi van Zyl

Aug 16, 2023

CONTENTS:

1	Introduction	1
2	How does it work ?	3
3	What's next	5
3.1	Installation	5
3.2	Caveats	6
3.3	Quickstart Guide	7
3.4	The command line interface (CLI)	10
3.5	The graphical user interface (GUI)	14
3.6	The landing page	14
3.7	Edit driftscan	15
3.8	Edit Time series	19
3.9	View plots	21
3.10	Radio frequency interference (RFI)	23
3.11	Tutorials	23
3.12	Commands	25

INTRODUCTION

DRAN is a data reduction and analysis software program developed to systematically reduce and analyze [HartRAO's](#) drift scan data collected by [HartRAO's 26m telescope](#). The software is a newly developed program intended to replace the old **LINES** program previously used at HartRAO.

The package was built on Python 3.8 and provides a set of tools to perform common analysis tasks that include but are not limited to,

- Data extraction and preperation,
- Data modeling and fitting, as well as
- Data visualization

Caution: This project is under active development

HOW DOES IT WORK ?

DRAN uses a simple workflow which can be summarised as follows. When a file is read-in, the program creates a dictionary where all the extracted and calculated source parameters are stored.

It then proceeds to

- cleans or filter out any *Radio frequency interference (RFI)* present in the data
- correct or remove any drift that may be present in the data due to telescope effects, and
- tries to fit the peak of the signal (i.e. upper 20 - 50% of the beam) depending on how good the signal is after cleansing.

The result and other statistics are then written into a database where they can then be analysed using various statistical tools. All accompanying plots are stored in the plots folder created in the users current working directory.

Note: Although data is primarily stored in an SQLite database, it can also easily be converted into a csv file.

The main mode of data processing in DRAN is through the *The command line interface (CLI)*. This interface is best suited for automated or semi-automated analysis. For a more hands on approach a *The graphical user interface (GUI)* is also provided and is best suited for handling single file inspection, fitting and timeseries analysis.

WHAT'S NEXT

To get a better grasp of how the program works a set of *Tutorials* have been provided. For instructions on how to install DRAN see the *Installation* page. To jump right in, a short *Quickstart Guide* guide is also provided. Also please read through the *Caveats* to keep updated on the program limitations and assumptions made before beginning any analysis.

3.1 Installation

The DRAN program was developed using the anaconda virtual environment. Running the code on a virtual environment is highly recommended to avoid package clashes, and the following installation shows how to do this using [anaconda](#). Anaconda is a data science toolkit with a wide variety of open-source packages and and libraries under one hub for building powerful projects.

Note: There is an issue with the installation of pyqt5 (the library required to run the GUI) on some machines while loading the requirements file during program installation. Thus it is highly recommended that the user utilizes an anaconda installation or a virtual environment of their choice when installing the program. Please also note that the software has only been tested on anaconda so using a different virtual enviroment server might not work.

3.1.1 Installing and running dran in an anaconda virtual environment

First you will need to [download anaconda](#) and install it on your machine. Once installed you will create the dran virtual environment and activate it using the following commands

```
$ mkdir dran
$ cd dran
$ conda create --name dran --file requests.txt
$ conda activate dran
```

Alternatively, you can use the pypi package manager

```
$ pip install dran
```

Once your virtual environment is activated. You can now head over to *Tutorials* to start using DRAN.

Note: The program has not been tested on pyenv or env yet. This is planned for future releases.

3.2 Caveats

It is highly recommended that you familiarize yourself with this page before starting any data reduction or analysis. This list here is not an exhaustive list and will be updated on a case by case basis.

Operating systems

This code has only been tested in MacOS and Linux. There is no provision at the moment for Windows systems.

Help options

Although there are multiple options you can select to run your data processing, they can not all be run simultaneously. For detailed information on how to use the options please refer to the [Tutorials](#) .

Order of commands when running the program

Warning: COMPLETE THIS!

The program may fail or refuse to process the data if the order of the commands is not placed appropriately. For example, this will run

but this

will produce the following error.

The above error showing up does not mean the program doesn't work. Its basically a warning telling you you need to rearrange the way you wrote your commands. What seems to work best is writing the command for reading in the file first then all the other extra commands after that. Not sure why this bug exists as yet.

Processing data from tiered directory

Warning: TRYING TO FIX THIS! as soon as possible

If for instance you have file located in data/files/more_files/file.fits . And there are also some other fits files in data/files/ . The program will try to process all the files in data/files/ that are .fits files first, then try to process your file in data/files/more_files/ . This tiered nature causes issues where the program fails to detect whether it has processed files in data/files/more_files/ or data/files/ leading to the program repeat fitting files that have already been processed and completely skipping others that have not been processed at all. This is a bug I am aware of but have not begun fixing yet. For best results make sure the file or files you are trying to process follow the folder/***.fits file path format or just get the direct path to the specific file you want.

The dashboard

When opening the dashboard, if the dashboard doesn't open immediately or you get a server not found error you may need to refresh the page so that the dashboard can show up on your screen.

Processed data storage

Currently the program only stores processed data in an [SQLite](#) database. The CALDB database stores calibrated data, this is all data from HartRAO monitored calibrators. Contact the author to get a list of these calibrators if you are unsure. The TARDB database stores all the processed target source information. These are sources that do not fall into the known calibrator list. If you process a source that is not in the HartRAO list of calibrators, this source will be treated as a target source and all the processed information will be stored in the TARDB database. Support for MySQL has not been implemented yet but will be in a future release.

Database error on initial run

When you run the code for the first time, because there will be no database created and the following warning will pop up

Warning: Failed to read FILENAME column from table ‘TABLENAME’, the table either doesn’t exists or is corrupted.

This warning will not stop the processing of data and the code will continue to run as per normal. In consequent runs this warning will no longer appear.

Database analysis

Analysis on sources default to Hydra A as the main calibration source. To select own calibrator it s advised to use the gui timeseries editor.

Note: Special cases with regards to observations exists. These cases are mainly concerned with different projects that are done to test certain properties of the telescope. The analyses of any calibrator contains only that data considered to be normal data. Instances where certain things are beign tested, e.g. gain curve calibration, and these data are recorded in the database, then these are ignored.

Memory allocation

The program focus at the moment is to ensure the correct output is delivered to the user. This causes problems with memory allocation when running large batch analysis processes. For MacOS users you may get a “Your system has run out of application memory error” during processing. To solve this problem, clear the python application and restart the analysis. Unfortunately this problem can only be addressed in future releases.

3.3 Quickstart Guide

After the successful [installation](#) of the package libraries you can now start reducing the data. DRAN provides you with a variety of options to process drift scan data which are discussed below.

3.3.1 Starting the program without commands

As a beginner, it is advised to start running your analysis by basically just entering the program name without any options. This is useful as it starts the program and shows you step by step options you can select to reduce and analyse your data.

To start the program type

dran

This command starts the program and displays a set of options a user can select to begin processing drift scan data.

```
#####
#
# #
# ##### # #
# # # # #
# # # # #
# # # # #
# ##### # # #
```

(continues on next page)

(continued from previous page)

```
#
#
#####

#    PROGRAM STARTED

## SELECT OPTION:
-----

1. Data reduction
2. Open GUI
3. Run command line analysis
4. Open documentation
5. Open dashboard
6. Exit program

Please select an option

Option Selected:
```

3.3.2 Starting the program with command line options

To start the program with command line options.

