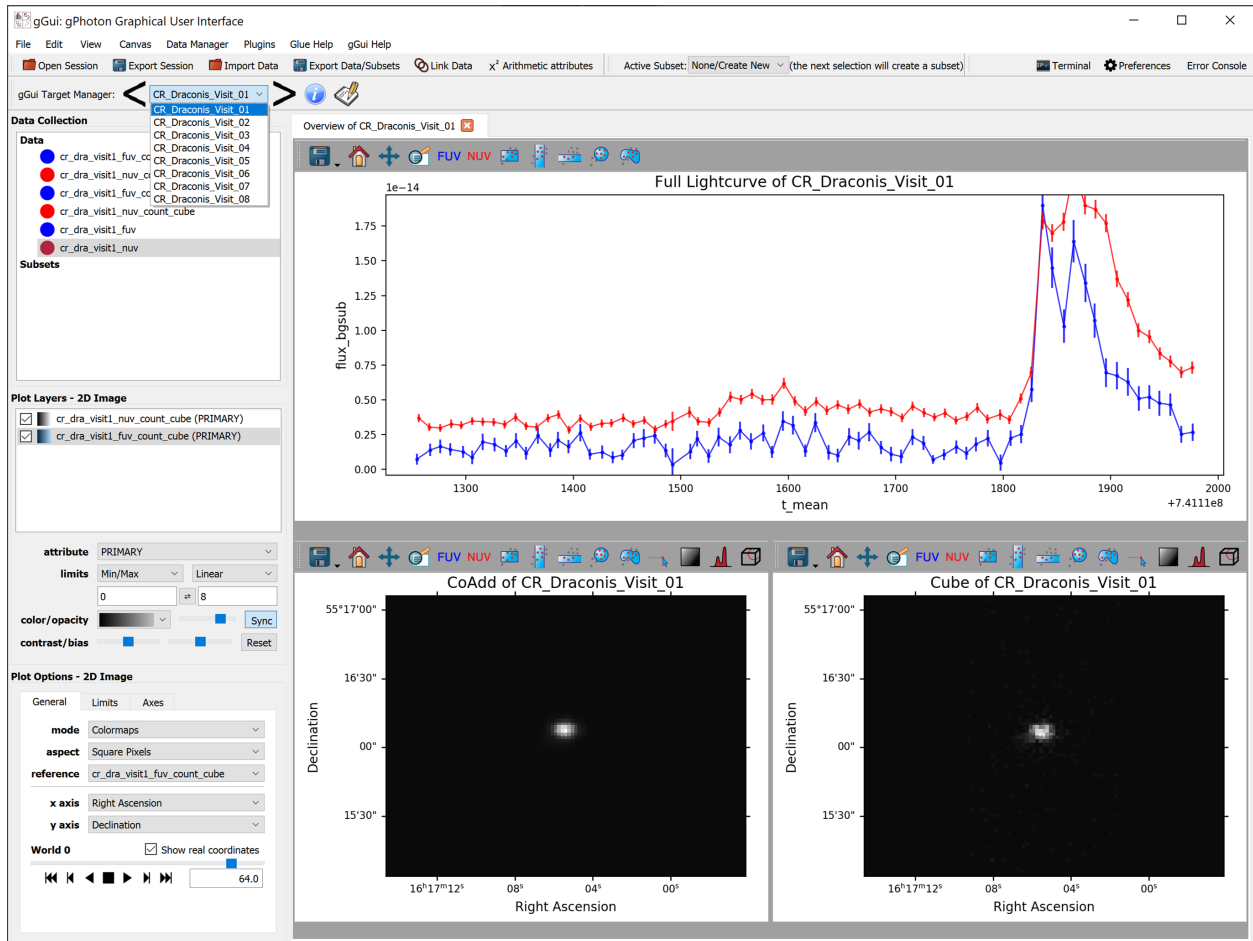

gGui

Release 1.2.0

Mar 23, 2020

1	About gGui	3
1.1	The Galaxy Evolution Explorer (GALEX) Space Telescope	3
1.2	gPhoton	3
2	Using gGui	5
2.1	Installing gGui	5
2.2	gGui Target Catalog	5
2.3	gGui Configuration File	6
2.4	Launching gGui	7
2.5	Demo Mode	8
2.6	The gGui Target Manager	8
2.7	The gGui Notepad	9
2.8	gGui Automatic Tabs	10
3	gGui Release List	13
3.1	gGui Release Version 1.2.0	13
3.2	gGui Release Version 1.1.0	14
4	The gGui Team	15
5	Known Issues with gGui	17
6	Indices and tables	19
7	Acknowledgements	21



