Emission Line Redshift Estimator GUI Tutorial

B. Liu and R. Bordoloi

North Carolina State University

Updated Date: 6/7/2022

Table of Contents

GUI Environment Installation

1. Install or update `conda`
    a. Anaconda Installation
2. Create a conda environment and install dependencies
    a. Run conda create --name zgui --file requirements_simple.txt
    b. Activate the conda environment - `conda activate <env_name>`
    c. Check and manage all conda environments - `conda env list`
    d. We Strongly recommend that you install all packages via conda, and NEVER perform `“sudo pip”.`
    e. If PyQt is installed via both conda and pip, this usually leads to a very unstable GUI.
3. Clone rbcodes to your work directory
    a. Change to your target directory - `cd <your_directory>`
    b. Fetch rbcodes - `git clone https://github.com/rongmon/rbcodes.git`

The GUI itself is dependent on a successful installation of adding the rbcodes repository to your PYTHONPATH:

- Add rbcodes to PYTHONPATH in your OS
  E.g. for folks using unix and tcsh shell add this line to your .cshrc file:
  setenv PYTHONPATH PATH_TO_RBCODES_FOLDER

GUI Database Preparation

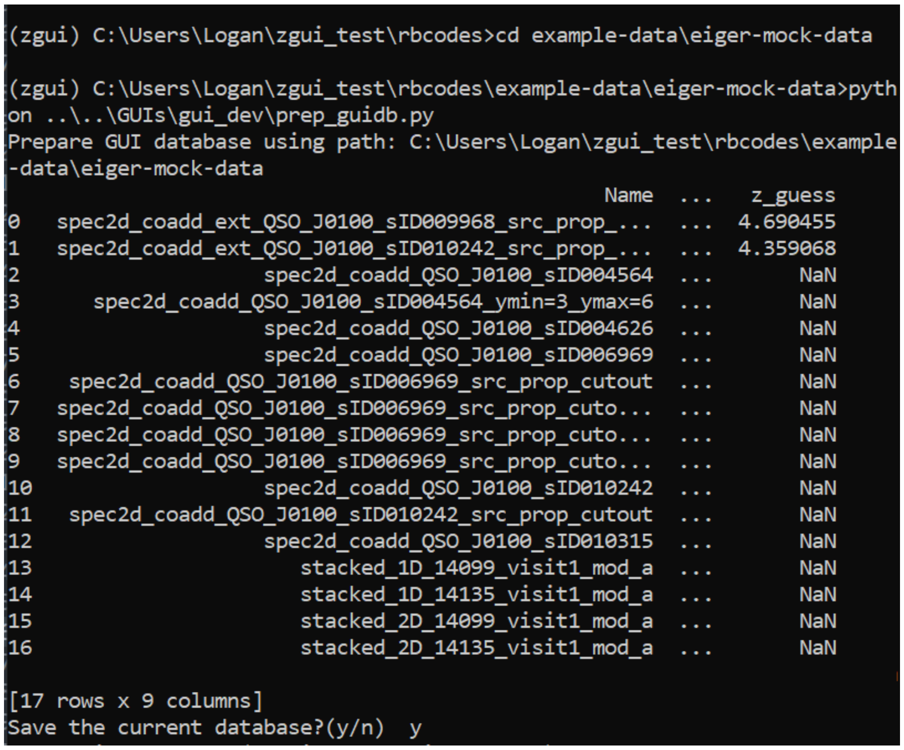
Once activating conda environment, change your current directory to the one that contains all FITS files you plan to work on using:

```
cd <path_to_your_FITS_file_directory>
```

and then run the following command to prepare GUI database:

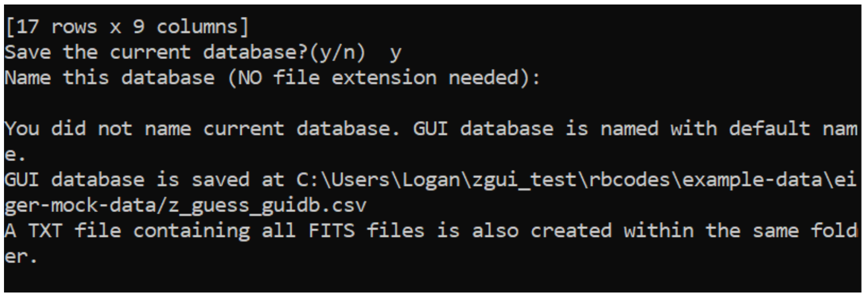
```
python <path_to_rbcodes>/rbcodes/GUIs/gui_dev/prep_guidb.py
```

If you compile successfully, a similar output should appear similar to the figure below.