The program also has an added functionality to let a user type in specific commands that can be run directly from the command line. To see these commands, type the following

```
dran -h
```

This will print out the set of options listed below,:

```
usage: dran.py [-h] [-db DB] [-f F] [-force FORCE] [-c C] [-b B]
              [-delete_db DELETE_DB] [-mfp MFP] [-keep KEEP]
              [-deleteFrom DELETEDFROM] [-conv CONV] [-saveLocs SAVELOCS]
              [-fitTheoretical FITTHEORETICAL]
              [-applyRFiremoval APPLYRFIREMOVAL] [-fitLocs FITLOCS]

Begin processing HartRA0 drift scan data

optional arguments:
-h, --help            show this help message and exit
-db DB                turn debugging on or off. e.g. -db on, by default
                     debug is off
-f F                  process file or folder at given path e.g. -f data/Hydr
                     aA_13NB/2019d133_16h12m15s_Cont_mike_HYDRA_A.fits or
                     -f data/HydraA_13NB
-force FORCE           force fit all drift scans y/n e.g. -force y. Default
                     is set to n
-c C                  initiate the command to run program. e.g. -c gui or -c
                     analysis or -c cli
```

(continues on next page)

(continued from previous page)

```

-b B          initiate browser. e.g. -b docs or -b dash
-delete_db DELETE_DB  delete database on program run. e.g. -delete_db y or
                        -delete_db CALDB.db
-mfp MFP        multi-file processing of data between two dates. e.g.
                        -mfp fileList.txt
-keep KEEP      keep original plots while processing data. e.g. -keep
                        y
-deleteFrom DELETEFROM
                        in coordination with the filename, use deleteFrom to
                        delete a row from a database e.g. deleteFrom CALDB
-conv CONV      convert database tables to csv. e.g. -conv CALDB
-saveLocs SAVELOCS  Save fit locations. e.g. -saveLocs y , default is no
-fitTheoretical FITTHEORETICAL
                        Fit theoretical locations of the baseline, i.e. the
                        FNBW locs. e.g. -fitTheoretical y , default is no.
-applyRFIremoval APPLYRFIREMOVAL
                        Apply or don't apply RFI removal e.g. -applyRFIremoval
                        n, default is yes
-fitLocs FITLOCS   Fit data at specified locations e.g. -fitLocs
                        pathToFile

```

Warning: If you enter a command that doesn't exist the following error pops up

e.g. \$ python dran.py command_that_doesnt_exists

```
usage: dran.py [-h] [-db DB] [-f F] [-force FORCE] [-c C] [-b B] [-delete_db DELETE_DB] [-mfp MFP] [-keep
KEEP] [-deleteFrom DELETEFROM] [-conv CONV] [-saveLocs SAVELOCS] [-fitTheoretical FITTHEORETI-
CAL] [-applyRFIremoval APPLYRFIREMOVAL] [-fitLocs FITLOCS]
```

```
dran.py: error: unrecognized arguments: command_that_doesnt_exists
```

For example:

To perform data reduction on a single file, type in

```
dran
```

This will bring up the menu shown below with options to select how you want to proceed with your data reduction or analysis.

In our case we would select option 1 as we are looking to get an automated reduction.

Option 1

Upon selecting option 1, the following prompt will be displayed

Here you will be shown 3 more options. - Option 1 : reduce 1 file - Option 2 : reduce a folder containing files of 1 source at a particular frequency - Option 3 : reduce all files in the directory where all of your sources are located.

In our case we will select option 1 again as we are only looking to process a single file. From here a gui will pop up asking you to locate the file you want to process. Once the file has been located, the program will automatically process and fit your data and make plots of the fit which will be located in the plots folder. Congratulations, you have just ran your first data reduction.

Option 2

Allows the user to manually reduce the data. With this option the user can tinker with fitting settings and options to get the best fit possible for what they are working on.

Option 3

Opens up the the GUI.

Option 4

Opens up the web resource. The resource contains the code documentation as well as **step by step tutorials** and results from the test data.

Option 0

Exits the program

As shown above some options contain follow up options to further refine your processing selection. Its ultimately up to the user to decide how they want to process their data.

More examples.

To perform autotomated data reduction on a single file given the absolute path

dran -f filepath

i.e. dran -f/Users/pfesesanivanzy/Final-PhD/code/all_hartrao_data/HydraA_13NB/2014d047_20h30m12s_Cont_mike_HYDRA_A.fits

To perform data reduction on a single file with debugging

dran -f filepath -db y

i.e. dran -f/Users/pfesesanivanzy/Final-PhD/code/all_hartrao_data/HydraA_13NB/2014d047_20h30m12s_Cont_mike_HYDRA_A.fits
-db y

To perform data reduction/analysis using the GUI dran -g gui

Once the GUI landing page pops up select the “Edit Driftscan” option.

For a full tutorial please view the web resource on option 4.

3.4 The command line interface (CLI)

In this mode data can be processed in either automated or manual mode. In automated mode, the user has the option to process the data using the predefined data reduction method “dran-auto” using the prompt

```
$ dran-auto path_to_file
```

or

```
$ dran-auto path_to_folder
```

This option automatically selects and locates all the required fitting parameters including the positions and locations of the baseline blocks and the position of the peak. The default fitting methods for the data performs a 1st order fit to the baseline blocks and a 2nd order polynomial fit for the peak. This processing method can also be semi-automated allowing for minor changes which can be found in the help resource which can be accessed using the following prompt

```
% dran-auto -h
```

```
usage: DRAN-AUTO [-h] [-db DB] [-f F] [-delete_db DELETE_DB] [-conv CONV]
               [-quickview QUICKVIEW] [--version]
```

(continues on next page)

(continued from previous page)

Begin processing HartRAO drift scan data

optional arguments:

```
-h, --help          show this help message and exit
-db DB              turn debugging on or off. e.g. -db on, by default debug is off
-f F                process file or folder at given path e.g. -f
                    data/HydraA_13NB/2019d133_16h12m15s_Cont_mike_HYDRA_A.fits or -
↳ f                data/HydraA_13NB
-delete_db DELETE_DB delete database on program run. e.g. -delete_db all or -delete_
↳ db CALDB.db
-conv CONV          convert database tables to csv. e.g. -conv CALDB
-quickview QUICKVIEW get quickview of data e.g. -quickview y
--version           show program's version number and exit
```

On the other hand manual mode gives the user total autonomy on the drift scan reduction process. However, this implementation has not been activated yet.

The following tutorial will show you how to process data in semi-automation.

3.4.1 automated data reduction

Semi-automated data reduction involves typing in commands to run certain types of analysis using the program. This is the preferred mode of data reduction as it caters for both single and batch mode data analysis.

Before starting any analysis it is recommended that you first read the help doc to familiarize yourself with the basic commands required to perform data analysis with DRAN. This is done using

```
$ python dran.py -h
```

The above line of code outputs the following:

```
1 usage: dran.py [-h] [-db DB] [-f F] [-force FORCE] [-c C] [-b B] [-delete_db DELETE_DB]
2 ↳ [-mfp MFP]
3 [-keep KEEP] [-delete_from DELETE_FROM] [-conv CONV]
4
5 Begin processing HartRAO drift scan data
6
7 optional arguments:
8   -h, --help          show this help message and exit
9   -db DB              turn debugging on or off. e.g. -db on, by default
10                      debug is off
11   -f F                process file or folder at given path e.g. -f data/Hydr
12                      aA_13NB/2019d133_16h12m15s_Cont_mike_HYDRA_A.fits or
13                      -f data/HydraA_13NB or -f data/
14   -force FORCE         force fit all drift scans y/n e.g. -force y. Default
15                      is set to n
16   -c C                initiate the command to run program. e.g. -c gui or -c
17                      run_auto_analysis or -c cmdl
18   -b B                initiate browser. e.g. -b docs or -b dash
19   -delete_db DELETE_DB delete database on program run. e.g. -delete_db y
```

(continues on next page)

(continued from previous page)

```

19  -mfp MFP          multi-file processing of data between two dates. e.g.
20                    -mfp fileList.txt
21  -keep KEEP       keep original plots while processing data. e.g. -keep y
22  -delete_from DELETE_FROM
23                    in coordination with the filename, use delete_from to delete a
24  row from a database
25                    e.g. delete_from CALDB
    -conv CONV       convert the database tables to csv. e.g. conv y

```

Depending on the process you want to run, you can select one or more of the available options. If there are more options you would like implemented please email the author.