gGui is a data-analysis package written atop the Glue Visualization Python library to visualize data from the GALEX Space Telescope data reduction pipeline, gPhoton.

gGui is currently actively developed and maintained by [Duy Nguyen](#) and [Scott Fleming](#)

The latest release is hosted on PyPI. To install, simply use pip! `pip install ggui`

gGui is a data-analysis software written to visualize astronomical data, specifically time-tagged lightcurves, coadds, and cubes. gGui serves as the primary data visualization tool for gPhoton, the data reduction pipeline for GALEX.

1.1 The Galaxy Evolution Explorer (GALEX) Space Telescope

GALEX was an ultraviolet space telescope commissioned by NASA and managed by the Jet Propulsion Laboratory (JPL) at the California Institute of Technology (CalTech).

The data currently resides in the [GALEX Public Archive](#) at the Space Telescope Science Institute's (STScI) Mikulski Archive for Space Telescopes (MAST).

1.2 gPhoton

gPhoton is a project to calibrate, archive, and make available every photon event observed by the GALEX mission in an effort to dramatically improve the flexibility and utility of the data with specific emphasis on enabling short time domain photometric studies.

2.1 Installing gGui

gGui is developed in the [Python language](#) and is hosted on [PyPI](#). To install, ensure you have [Python installed](#) (version 3.6 or greater with Tcl/Tk support) on your system and simply use pip:

```
pip install ggui
```

Warning: Ensure pip is upgraded by executing (Linux, macOS) *pip install --upgrade pip* or (Windows) *python -m pip install --upgrade pip*. We are aware of users with out of date versions of pip experiencing installation issues with a ‘numpy==1.13.3’ issue when installing one of our dependencies, ‘bottleneck’.

Alternatively you may download and install directly from our GitHub repository, again using pip:

```
git clone https://github.com/gphoton-tools/ggui.git
pip install ./ggui
```

Warning: Users who wish to install directly using our `setup.py` file should note a known bug and workaround on our [Known Issues page](#)

2.2 gGui Target Catalog

gGui ingests a list of targets and associated paths to data products.

To create this file, you can use our `make_param` utility. Alternatively, you can write this file yourself. This file is written in the YAML standard and expects the following format:

```
<Target Name>
  <Data Product Type>
    <Band>: <Absolute path to file>
```

for example:

```
Andromeda Galaxy
  lightcurve
    NUV: /home/ggui_data/andromeda_nuv_lightcurve.csv
    FUV: /home/ggui_data/andromeda_fuv_lightcurve.csv
  coadd
    NUV: /home/ggui_data/andromeda_nuv_coadd.fits
  cube
    NUV: /home/ggui_data/andromeda_nuv_cube.fits
    FUV: /home/ggui_data/andromeda_fuv_cube.fits
```

Paths can also be specified relatively:

```
Andromeda Galaxy
  lightcurve
    NUV: ../ggui_data/andromeda_nuv_lightcurve.csv
    FUV: ../ggui_data/andromeda_fuv_lightcurve.csv
```

gGui is cross platform! The following target catalog is identical to the above:

```
Andromeda Galaxy
  lightcurve
    NUV: ../ggui_data\andromeda_nuv_lightcurve.csv
    FUV: ../ggui_data/andromeda_fuv_lightcurve.csv
```

2.3 gGui Configuration File

gGui configures itself, and its environment, using a python configuration file: `ggui.conf`. This file is located with the source code (the result of `import ggui; import os; os.path.dirname(ggui.__file__)`) and should be modified to fit your needs. The configuration file contains two fields.