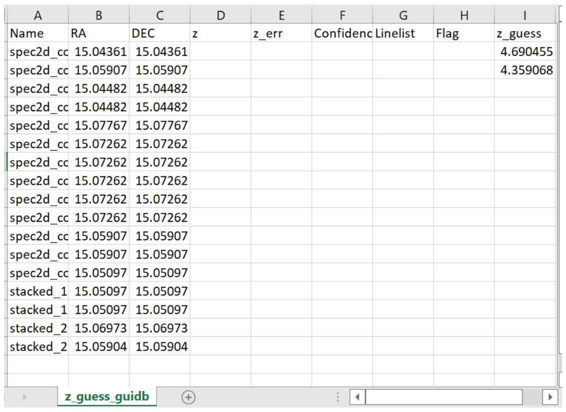
```
(zgui) C:\Users\Logan\zgui_test\rbcodes>cd example-data\eiger-mock-data

(zgui) C:\Users\Logan\zgui_test\rbcodes\example-data\eiger-mock-data>pyth
on ..\..\GUIs\gui_dev\prep_guidb.py
Prepare GUI database using path: C:\Users\Logan\zgui_test\rbcodes\example
-data\eiger-mock-data
                                          Name  ...   z_guess
0   spec2d_coadd_ext_QSO_J0100_sID009968_src_prop_...  ...  4.690455
1   spec2d_coadd_ext_QSO_J0100_sID010242_src_prop_...  ...  4.359068
2                    spec2d_coadd_QSO_J0100_sID004564  ...       NaN
3      spec2d_coadd_QSO_J0100_sID004564_ymin=3_ymax=6  ...       NaN
4                    spec2d_coadd_QSO_J0100_sID004626  ...       NaN
5                    spec2d_coadd_QSO_J0100_sID006969  ...       NaN
6    spec2d_coadd_QSO_J0100_sID006969_src_prop_cutout  ...       NaN
7   spec2d_coadd_QSO_J0100_sID006969_src_prop_cuto...  ...       NaN
8   spec2d_coadd_QSO_J0100_sID006969_src_prop_cuto...  ...       NaN
9   spec2d_coadd_QSO_J0100_sID006969_src_prop_cuto...  ...       NaN
10                   spec2d_coadd_QSO_J0100_sID010242  ...       NaN
11   spec2d_coadd_QSO_J0100_sID010242_src_prop_cutout  ...       NaN
12                   spec2d_coadd_QSO_J0100_sID010315  ...       NaN
13                      stacked_1D_14099_visit1_mod_a  ...       NaN
14                      stacked_1D_14135_visit1_mod_a  ...       NaN
15                      stacked_2D_14099_visit1_mod_a  ...       NaN
16                      stacked_2D_14135_visit1_mod_a  ...       NaN

[17 rows x 9 columns]
Save the current database?(y/n)  y
```

After a while, a compiled table will be displayed at your terminal and ask you if you want to save this database (y - yes; N - No). You may also name this database. If not, the database will automatically be saved with its default name (`z_guess_guidb.csv`).



```
[17 rows x 9 columns]
Save the current database?(y/n)  y
Name this database (NO file extension needed):

You did not name current database. GUI database is named with default nam
e.
GUI database is saved at C:\Users\Logan\zgui_test\rbcodes\example-data\ei
ger-mock-data/z_guess_guidb.csv
A TXT file containing all FITS files is also created within the same fold
er.
```

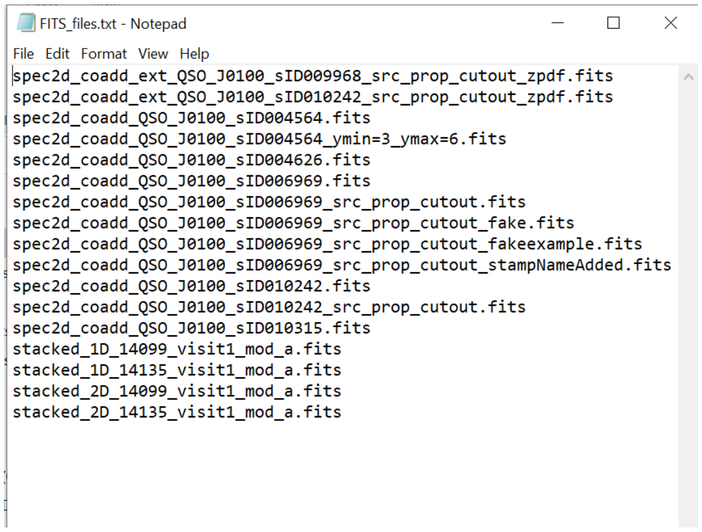
Note: The GUI database can only be saved as CSV file.