To perform an automated data reduction process on a single file

```
$ python dran.py -f path-to-file/filename.fits
```

if you want to set debuggin on

```
$ python dran.py -f path-to-file/filename.fits -db on
```

To force a fit on all drift scans, especially those that the program would generally categorize as bad scans and not fit them

```
$ python dran.py -f path-to-file/filename.fits -db on -force y
```

To delete both the calibration and target databases everytime the program starts

```
$ python dran.py -f path-to-file/filename.fits -delete_db y
```

To process multiple chunks of data within separate periods of the year you use the following command

```
$ python dran.py -mfp file_list.txt
```

Note: This assumes that there exists a file called fileList.txt in the current directory that has the full path to the folder containing the source fits files you want to process. This file also has a start and end date in “YYYYdDDD” format stipulating the data range you want to process.

To process all the data located in your directory, this is the directory that contains all the folders containing your fits files

```
$ python dran.py -f path-to-directory
```

To run data reduction using the GUI

```
$ python dran.py -c gui
```

To run data reduction using the GUI with a pre-selected file

```
$ python dran.py -c gui -f path-to-file
```

To initiate the browser to view the analyzed data on a dashboard


```
$ python dran.py -f path-to-file/filename.fits -b dash
```

To view the web documentation guide of the software

```
$ python dran.py -f path-to-file/filename.fits -b docs
```

To run the semi-automated data reduction process through the command line

```
$ python dran.py
```

and follow the prompts. This will bring up the menu shown below with options to select how you want to proceed with your data reduction or analysis.

```

1 #####
2 #
3 #           #####  #####  #####  #   #
4 #           #   #   #   #   #   #   #   #
5 #           #   #   #####  #####  #   #
6 #           #   #   #   #   #   #   ##
7 #           #####  #   #   #   #   #   #
8 #
9 #
10 #####
11 #           PROGRAM STARTED
12
13
14 ## SELECT OPTION:
15 -----
16
17 1. Data reduction
18 2. Open GUI
19 3. Run command line analysis
20 4. Open documentation
21 5. Open dashboard
22 6. Exit program
23
24 Please select an option
25
26 Option Selected:

```

To process data using the gui

```
$ python dran.py -f path-to-file/filename.fits -c gui
```

3.4.2 Manual data reduction

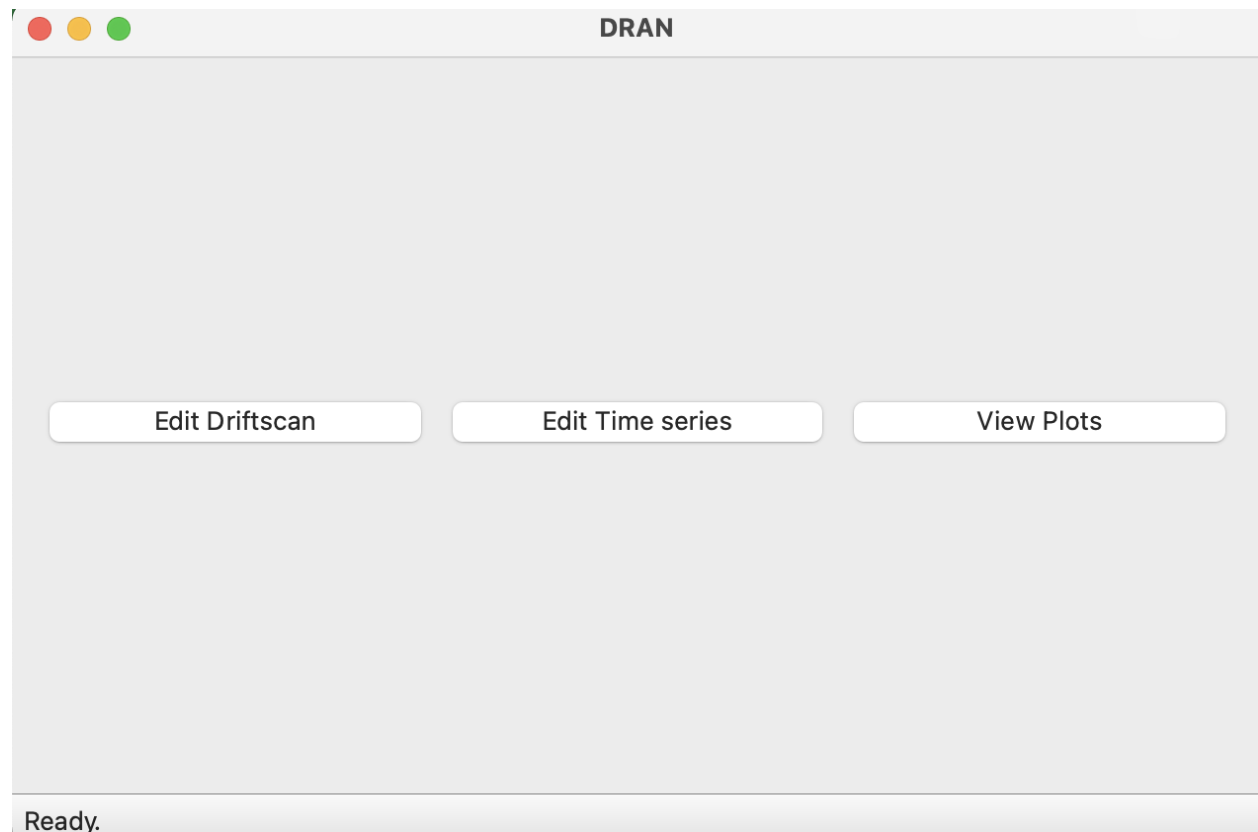
This has been implemented but not complete for testing yet.

3.5 The graphical user interface (GUI)

The graphical user interface lets the user manually examine a drift scan. This is done through a graphical interface that enables a user to select specific locations on the drift scan they would like to fit. The gui also allows the user to view changes made to a drift scan and adjust them accordingly. A walk through of the GUI is given below.