[Mandatory Fields] defines which fields will be assigned to which axes. gGui comes out-of-the-box configured for gPhoton lightcurves, coadds, and cubes:

```
[Mandatory Fields]
lightcurve_x = t_mean
lightcurve_y = flux_bgsub

coadd_x = Right Ascension
coadd_y = Declination

cube_x = Right Ascension
cube_y = Declination
```

If multiple bands are provided for each data product type (i.e. lightcurves, coadds, cubes), gGui will automatically attempt to **glue** these fields together.

[Additional Fields to Glue] defines additional fields gGui should associate, and therefore glue together. Multiple fields can be specified, comma delimited. Once again, gGui is configured by default for gPhoton data products:

```
[Additional Fields To Glue]
cube = World 0
lightcurve = t0,t1
```

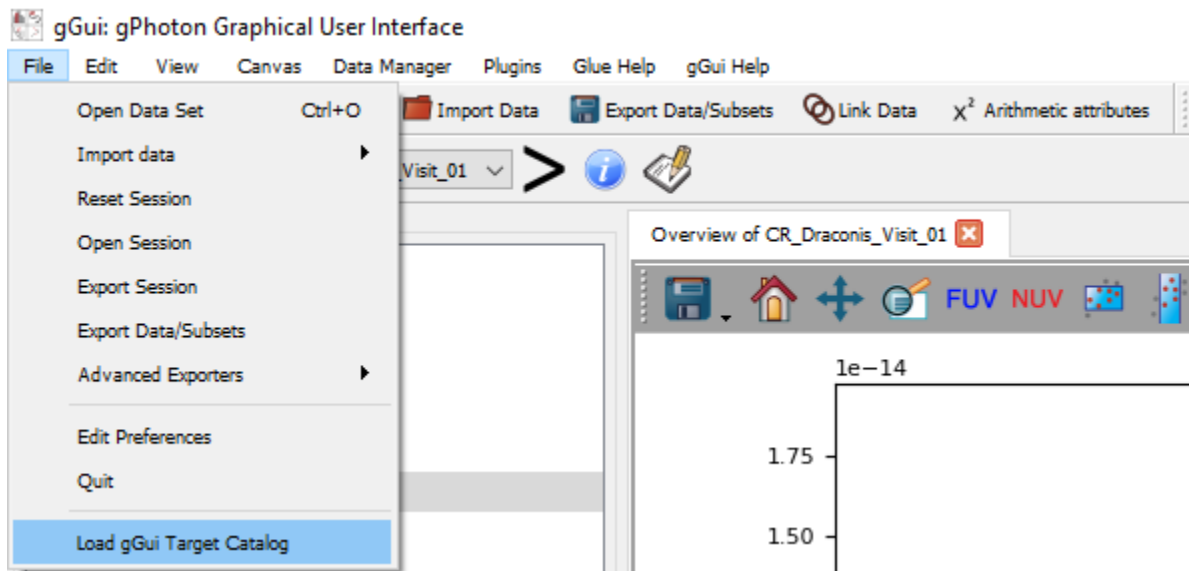
The default keyboard shortcuts for gGui may be modified here as well:

```
[Target Manager Shortcuts]
next_target = Right
previous_target = Left
```

2.4 Launching gGui

Once gGui is installed, gGui can be launched on the command line by invoking `ggui`.

Your gGui target catalog can be loaded either from within gGui or as a command line argument, assuming you have a properly formatted target list, and have configured the gGui fields of your data. From within gGui, you can use the gGui Target Catalog Selector under the “File” Menu:

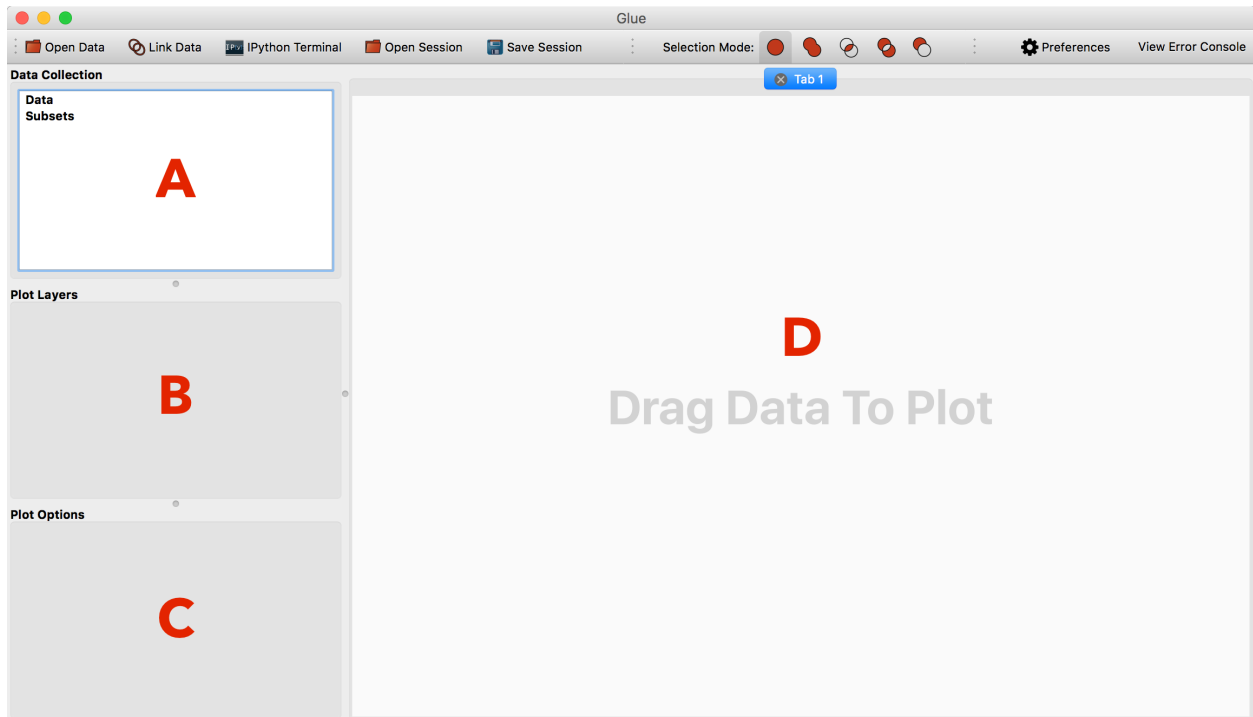


gGui can also be started with two independent flags:

- `ggui --target_list <path to gGui Target List>` will automatically load specified target list(s) into a new gGui session.
- `ggui --yaml_select` will prompt the user to select your target list(s) before gGui loads. After which, gGui will load these targets into a new gGui session.

Thirdly if you are in an IPython environment, you can invoke gGui’s `main()` function to use these flags as well: `from ggui import ggui; ggui.main(['--target_list', '<path to gGui Target List>'])`

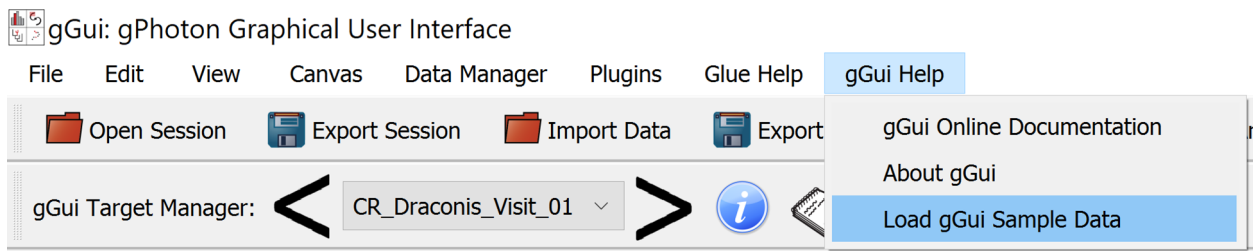
gGui is built atop the Glue Visualization Python Library. To learn more about the framework, please see [Glue’s Quickstart Guide](#):



gGui provides three major features atop Glue: The gGui Target Manager, gGui Notepad, and automatic tabview generation.

