gPhoton Documentation

Release 1.28.2

Chase Million, Scott W. Fleming

February 28, 2017

Contents

In addition to the module-by-module documentation here, there is also a User Guide available with details on installation, use cases, input/output descriptions, and warnings/caveats.

Contents:

CHAPTER 1

gPipeline API

gPipeline.**check_args**(*args*)

Checks validity of command line arguments and, in some cases modifies them a little bit.

Parameters **args** (*argparse.ArgumentParser Namespace*) – The command-line arguments.

Returns argparse.ArgumentParser Namespace – The updated command-line arguments.

gPipeline.**gpipeline**(*raw6file*, *scstfile*, *band*, *outbase*, *aspfile*, *ssdfile*, *nullout*, *retries=100*)

Wrapper that calls the PhotonPipe method.

Parameters

- **raw6file** (*str*) – Name of the raw6 file to use.
- **scstfile** (*str*) – Spacecraft state file to use.
- **band** (*str*) – Name of the band to use, either 'FUV' or 'NUV'.
- **outbase** (*str*) – Base of the output file names.
- **aspfile** (*str*) – Name of aspect file to use.
- **ssdfile** (*str*) – Name of Stim Separation Data file to use.
- **nullfile** (*str*) – Name of output file to record NULL lines.
- **retries** (*int*) – Number of query retries to attempt before giving up.

gPipeline.**setup_parser**()

Defines command-line arguments.

Returns argparse.ArgumentParser Namespace – The command-line arguments.

CHAPTER 2

gFind API

gFind.**check_args**(*args*, *iam='gfind'*, *allow_no_coords=False*)

Checks validity of command line arguments and, in some cases, modifies them a little bit.

Parameters

- **args** (*argparse.ArgumentParser Namespace*) – The command-line arguments.
- **iam** (*str*) – The name of the method being called.
- **allow_no_coords** (*bool*) – If True, do not require coordinates to be provided.

Returns argparse.ArgumentParser Namespace – The updated command-line arguments.

gFind.**gfind**(*band='both'*, *detsize=1.1*, *exponly=False*, *gaper=False*, *maxgap=1500.0*, *minexp=1.0*, *quiet=False*, *retries=100*, *skypos=None*, *trange=None*, *verbose=0*, *skyrange=None*)

Primary program in the module. Prints time ranges to the screen and returns the total exposure time as a float.

Parameters

- **band** (*str*) – The band being used, either 'BOTH', 'FUV', or 'NUV'.
- **detsize** (*float*) – Effective diameter, in degrees, of the field-of-view.
- **exponly** (*bool*) – If True, only report the exposure times.
- **gaper** (*bool*) – Return time ranges in a format that can be copy-pasted as a valid gAperture call.
- **maxgap** (*float*) – Maximum gap size, in seconds, for data to be considered contiguous.
- **minexp** (*float*) – Minimum gap size, in seconds, for data to be considered contiguous.
- **quiet** (*bool*) – If True, don't print anything to STDOUT. Overrides verbose.
- **retries** (*int*) – Number of query retries to attempt before giving up.
- **skypos** (*list*) – The right ascension and declination, in degrees.
- **trange** (*list*) – Minimum and maximum time range to make a light curve, in GALEX time.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **skyrange** (*list*) – RA and Dec extents, in degrees, defining the lengths of sides of a box on the sky that circumscribes the region of interest.

Returns dict – The available data for the requested band(s).

gFind.**setup_parser**(*iam='gfind'*, *parser=None*)

If a parser object is not provided, make one here.

Parameters

- **iam** (*str*) – The name of the method being called.
- **parser** (*argparse.ArgumentParser Namespace*) – The command-line arguments.

Returns argparse.ArgumentParser Namespace – The command-line arguments.

CHAPTER 3

gMap API

gMap.**check_args**(*args*, *iam='gmap'*)

Checks validity of command line arguments and, in some cases modifies them a little bit.

Parameters

- **args** (*argparse.ArgumentParser Namespace*) – The command-line arguments.
- **iam** (*str*) – The name of the method being called.

Returns argparse.ArgumentParser Namespace – The updated command-line arguments.

gMap.**gmap**(*band*, *cntfile=None*, *coadd=None*, *detsize=1.1*, *intfile=None*, *skypos=None*, *maxgap=1500.0*, *memlight=100.0*, *minexp=1.0*, *overwrite=False*, *retries=100*, *skyrange=None*, *stepsz=0.0*, *trange=None*, *verbose=0*, *cntcoaddfile=False*, *intcoaddfile=False*)

Use a mix of strings (if we want to make an output file) and Booleans (False if we do not). I don't think this is the best way (I'd rather have separate variables for the True/False to create an image, and a string with the name of the output image), but for now this is kept since this is how the code was originally written.