A	B	C	D	E	F	G	H	I
Name	RA	DEC	z	z_err	Confidenc	Linelist	Flag	z_guess
spec2d_cc	15.04361	15.04361						4.690455
spec2d_cc	15.05907	15.05907						4.359068
spec2d_cc	15.04482	15.04482						
spec2d_cc	15.04482	15.04482						
spec2d_cc	15.07767	15.07767						
spec2d_cc	15.07262	15.07262						
spec2d_cc	15.07262	15.07262						
spec2d_cc	15.07262	15.07262						
spec2d_cc	15.07262	15.07262						
spec2d_cc	15.07262	15.07262						
spec2d_cc	15.05907	15.05907						
spec2d_cc	15.05907	15.05907						
spec2d_cc	15.05097	15.05097						
stacked_1	15.05097	15.05097						
stacked_1	15.05097	15.05097						
stacked_2	15.06973	15.06973						
stacked_2	15.05904	15.05904						

z_guess_guidb

If you successfully save this database, a TXT file (named `FITS_files.txt`) with all filenames is also saved in the current directory, shown in the figure below. This can be used to preload everything into the GUI for a faster reading process.



FITS_files.txt - Notepad

File Edit Format View Help

```
spec2d_coadd_ext_QSO_J0100_sID009968_src_prop_cutout_zpdf.fits
spec2d_coadd_ext_QSO_J0100_sID010242_src_prop_cutout_zpdf.fits
spec2d_coadd_QSO_J0100_sID004564.fits
spec2d_coadd_QSO_J0100_sID004564_ymin=3_ymax=6.fits
spec2d_coadd_QSO_J0100_sID004626.fits
spec2d_coadd_QSO_J0100_sID006969.fits
spec2d_coadd_QSO_J0100_sID006969_src_prop_cutout.fits
spec2d_coadd_QSO_J0100_sID006969_src_prop_cutout_fake.fits
spec2d_coadd_QSO_J0100_sID006969_src_prop_cutout_fakeexample.fits
spec2d_coadd_QSO_J0100_sID006969_src_prop_cutout_stampNameAdded.fits
spec2d_coadd_QSO_J0100_sID010242.fits
spec2d_coadd_QSO_J0100_sID010242_src_prop_cutout.fits
spec2d_coadd_QSO_J0100_sID010315.fits
stacked_1D_14099_visit1_mod_a.fits
stacked_1D_14135_visit1_mod_a.fits
stacked_2D_14099_visit1_mod_a.fits
stacked_2D_14135_visit1_mod_a.fits
```

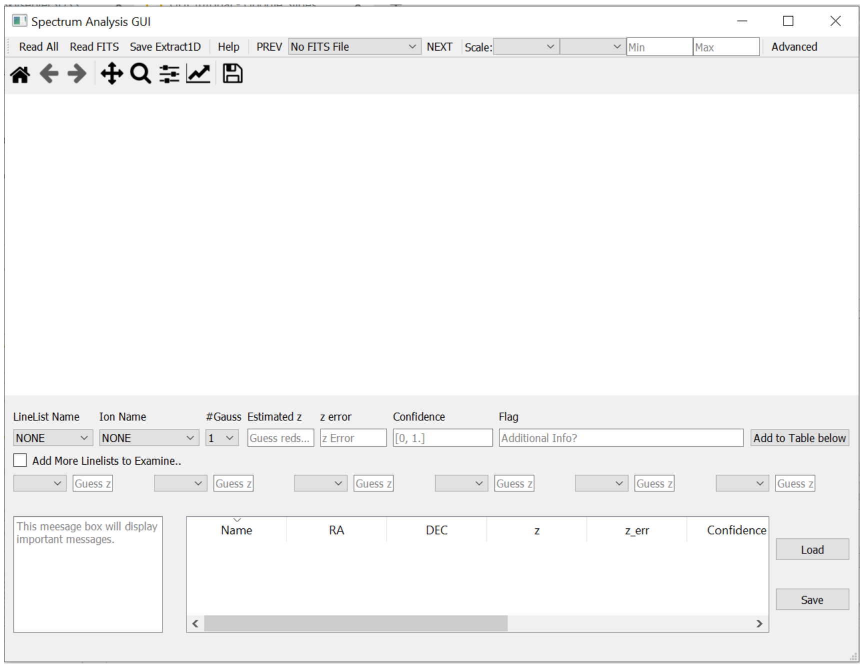
GUI Initialization

Easy initialization

Once activating conda environment, run `python <your_path_to_rbcodes>/rbcodes/GUIs/gui_dev/main.py` to open the GUI

Toggling your cursor to the toolbar buttons at the top will give you additional information shown at the bottom.

The figure below shows the GUI main window after launching.