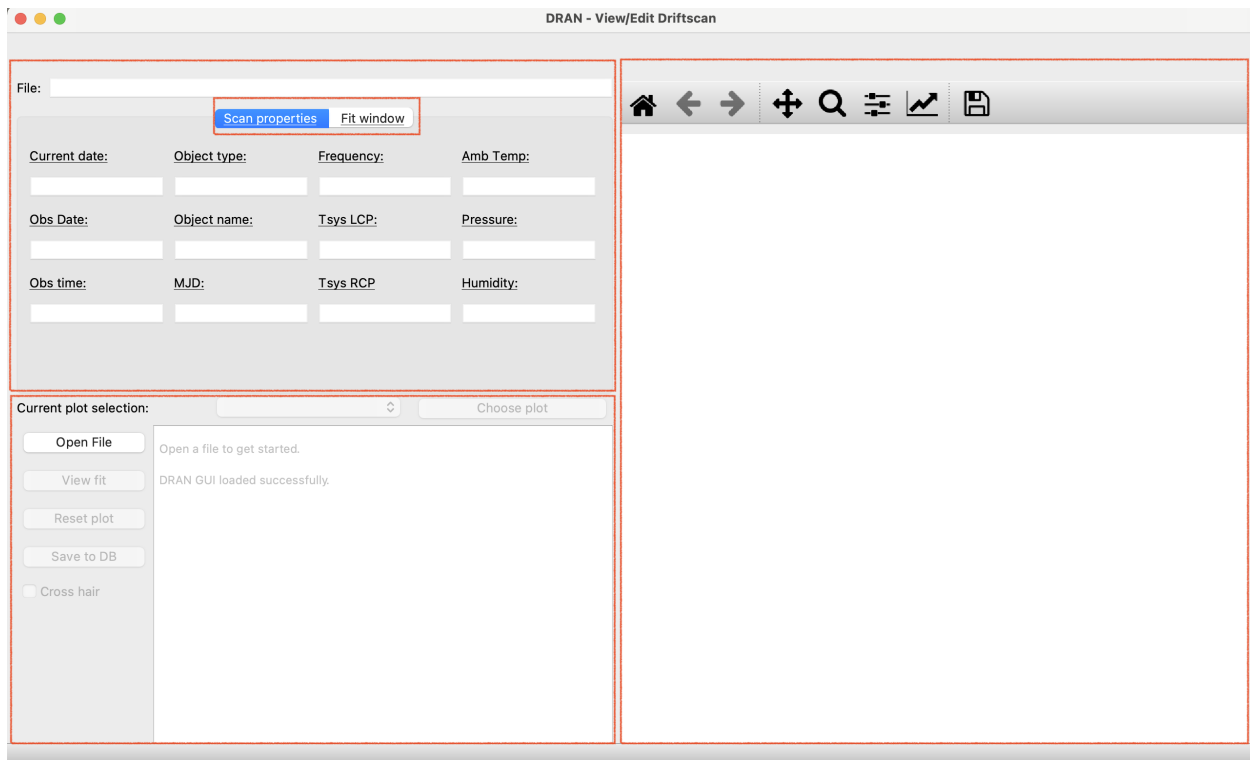
3.6 The landing page

The first page you see when the GUI is initiated is the landing page. This page contains the different options the GUI offers with regards to viewing and processing your data. You can edit your drift scan (e.g. perform fitting), edit your timeseries (e.g. manually remove outliers) and view plots of the processed data. A brief summary of each of these options is given below.

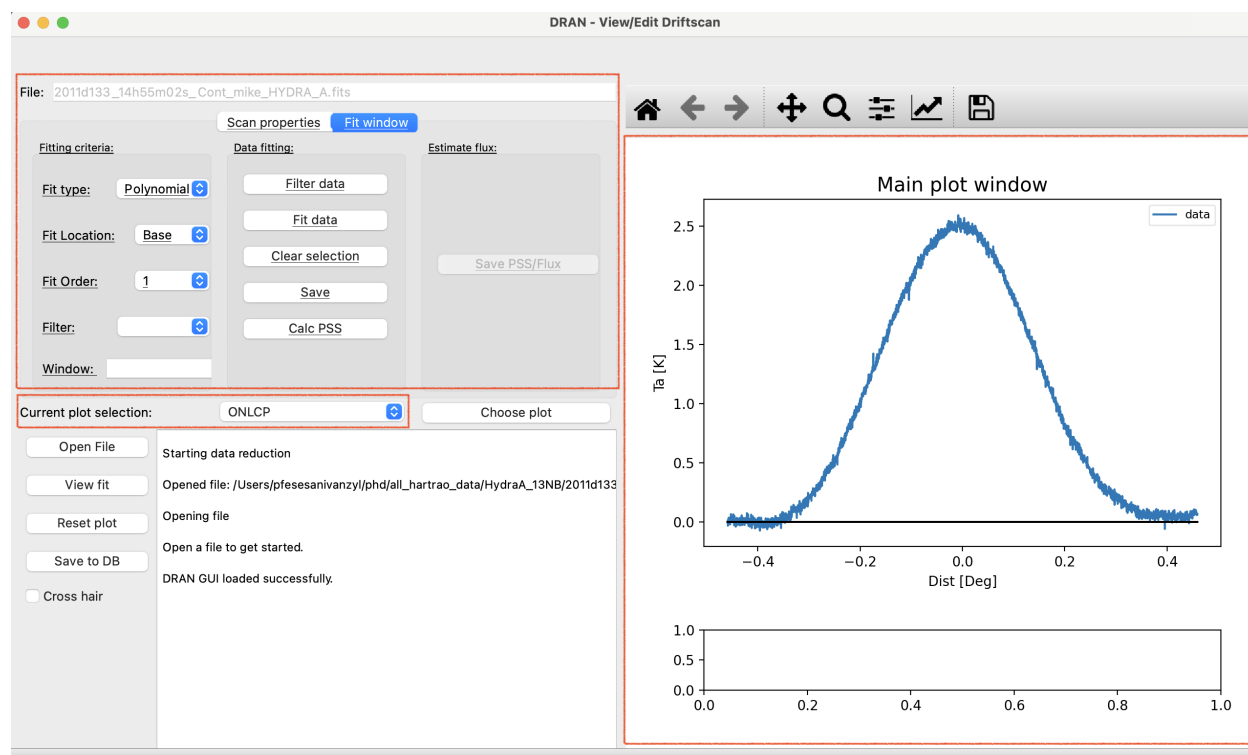


3.7 Edit driftscan

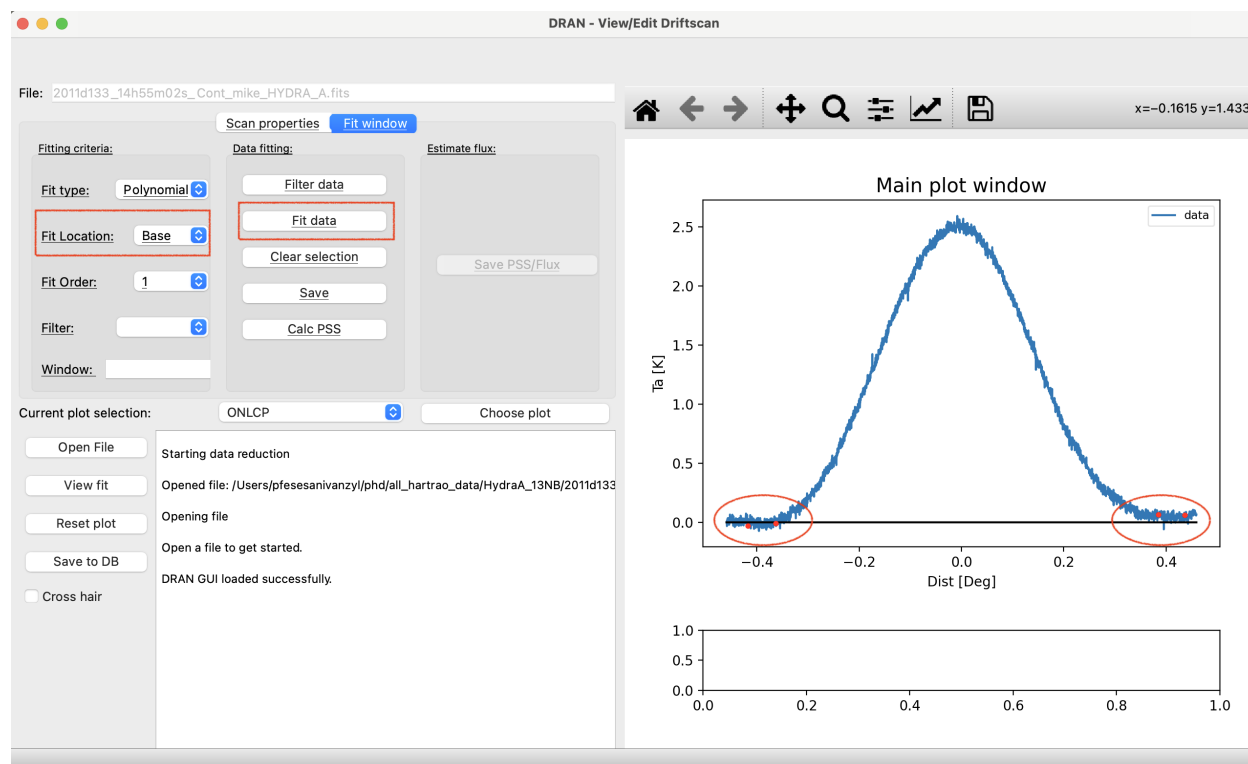
The “Edit Driftscan” button opens up the drift scan editing window. This page shows the basic layout of the GUI. On the left is the “Scan properties” tab, this tab consists of a curated list of some of the drift scan properties found in the drift scan fits file on the observed source. On the second tab, “Fit window”, is information used for the actual fitting. This is currently hidden here but will be shown in the next image. We also have a section to display all the text outputs resulting from running operations on the program, as well as buttons to open a file, view, reset and save drift scan fitting information. On the right is the plot window where the drift scan data is displayed. The next plot shows what the GUI looks like once a drift scan file is opened.