Parameters

- **band** (*str*) – The band being used, either 'FUV' or 'NUV'.
- **cntfile** (*str*) – Name of the count file to make.
- **coadd** (*bool*) – If true, create a coadd image using all available data within the specified time range (don't use time bins).
- **detsize** (*float*) – Effective diameter, in degrees, of the field-of-view.
- **intfile** (*str*) – Name of intensity file to make.
- **skypos** (*list*) – The right ascension and declination, in degrees.
- **maxgap** (*float*) – Maximum gap size, in seconds, for data to be considered contiguous.
- **memlight** (*float*) – If specified, breaks the queries into sections of this many seconds.
- **minexp** (*float*) – Minimum gap size, in seconds, for data to be considered contiguous.
- **overwrite** (*bool*) – If True, overwite an existing output file.
- **retries** (*int*) – Number of query retries to attempt before giving up.
- **skyrange** (*list*) – RA and Dec extents, in degrees, that define the lengths of edges of a box on the sky, centered at skypos, that defines the sky region of interest.
- **stepsz** (*float*) – The size of the time bins to use, in seconds.

- **trange** (*list*) – Minimum and maximum time range to make images for, in GALEX time.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **cntcoaddfile** (*str*) – Name of the count coadd file to make.
- **intcoaddfile** (*str*) – Name of the intensity coadd file to make.

gMap.**setup_parser**(*iam='gmap'*)

Defines command line arguments.

Parameters **iam** (*str*) – The name of the method being called.

Returns argparse.ArgumentParser Namespace – The command-line arguments.

CHAPTER 4

gAperture API

gAperture.**check_annulus**(*args*)
Checks and formats the annulus values.

Parameters **args** (*argparse.ArgumentParser Namespace*) – The command-line arguments.

Returns argparse.ArgumentParser Namespace – The updated command-line arguments.

gAperture.**check_args**(*args*, *iam='gaperture'*)

Checks validity of command line arguments and, in some cases modifies them a little bit.

Parameters

- **args** (*argparse.ArgumentParser Namespace*) – The command-line arguments.
- **iam** (*str*) – The name of the method being called.

Returns argparse.ArgumentParser Namespace – The updated command-line arguments.

gAperture.**check_radius**(*args*)
Checks the radius value.

Parameters **args** (*argparse.ArgumentParser Namespace*) – The command-line arguments.

Returns argparse.ArgumentParser Namespace – The updated command-line arguments.

gAperture.**gaperture**(*band*, *skypos*, *radius*, *csvfile=None*, *annulus=None*, *coadd=False*, *stepsz=False*, *verbose=0*, *overwrite=False*, *trange=None*, *tranges=None*, *minexp=1.0*, *maxgap=1500.0*, *iocode='w'*, *detsize=1.1*, *minimal_output=False*, *photoncsvfile=None*)

Creates a light curve and returns the data in a python dict() and as a CSV file, if outfile is specified. Can be called from the interpreter.

Parameters

- **band** (*str*) – The band being used, either 'FUV' or 'NUV'.
- **skypos** (*list*) – The right ascension and declination, in degrees.
- **radius** (*float*) – The radius of the photometric aperture, in degrees.
- **csvfile** (*str*) – Name of the photon event CSV file to use for lightcurve.
- **annulus** (*list*) – Radii of the inner and outer of an annulus, in degrees, within which to measure the background.

- **coadd** (*bool*) – Set to True if calculating a total flux instead of flux from each time bin.
- **stepsz** (*float*) – The size of the time bins to use, in seconds.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **overwrite** (*bool*) – If True, overwite an existing output file.
- **trange** (*list*) – Minimum and maximum time range to make a light curve, in GALEX time seconds.
- **tranges** (*list*) – Set of time ranges to query within, in GALEX time seconds.
- **minexp** (*float*) – Minimum gap size, in seconds, for data to be considered contiguous.
- **maxgap** (*float*) – Maximum gap size, in seconds, for data to be considered contiguous.
- **iocode** (*str*) – The code to use when writing the output file.
- **detsize** (*float*) – Effective diameter, in degrees, of the field-of-view.
- **minimal_output** (*bool*) – If True, produce an output file with a minimum number of columns.
- **photoncsvfile** (*str*) – Name of the photon event CSV file to use for photons.

Returns dict – The light curve, including input parameters.

gAperture.**reconstruct_command**(*args*)

Rebuild the equivalent command line for this call to gAperture.

Parameters **args** (*argparse.ArgumentParser Namespace*) – The command-line arguments.

Returns str – The equivalent command-line for this call to gAperture.

gAperture.**setup_file**(*args*)

If requested, create a header for the CSV that includes the column names and a reconstruction of the command line call.