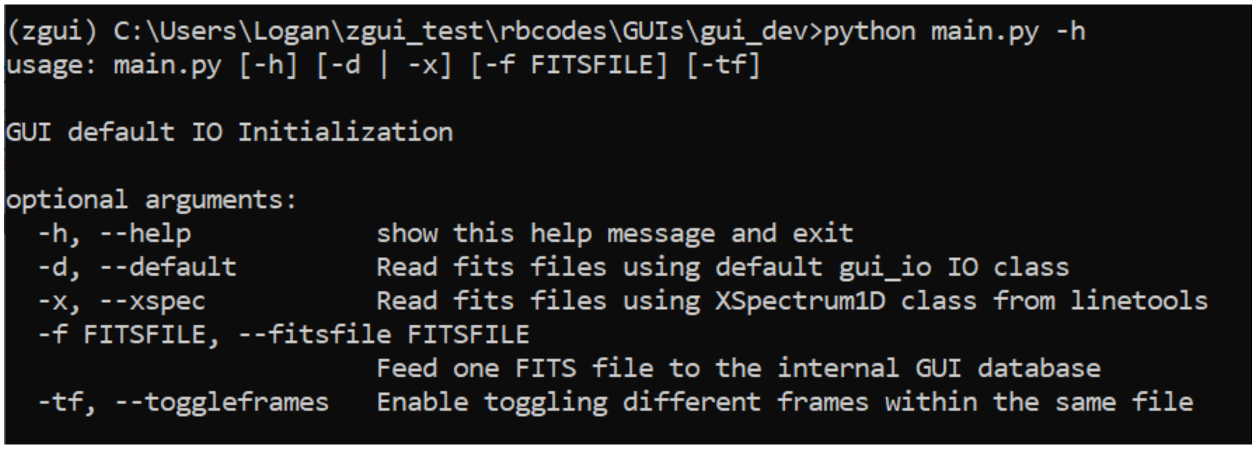


This message box will display important messages.

Initialization with flags

There are several flags (a.k.a, optional arguments) pre-defined within the GUI initialization serving for different usages. To check available flags, run
`python <your_path_to_rbcodes>/rbcodes/GUIs/gui_dev/main.py -h`, or
`python <your_path_to_rbcodes>/rbcodes/GUIs/gui_dev/main.py -help`

to see the help message, shown in the figure below.



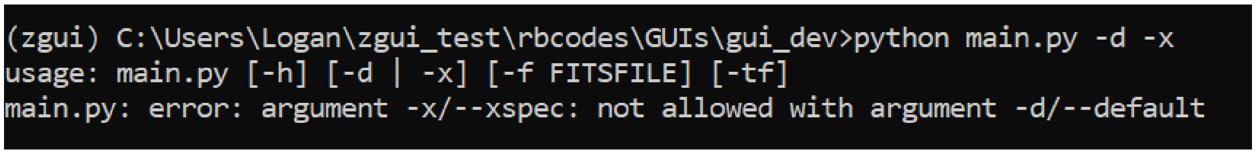
```
(zgui) C:\Users\Logan\zgui_test\rbcodes\GUIs\gui_dev>python main.py -h
usage: main.py [-h] [-d | -x] [-f FITSFILE] [-tf]

GUI default IO Initialization

optional arguments:
  -h, --help            show this help message and exit
  -d, --default         Read fits files using default gui_io IO class
  -x, --xspec           Read fits files using XSpectrum1D class from linetools
  -f FITSFILE, --fitsfile FITSFILE
                        Feed one FITS file to the internal GUI database
  -tf, --toggleframes   Enable toggling different frames within the same file
```

The default flag (-d, or –default) is optional for users to add. The GUI always initializes with its default IO class if users do not specify any flag. The default IO class can handle 1D and 2D spectra with pre-defined FITS format.

The xspec flag (-x, or –xspec) initializes the XSpectrum1D IO class from linetools repository. Note that the XSpectrum1D IO class is versatile enough to open various FITS files but it is not compatible with 2D spectra. Users should be aware of what types of files they want to read before launching the GUI with this flag. To initialize this setup, run

`python <your_path_to_rbcodes>/rbcodes/GUIs/gui_dev/main.py -x`, or
`python <your_path_to_rbcodes>/rbcodes/GUIs/gui_dev/main.py –xspec`

Note: The default flag and the xspec flag are mutually exclusive, meaning that users can only specify one of them at a time. Specifying both flags results in an argument error:

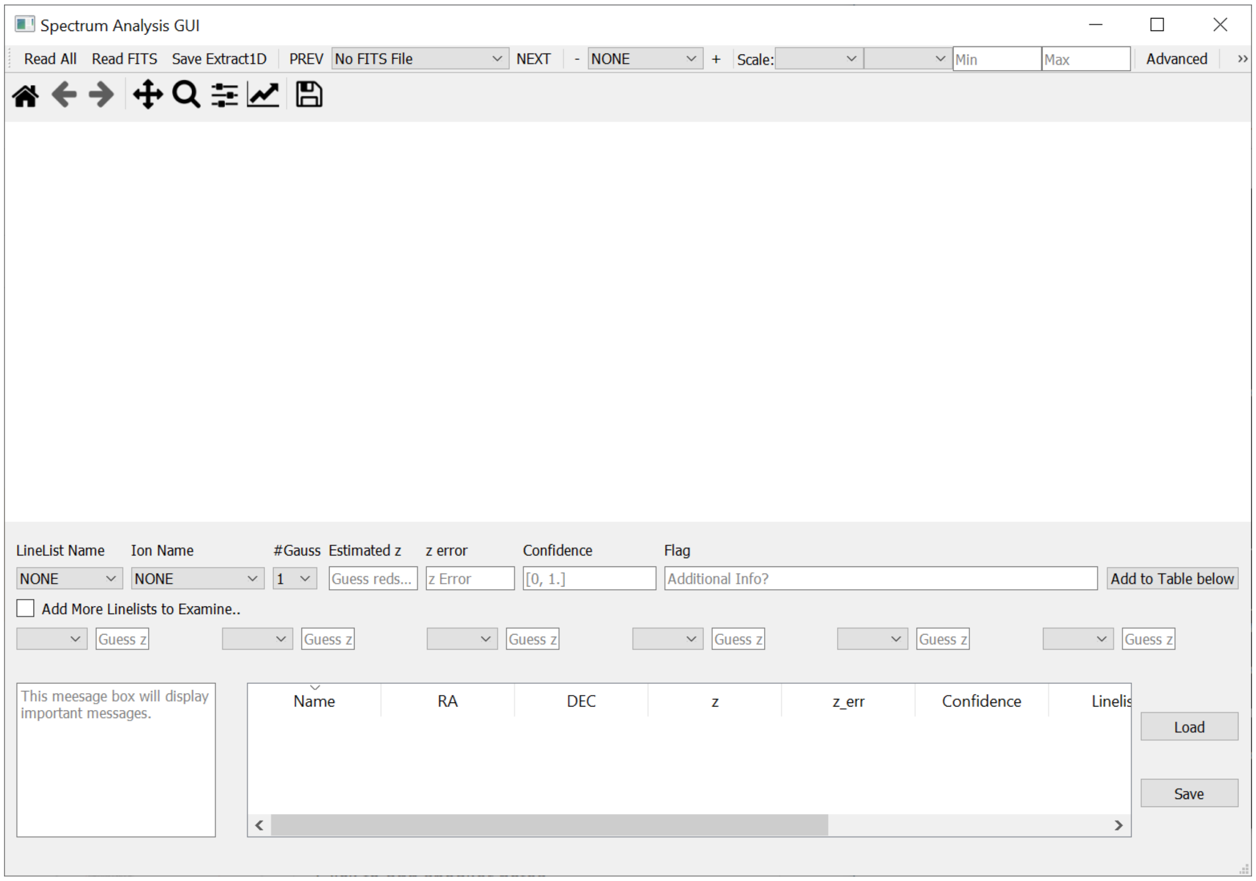


```
(zgui) C:\Users\Logan\zgui_test\rbcodes\GUIs\gui_dev>python main.py -d -x
usage: main.py [-h] [-d | -x] [-f FITSFILE] [-tf]
main.py: error: argument -x/--xspec: not allowed with argument -d/--default
```

The toogleframe flag (-tf, or –toogleframes) is designed specifically for 2D spectra and multi-frame FITS files. Users can switch between different frames with this GUI initialization. To use this setup, run

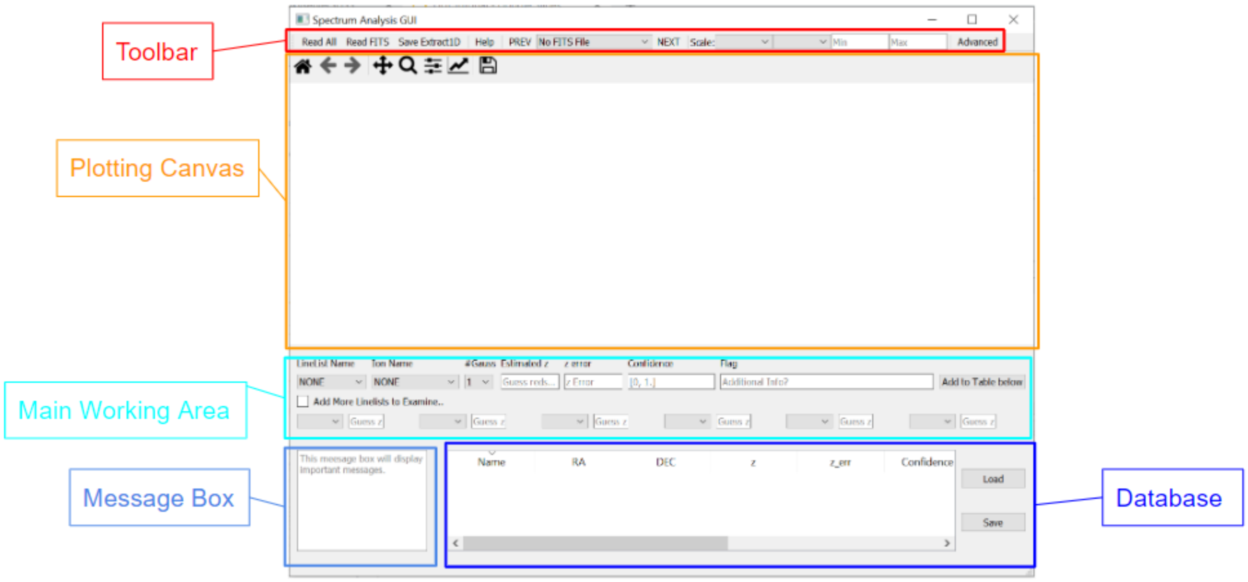
`python <your_path_to_rbcodes>/rbcodes/GUIs/gui_dev/main.py -tf`, or
`python <your_path_to_rbcodes>/rbcodes/GUIs/gui_dev/main.py –toggleframes`