Upon opening a file, the drift scan data is loaded onto the plot window. The top plot displays the current drift scan, and the bottom one is a place holder for the residuals. This operation also toggles the display to the “Fit window” mentioned previously. On this window is the information we need to fit the beam/beams of the drift scan. To toggle between different drift scans we use the “current plot selection” toggle.

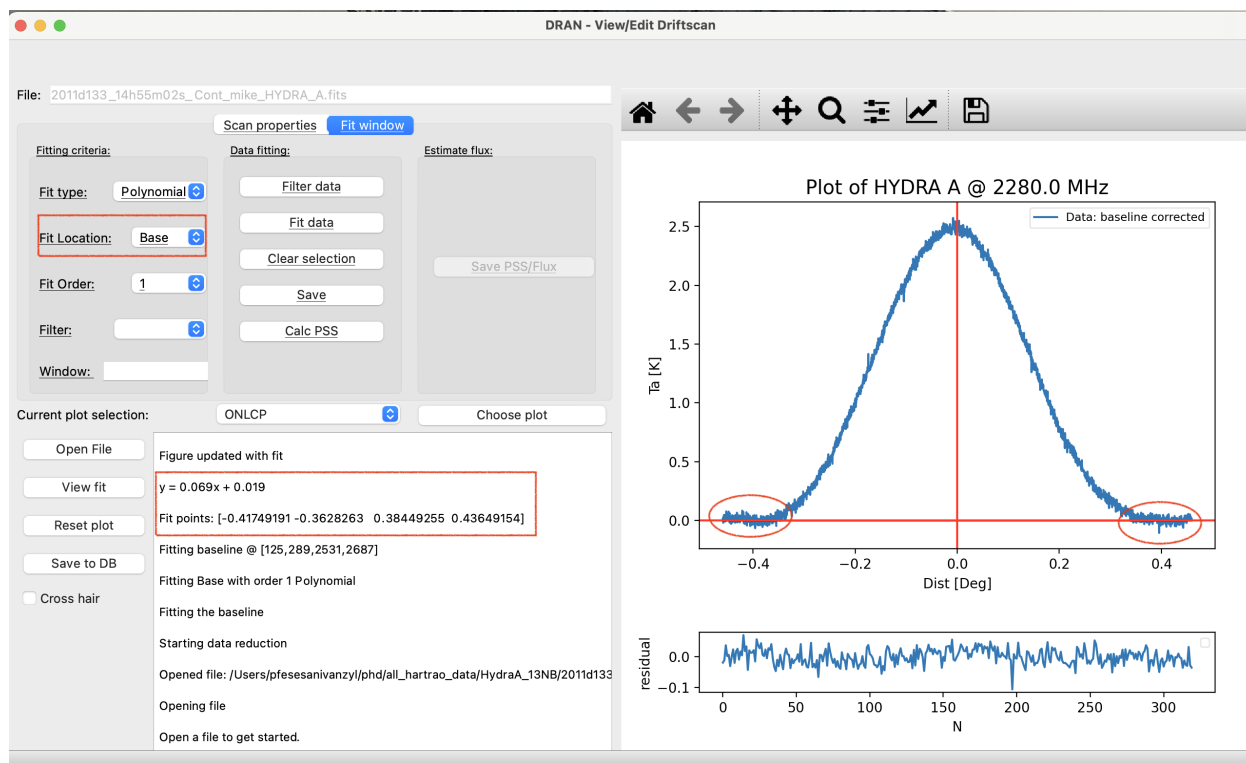


To beginning fitting the drift scan, one can click on the drift scan at the location at which the fit is to be done as shown in the following image.

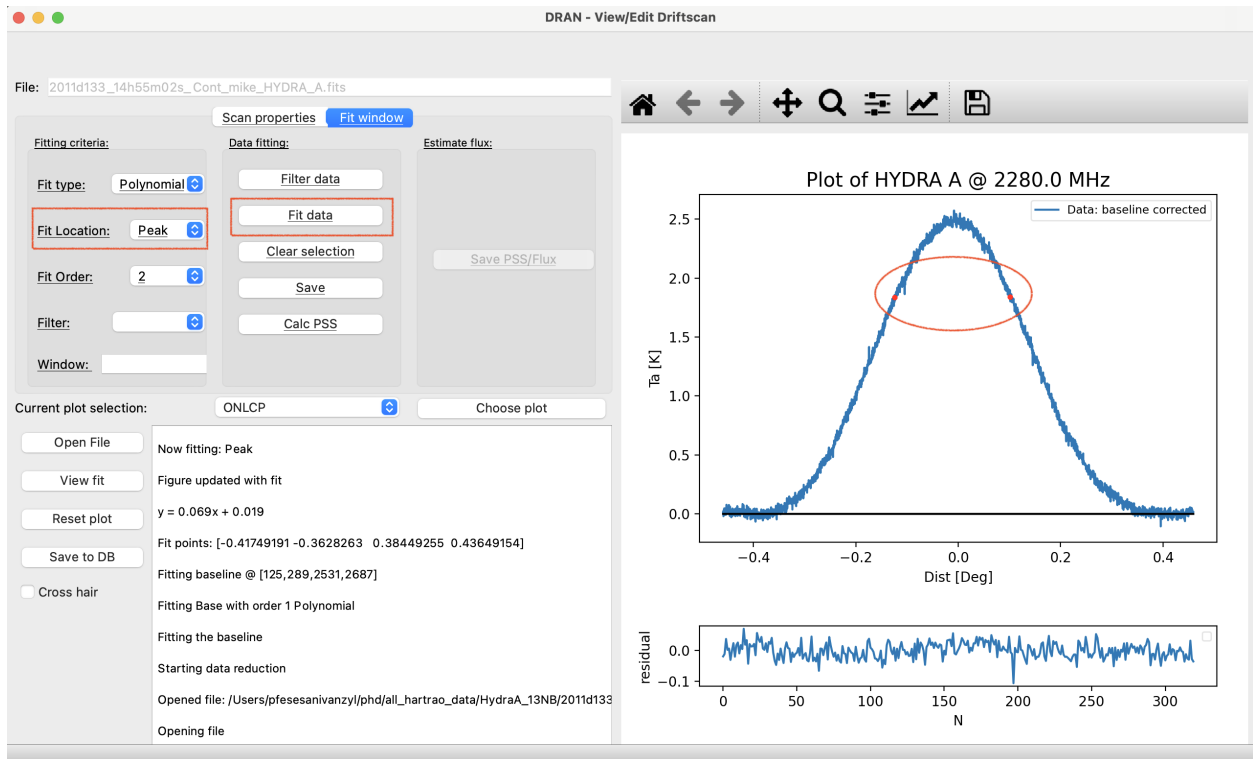


The code is hot wired to process fitting the baseline and peak differently. This is controlled by the “Fit Location” toggle. If you want to fit the baseline, the toggle needs to be on “Base”. Then once you are happy with the locations you clicked on or selected on the plot, you use the “Fit data” button to initiate the actual fit to the baseline blocks selected in the