2.5 Demo Mode

If you're looking for a quick demo to try gGui, we have prepared some sample data for your to try out gGui's feature set. Simply select "Load gGui Sample Data" under the "gGui Help" menu.



2.6 The gGui Target Manager

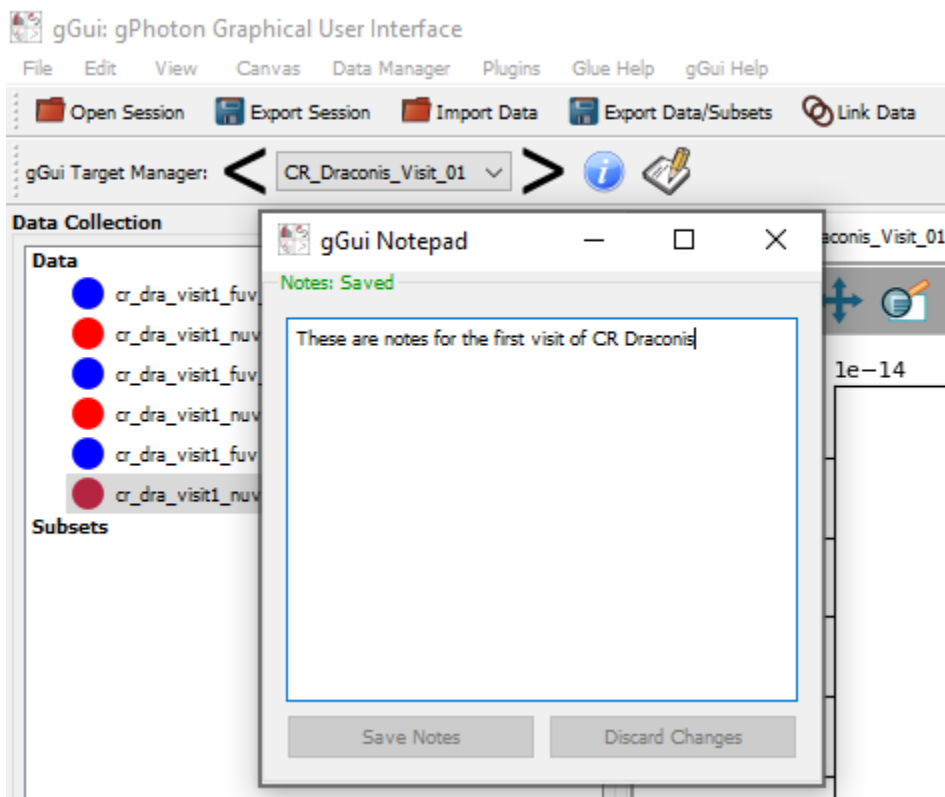
Accessed via the primary Glue toolbar in the upper left, the gGui Target Manager handles the loading of your target's data:



The Target Manager stores all of the targets identified by gGui from your target list(s). gGui only loads the data of the target selected, also known as *lazy evaluation*. One can advance targets by selecting the specific target from the dropdown list, or using the left/right arrows to advance to the previous/next target respectively. The information button will display the current target name and the parent gGui Target Catalog file this target originated from.

2.7 The gGui Notepad

The gGui Notepad features a collaborative space for teams to store observations, comments, and notes to each target. These notes are saved to the gGui target catalog. The gGui Notepad window can be launched via the notepad icon next to the Target Manager



The notepad displays the state of the notes above the editing window.

“Save Notes” will overwrite the notes on disk with those entered in the editing window.

“Discard Changes” will discard all changes in the editing window and restore the window to the notes stored on disk, in the gGui Target Catalog.