Users should see that the layout in the toolbar is different from other initialization.

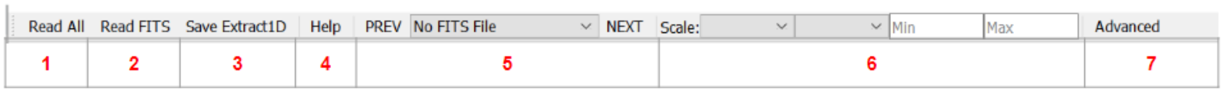


Window layout

The GUI is composed of several widgets, which are the toolbar, the main plotting canvas, the main working area, a message box, and a GUI database.



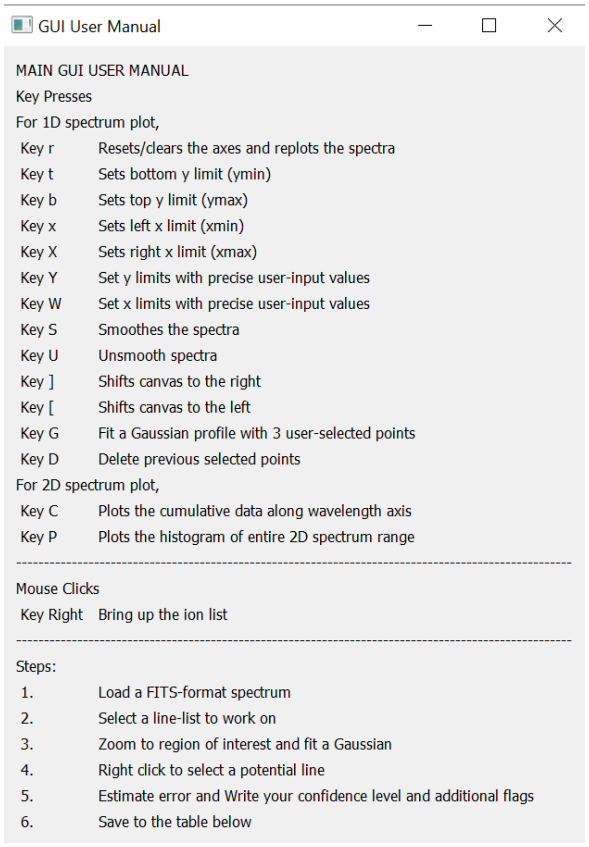
The toolbar widget is used for file manipulation. Users can read or save a FITS file, check the user manual, toggle different FITS files, toggle different frames (only working for initialization with -tf flag), apply simple image transformations on 2D spectra, and show the Advanced displays. The specific usage for each button is shown in the figure below.



1. **Read All** - read a txt file containing all FITS filenames on the backend and only read and plot the spectra when selecting one filename from the filename dropdown box.
2. **Read FITS** - read single/multiple fits files directly from the pop-up dialog window
3. **Save FITS** - save currently displayed spectrum to a unified-format fits file
4. **Help** - bring up user manual for keypresses and mouse-clicks
5. **Filename dropdown box** - All filenames currently loaded on the backend of the GUI; PREV/NEXT button for previous and next fits file loaded in GUI
6. **Scale** - Image visualization processing on 2D spectra only
7. **Advanced** - Additional Display for more information saved in FITS file

Note: GUI can read a fits file if and only if its format is compatible with what's implemented in GUI IO file.

User can click on the Help button to check the user manual. Make sure to click on the plotting canvas before pressing any key.