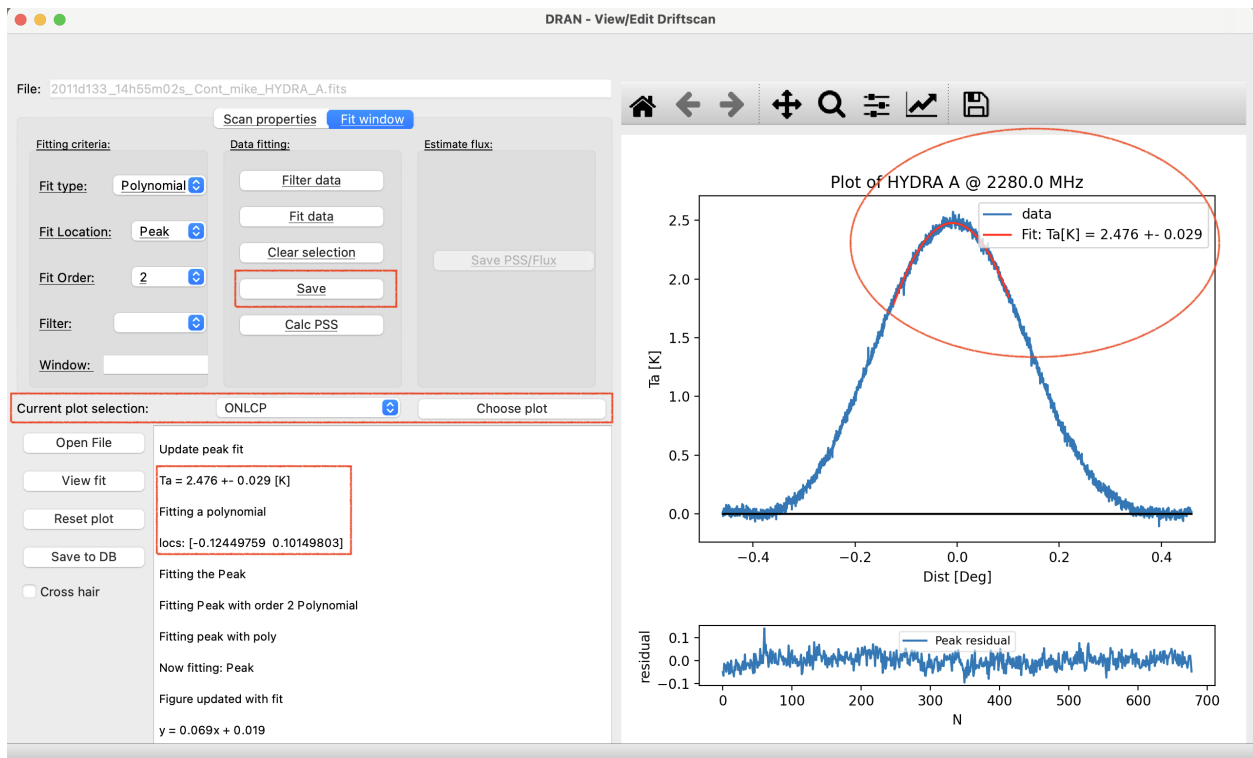
previous step. This creates a new plot with the baseline drift removed as shown below.



The default baseline fit is a 1st order polynomial, however, this can be adjusted accordingly using the “Fit order” toggle. There are also options to filter the data (smoothing/removing RFI) if needed, with a smoothing window provided to cater for that. After the baseline is corrected the equation of a straight line as well as the points used for the fit are displayed as well. When fitting the peak, the peak selection works the same way as the baseline selection with the exception that the “Fit Location” toggle is set to “Peak”. Depending on where you want to fit around the peak, you need to select peak fitting points as shown below.

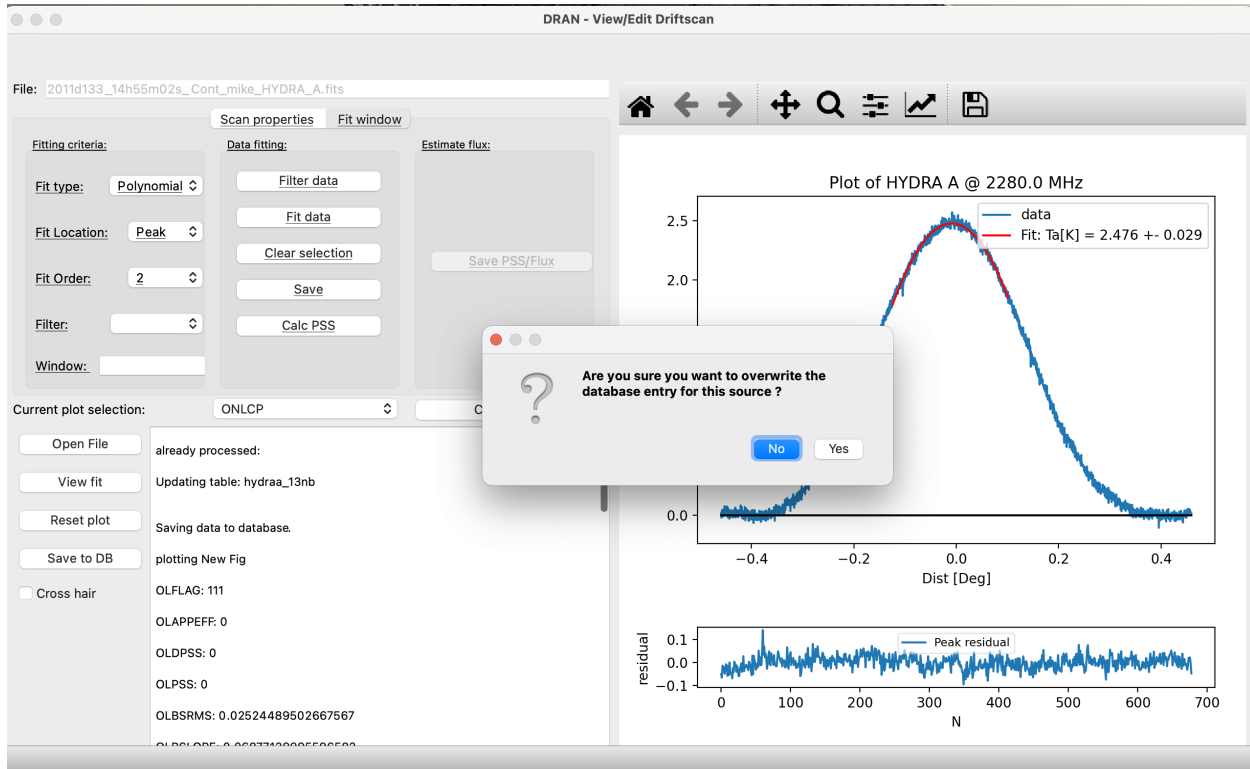


Once you are happy with the selection points, you press “Fit data” again so the program fits the peak of the corrected data. This will create a new fit overlayed on the plot with information on the peak fit and its error. It is this value that we use to calculate the PSS or the source flux depending on the target object under investigation.



When you are happy with the fit, you need to save it using the “Save” button. This saves the changes made to the current