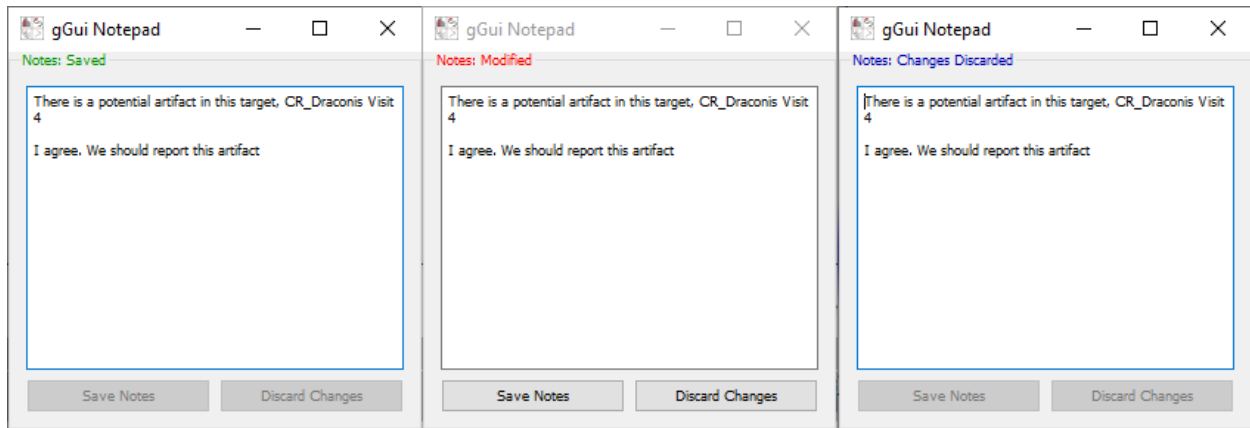


Fig. 1: The gGui Notepad will show the current state of the notes via three status: (a) Saved, (b) Modified and Unsaved, (c) Discarded

2.8 gGui Automatic Tabs

gGui currently automatically generates (1) tab:

- The Target Overview Tab

2.8.1 Overview Tab

The Overview Tab is currently the only tab implemented in gGui. It is intended to give a quick glance into the three data products (lightcurves, coadds, cubes) of all available bands simultaneously:

When the user selects one of the widgets, gGui will display the appropriate plot options and layers for that particular widget.

Each widget generated by gGui includes FUV and NUV toggle tools. If your data originates from gPhoton, or is specified to have FUV and/or NUV bands, you can toggle the on and off via these tools.