GUI User Manual

MAIN GUI USER MANUAL

Key Presses

For 1D spectrum plot,

Key r	Resets/clears the axes and replots the spectra
Key t	Sets bottom y limit (ymin)
Key b	Sets top y limit (ymax)
Key x	Sets left x limit (xmin)
Key X	Sets right x limit (xmax)
Key Y	Set y limits with precise user-input values
Key W	Set x limits with precise user-input values
Key S	Smoothes the spectra
Key U	Unsmooth spectra
Key]	Shifts canvas to the right
Key [	Shifts canvas to the left
Key G	Fit a Gaussian profile with 3 user-selected points
Key D	Delete previous selected points

For 2D spectrum plot,

Key C	Plots the cumulative data along wavelength axis
Key P	Plots the histogram of entire 2D spectrum range

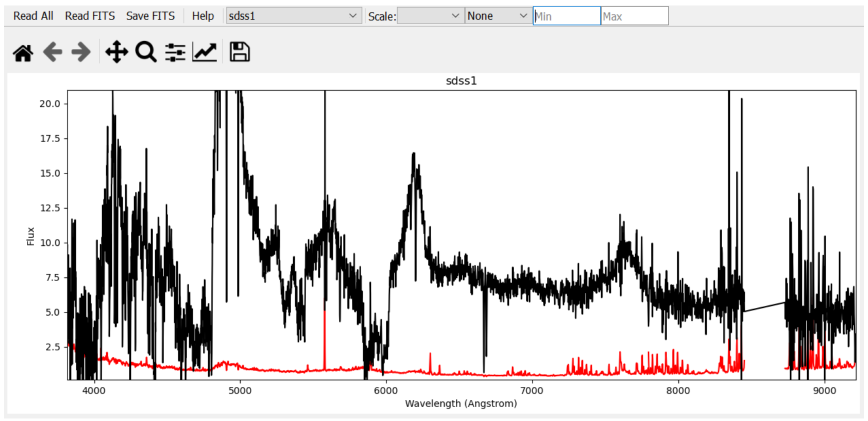
Mouse Clicks

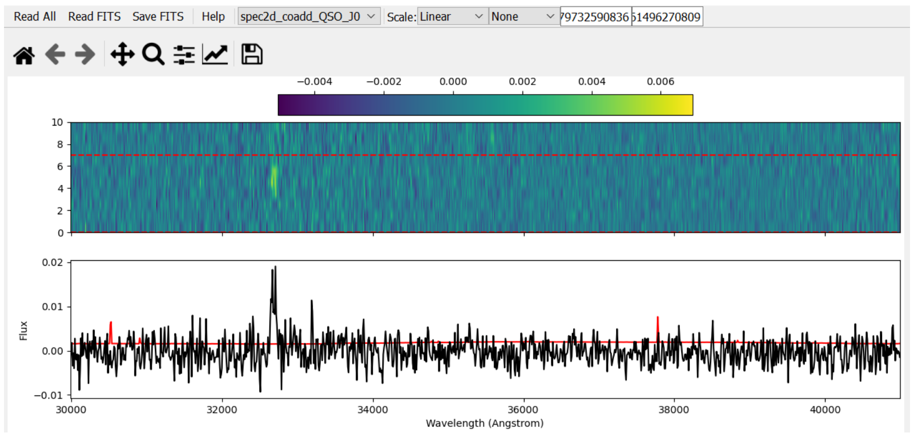
Key Right	Bring up the ion list

Steps:

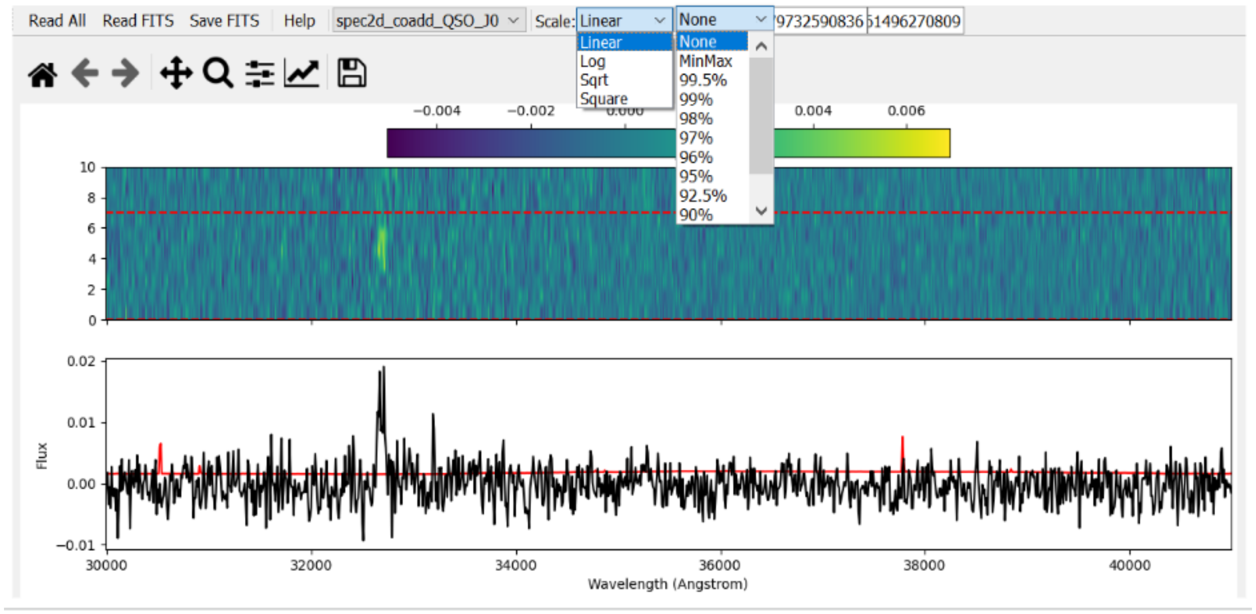
1. Load a FITS-format spectrum
2. Select a line-list to work on
3. Zoom to region of interest and fit a Gaussian
4. Right click to select a potential line
5. Estimate error and Write your confidence level and additional flags
6. Save to the table below