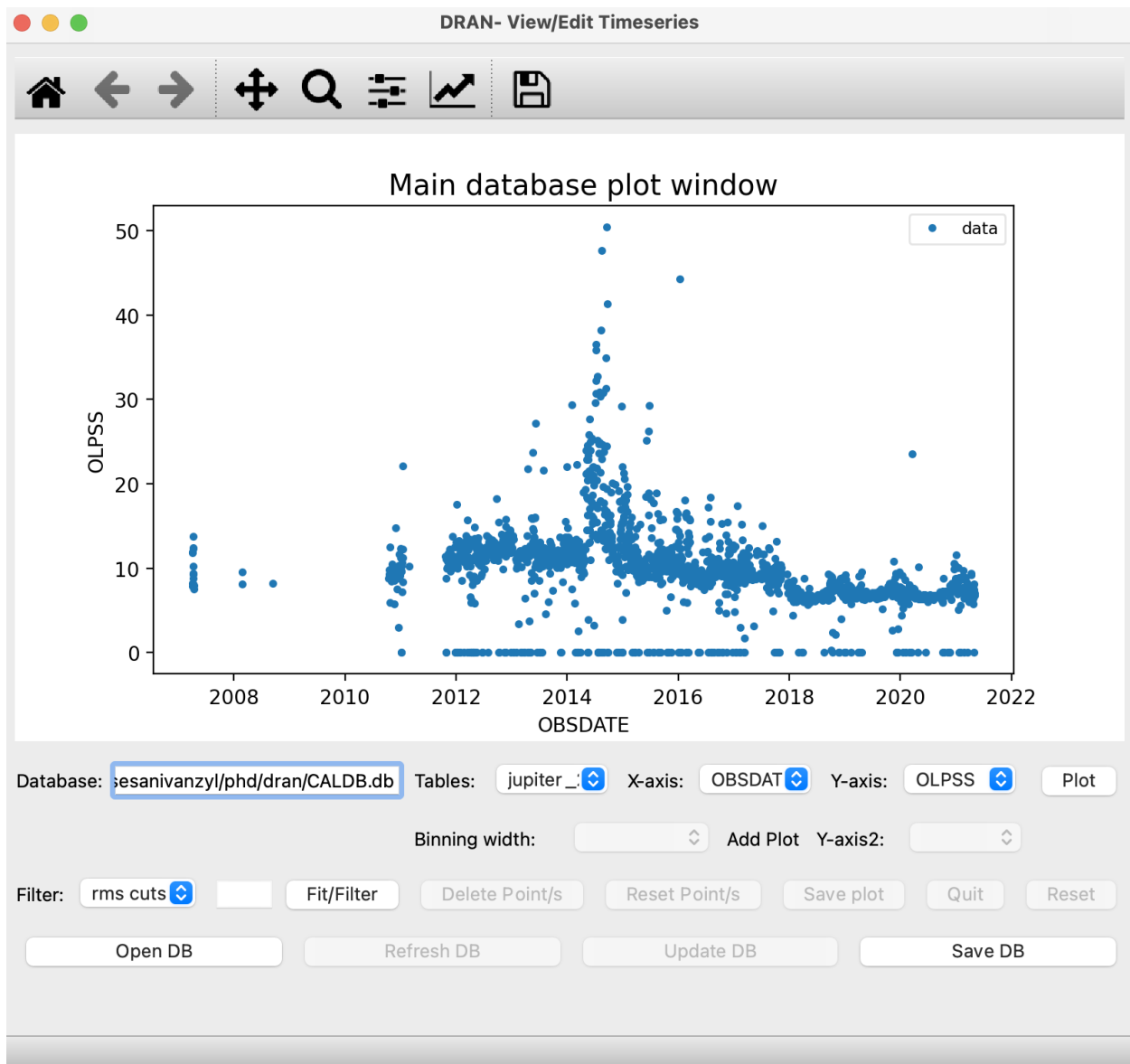
session, in this example, we are saving changes made to the LCP ON SCAN drift scan. The minute you press save, any previous data you had previously processed on this observation for this scan will be replaced by the changes you made. This will also update the previous image you had. If you want to make changes to other sessions as well you can use the “Current plot selection” toggle to change to new drift scan session and load a different set of data. It is imperative to remember that for any changes to be save you need to click on the “Save” button for each session or each different drift scan. To view the values that will be modified on the database, click on “View fit”. At this point the changes are still local to your session, so to make the changes permanent on the database you need to click on “Save to DB”. This updates the database and makes your changes permanent. A popup will appear and ask if you are sure you want to continue modifying the values in the database, If you are, you click yes and end your session. If you want for example to revert to the previously automated fit that you accidentally modified using the GUI, you need to go into the database and delete the observation from there, then run the code again in automated mode to re-process the automated fit.



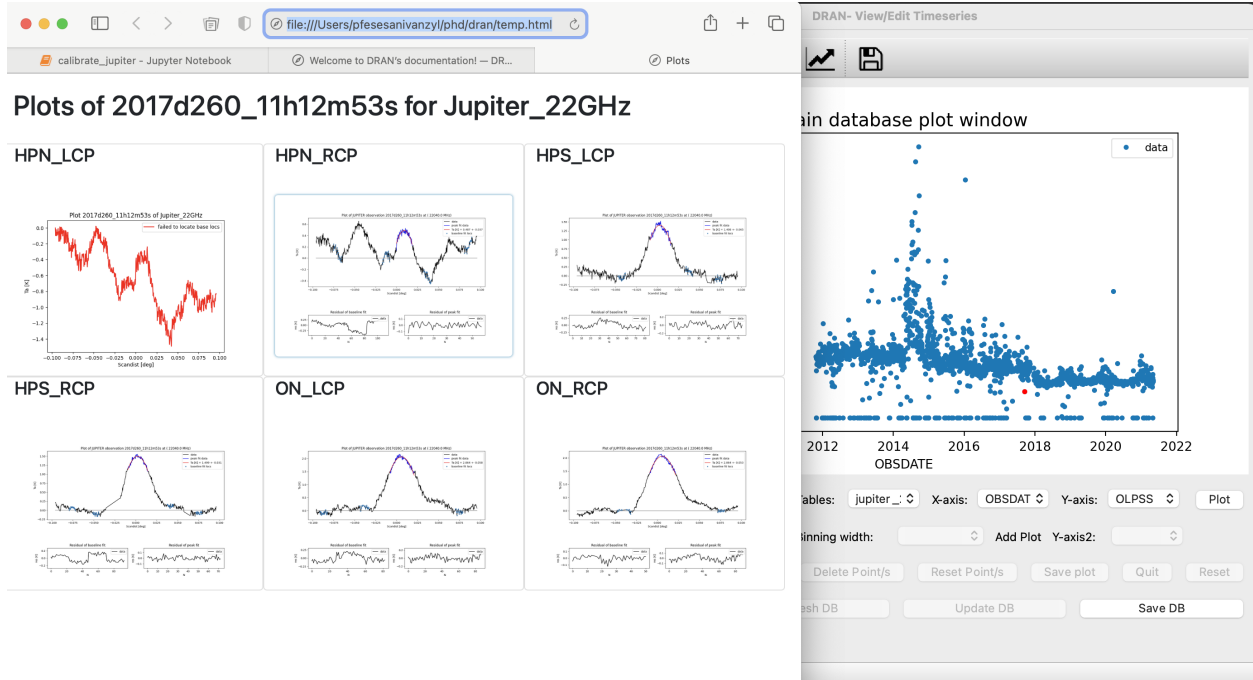
3.8 Edit Time series

Currently this feature is not fully operational. The plan is to have this gui allow you to modify timeseries data manually. Right now, all you can do is view the timeseries data and view the drift scan plots responsible for the points on the time series data.

Opening a database in “Edit Time series” lets you plot any timeseries on the data provided. For this example I’m plotting observed date vs PSS.