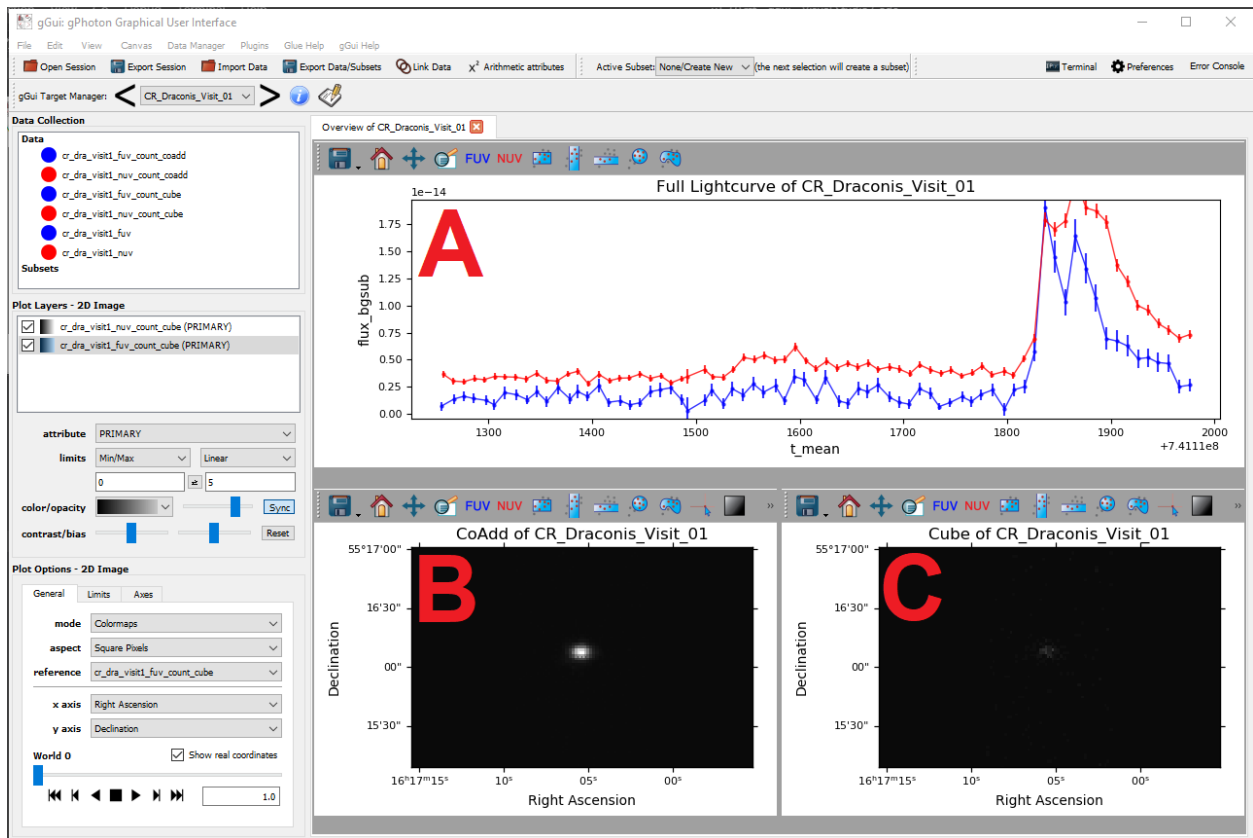


Fig. 2: The Overview Tab summarizing the target's (A) lightcurve, (B), Coadd, and (C) Cube, each with two bands (GALEX FUV, GALEX NUV)

3.1 gGui Release Version 1.2.0

Dear gGui users and colleagues,

I am excited to announce the release of gGui v1.2!

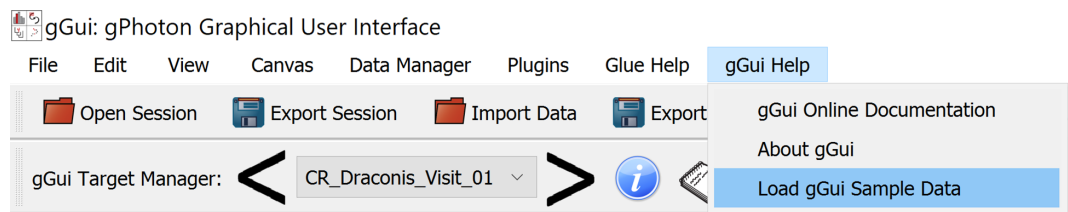
gGui is a data-analysis package written atop the Glue Visualization Python library to visualize data from the GALEX Space Telescope’s photon event archive, gPhoton.

I will be [presenting gGui](#) at the 235th American Astronomical Society conference in Honolulu, Hawai’i! We would love to chat with our users about anything gGui! Please visit us at [Poster Session 109 \(poster 14\)](#) on [2020 January 5th at 9:00AM HST!](#)

3.1.1 Notable changes

Version 1.2.0 includes improvements to many features of gGui:

1. The *gGui Target Catalog* is now cross-platform! gGui will automatically convert specified paths written on any OS into your native OS environment. This allows users to read target catalogs written by collaborators on a different OS, without any changes necessary to the gGui target catalog itself.
2. A quick demo mode has been introduced to gGui. To quickly download sample data we’ve prepared on our GitHub page, simply select “Load gGui Sample Data” under the “gGui Help” menu.



3. Plot options will automatically change as you click on a data product. This change deprecates the focus tool.

4. *The gGui Target Manager* now has configurable keyboard shortcuts! By default, the left and right arrows will advance targets for you! Users can configure these keyboard shortcuts in the *gGui Configuration File*
5. gGui now supports high DPI desktops.

gGui is hosted on PyPI. To update gGui, simply invoke pip: `pip install ggui --upgrade`

Please report any bugs and feature requests to our [GitHub page](https://github.com/gphoton-tools/ggui/issues): <https://github.com/gphoton-tools/ggui/issues>

As always, thank you for supporting gGui!