The plotting canvas widget in the GUI can automatically identify if the FITS file contains a 1D or 2D spectra and correctly display them. Note that the scaling feature for image transformations is not available for 1D spectra, and only reserved for 2D spectra. In the figure below, the top panel shows a FITS file only containing a 1D spectra, whereas the bottom panel shows an example of 2D spectra and the extracted 1D spectra corresponding to the extraction box defined within the GUI.

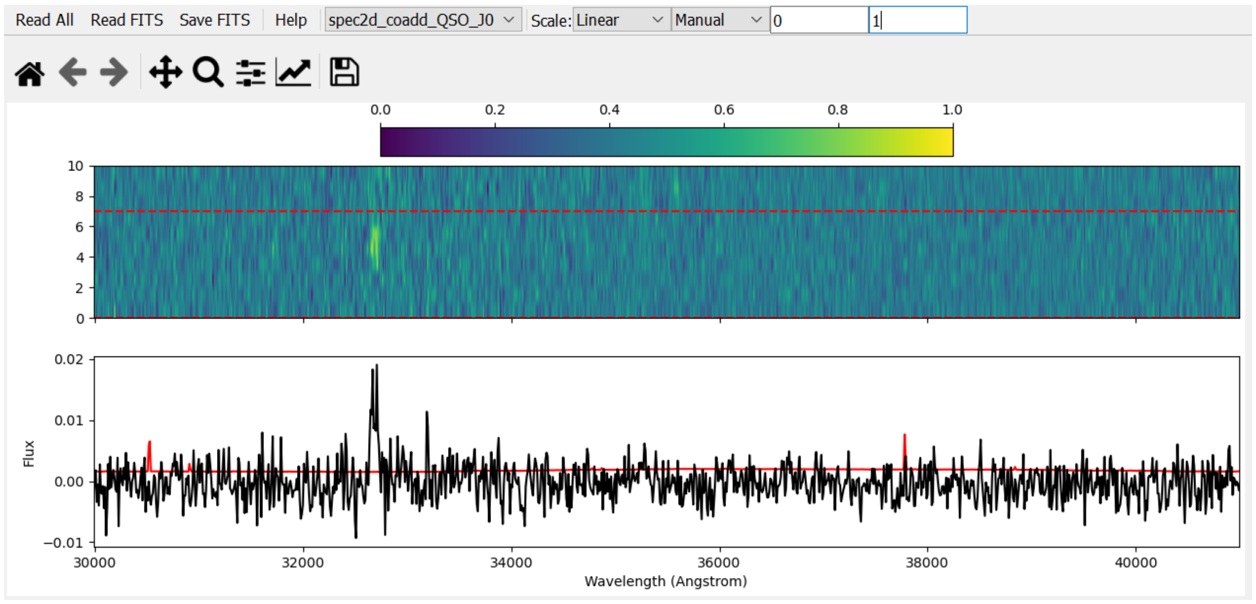




The GUI currently defined four image transformations, which are linear, logarithmic, square-root, and squared. The second drop-down box contains several percentage options for image normalization.



Users can also manually define the normalization range by selecting the manual option in the normalization box.


Read All
Read FITS
Save FITS
Help
spec2d_coadd_QSO_J0
Scale: Linear
Manual
0
1
0.0
0.2
0.4
0.6
0.8
1.0
10
8
6
4
2
0
0.02
0.01
0.00
−0.01
Flux
30000
32000
34000
36000
38000
40000
Wavelength (Angstrom)