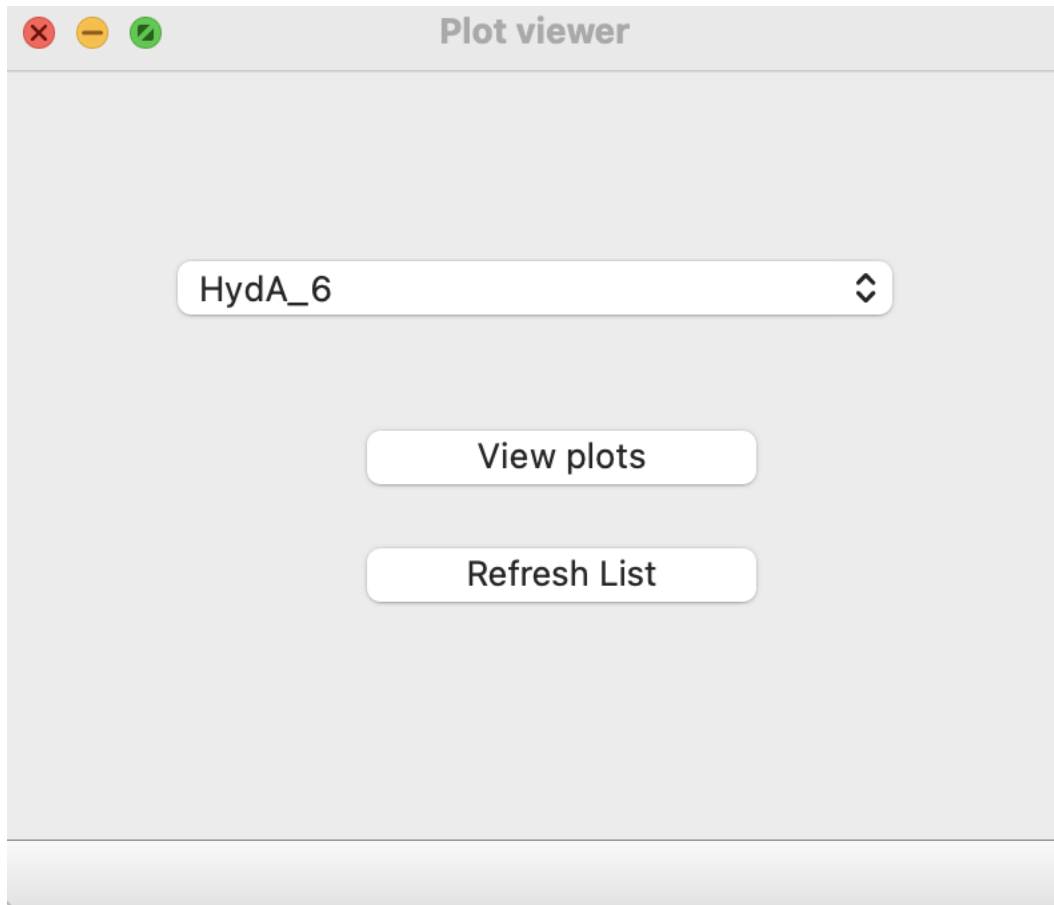
To view the drift scan plots that provided a specific point on the timeseries you click on the point in the timeseries and a popup html will show up with all the processed scans for that specific point.



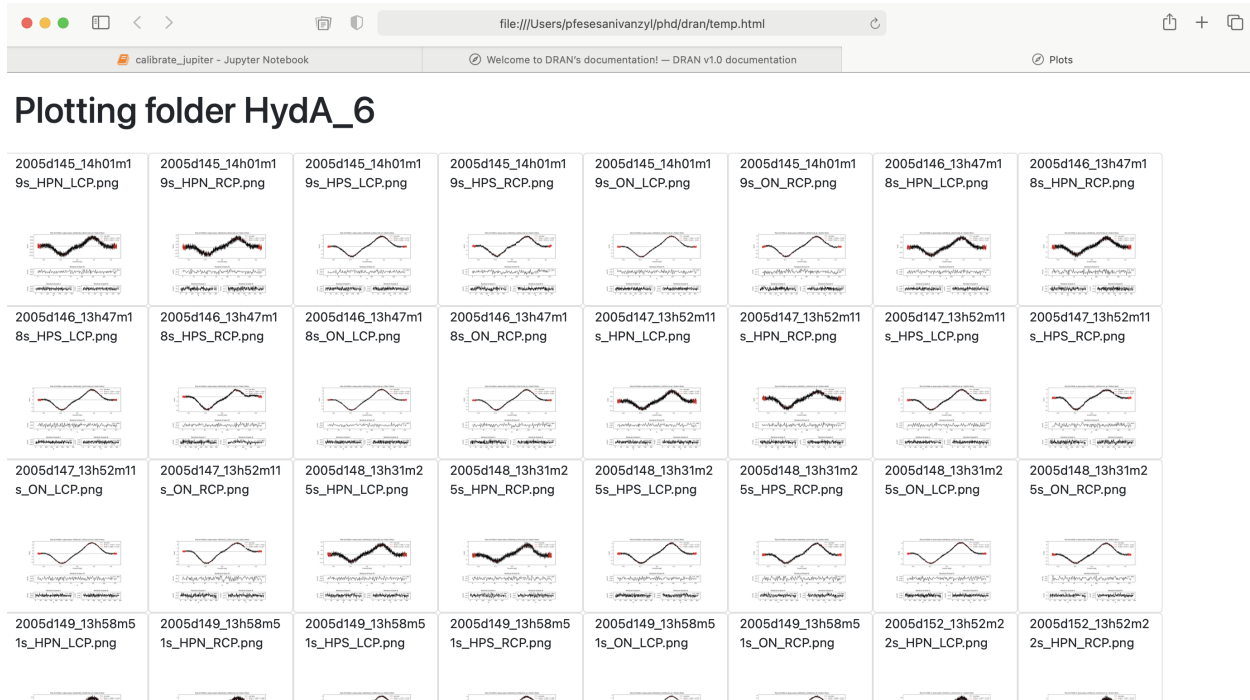
Note: In future releases the GUI will be adapted to also handle timeseries analysis.

3.9 View plots

View plots lets you view the thumbnails of all the plots made by the code for each observed object.



Selecting the object name from a dropdown list provides a tile view of all the plots made for that current object. Clicking on the tiles provides a zoomed in image of the thumbnail.



Currently this does not work well when there is a lot of data because the page takes a long time to load. In future work this will be modified for efficient page loading.

3.10 Radio frequency interference (RFI)

RFI is a problem for many astronomers.

3.11 Tutorials

The following tutorial will show you how to run a basic data reduction for a Hydra A 13cm / 2 GHz observation.

3.11.1 Data reduction of 2 GHz data.

First we need to read in data from the file we want to process

```
$ python dran.py -f test_data/HydraA_13NB/2011d285_04h55m29s_Cont_mike_HYDRA_A.fits
```

This will produce the following output

```

1 *****
2 # PROCESSING SOURCE:
3 *****
4 # File name: 2011d285_04h55m29s_Cont_mike_HYDRA_A.fits
5 # Object: HYDRA A
6 # Object type: CAL
7 # Central Freq: 2280.0
8 # Observed : 2011-10-12
9 *****
10
11 - Processing: ON_LCP
12
13
14 # No sidelobes detected
15
16
17 * Center of baseline blocks on left and right of peak:
18 min pos left: [-0.4041615] @ loc/s [160]
19 min pos right: [0.391495] @ loc/s [2497]
20 scan len: 2699
21
22 # Fit the baseline
23 *****
24
25 # Fit = 0.073x + (-0.034), rms error = 0.0292
26
27 # Fit the peak
28 *****
29
30 # Peak = 2.464 +- 0.030 [K]
```

(continues on next page)

(continued from previous page)

```

31 # S/N: 86.77
32
33 - Processing: ON_RCP
34
35
36 # No sidelobes detected
37
38
39 * Center of baseline blocks on left and right of peak:
40 min pos left: [-0.40816145] @ loc/s [153]
41 min pos right: [0.39016168] @ loc/s [2525]
42 scan len: 2731
43
44 # Fit the baseline
45 *****
46
47 # Fit = -0.000632x + (-0.0406), rms error = 0.0254
48
49 # Fit the peak
50 *****
51
52 # Peak = 2.493 +- 0.025 [K]
53
54
55 # S/N: 100.69

```

We are now going to breakdown the results returned

Lines 1 - 9 give us basic details on the source under observation. This includes the name of the file being processed, the object being observed, the type of object it is (CAL = Calibrator or TAR = Target), the observing frequency, as well as the date the source was observed.

Line 11 tells you the drift scan currently being processed, in this case its the LCP On scan drift scan.

Line 14 is a debugging output that lets you know if any large sidelobes were detected, these are sidelobes which are larger than half the peak maximum.

Once the data is loaded and prepped, the program begins processing the data. First it tries to correct or remove any drift in the data that may exists. Using a gradient descent type algorithm, the program fits a spline through the data and detects the location of the lowest minimum locations on either sides of the center of the drift scan.

Line 17 - 20 give us information on the positions selected as the local minimum points. An also gives the length of the scan.

4% of the scan length is then used as the number of points required to get enough data around the local minimum points in order to fit a polynomila through the data.

The equation of the line that is used to correct the drift in the data is then displayed in line 25.

3.12 Commands

The following commands are required when optiong to run [CLI](#) analysis.