–Duy Nguyen

3.2 gGui Release Version 1.1.0

Dear gGui users and colleagues,

I am excited to announce the release of gGui v1.1!

gGui is a data-analysis package written atop the Glue Visualization Python library to visualize data from the GALEX Space Telescope data reduction pipeline, gPhoton.

3.2.1 Notable changes

Version 1.1.0 introduces the gGui Notepad as well as provides general improvements to the Target Manager and the startup process.

1. *The gGui Notepad* is a collaborative space for researchers to write notes about a particular target; these notes are stored in the gGui Target Catalog, allowing the sharing of notes between collaborators.
2. The gGui startup procedure has been enhanced. For more information, see *Launching gGui*.
 - Users may now also load a new gGui Target Catalog at runtime by selecting “File” > “Load gGui Target Catalog”
 - The `--target_list` command line argument now supports multiple gGui Target Catalogs, space separated
3. *The gGui Target Manager* has received a number of improvements:
 - Targets with the same name across multiple targets can be loaded and appropriately distinguished
 - Added an information button to display the current target’s gGui Target Catalog, from which it originated

Version 1.1.0 also resolves the bug of consecutive targets with differing data products. Targets with differing data products will now render correctly.

gGui is hosted on PyPI. To update gGui, simply invoke pip: `pip install ggui --upgrade`

Please report any bugs and feature requests to our [GitHub page](https://github.com/gphoton-tools/ggui/issues): <https://github.com/gphoton-tools/ggui/issues>

As always, thank you for supporting gGui!

–Duy Nguyen

CHAPTER 4

The gGui Team

Duy Nguyen is the Principal Software Architect and Developer of gGui. They work at Space Telescope Science Institute on the Data Analysis Tools Branch as a Software Engineer

- dnguyen@stsci.edu
- (667)218-6459
- 3700 San Martin Dr, Baltimore, MD 21218

Scott Fleming is the Principal Investigator of gGui. He works at the Space Telescope Science Institute as an Archive Scientist

Known Issues with gGui

If you run into any issues with gGui, please let us know by creating a GitHub issue! We'd rather an issue get reported multiple times than it not be reported at all! Listed below are known issues with gGui. Please reference this page if you are getting stuck.

- gGui abruptly crashes upon unrecognized field in [Additional Fields to Glue] heading in ggui.conf
- We are aware of installation issues for users who wish not to use pip and instead invoke our setup.py directly. To install ggui from source without using pip, please note a workaround we have found to be successful:

1. Clone source from GitHub into an empty directory: `git clone https://github.com/gphoton-tools/ggui.git`
2. `python ./setup.py install` from local git root
3. Modify qtpy to use PyQt5 instead of PySide
 - a. Determine qtpy initialization file from python interpreter: `import qtpy; qtpy.__file__`
 - b. Open this file and determine where the PYSIDE_API detection occurs (As of QtPy version 1.9.0, this occurs around line 202)
 - c. Modify the qtpy initialization file by adding `API = 'pyqt5'` to PySide detection. The file should read:

```
...
API = 'pyqt5' #Add this line
if API in PYSIDE_API:
    try:
        from PySide import __version__ as PYSIDE_VERSION #
↪analysis:ignore
...
```

4. Uninstall PyQt5 and PyQt5-sip:
 - a. `pip uninstall PyQt5`
 - b. `pip uninstall PyQt5-sip`

5. Reinstall PyQt5, which will automatically install PyQt5 as dependency: `pip install PyQt5`
6. You should now be able to start ggui normally

CHAPTER 6

Indices and tables

- `genindex`
- `modindex`
- `search`

CHAPTER 7

Acknowledgements

We would like to acknowledge Chase Million for producing and maintaining the gPhoton project, whose data gGui is targeted to process. We would also like to thank Alyssa Goodman, Thomas Robitaille, Christopher Beaumont, and the entire Glue team for their contributions to the Glue visualization tool, without whom gGui could not exist.