Parameters **args** (*argparse.ArgumentParser Namespace*) – The command-line arguments.

Returns argparse.ArgumentParser Namespace – The updated command-line arguments.

gAperture.**setup_parser**(*iam='gaperture'*)

Defines command line arguments.

Parameters **iam** (*str*) – The name of the method being called.

Returns argparse.ArgumentParser Namespace – The command-line arguments.

gAperture.**stamp**(*args*)

Creates a jpeg preview image stamp of the targeted region.

Parameters **args** (*argparse.ArgumentParser Namespace*) – The command-line arguments.

CHAPTER 5

Other Module API's

CalUtils

gPhoton.CalUtils.**avg_stimpos**(*band*, *eclipse*)
Define the mean detector stim positions.

Parameters

- **band** (*str*) – The band to return the average stim positions for, either 'FUV' or 'NUV'.
- **eclipse** (*int*) – The eclipse number to return the average stim positions for.

Returns dict – A dict containing the average x and y positions of the four stims (for the requested band).

gPhoton.CalUtils.**clk_cen_scl_slp**(*band*, *eclipse*)

Return the detector clock, center, scale, and slope constants. These are empirically determined detector-space calibration parameters that help define the relationship between raw event data and its physical position on the detector.

Parameters

- **band** (*str*) – The band to return the constants for, either 'FUV' or 'NUV'.
- **eclipse** (*int*) – The eclipse number to return the constants for.

Returns tuple – a tuple containing the x-clock, y-clock, x-center, y-center, x-scale, y-scale, x-slope, and y-slope constants.

gPhoton.CalUtils.**compute_stimstats**(*raw6file*, *band*, *eclipse*)
Computes statistics for stim events for the post-CSP stim-based correction.

Parameters

- **raw6file** (*str*) – The name of the raw6 FITS file to read.
- **band** (*str*) – The band to return the stim data for, either 'FUV' or 'NUV'.
- **eclipse** (*int*) – The eclipse number to return the stim data for.

Returns tuple – Six-element tuple containing information on the trend in relative positions of the stims over time used for the Post-CSP stim correction.

gPhoton.CalUtils.**create_ssd**(*raw6file*, *band*, *eclipse*, *ssdfile=None*)
Creates a Stim Separation Data (SSD) table file.

Parameters

- **raw6file** (*str*) – The name of the raw6 FITS file to read.
- **band** (*str*) – The band to create the SSD file for, either 'FUV' or 'NUV'.
- **eclipse** (*int*) – The eclipse number to create the SSD file for.
- **ssdfile** (*str*) – Name of stim separation data (SSD) output file to create.

Returns tuple – 2xN tuple containing slope and intercept of stim positions over time.

gPhoton.CalUtils.**deadtime_fromlist**(*photonfile, t0, t1, band, method=0, colnames=['t', 'x', 'y', 'xa', 'ya', 'q', 'xi', 'eta', 'ra', 'dec', 'flags']*)

Given a photon event list, calculate the deadtime using the specified method.

Parameters

- **photonfile** (*str*) – Name of the file containing photon events.
- **t0** (*float*) – The mininum of the time range to calculate the deadtime within.
- **t1** (*float*) – The maximum of the time range to calculate the deadtime within.
- **band** (*str*) – The band to return the constants for, either 'FUV' or 'NUV'.
- **method** (*int*) – which deadtime_method[012] to use when calculating deadtime
- **colnames** (*list*) – The column names within the photon event file.

Returns float or numpy.ndarray – The deadtime within the requested time range. If using Method 0, return value is a float, otherwise, return value is an array containing the deadtimes per bin.

gPhoton.CalUtils.**deadtime_method0**(*data*, *t0*, *t1*, *band*, *feeclkratio=0.966*, *tec2fdead=5.52e-06*)

Given a dict() containing 't', a list of global event times, computes the deadtime using an empirical formula based on global count rate over the whole time range.

Parameters

- **data** (*dict*) – The event times from which to calculate the deadtime.
- **t0** (*float*) – The mininum of the time range to calculate the deadtime within.
- **t1** (*float*) – The maximum of the time range to calculate the deadtime within.
- **band** (*str*) – The band to return the constants for, either 'FUV' or 'NUV'. Not actually used in this function, but retained to make this function definition consistent with deadtime_method[12].
- **feeclkratio** (*float*) – Ratio of Front End Electronics clock rates.
- **tec2fdead** (*float*) – The nominal amount of time following an event that the detector is unable to detect another event.

Returns float – The deadtime for the specified time range.

gPhoton.CalUtils.**deadtime_method1**(*data*, *t0*, *t1*, *band*, *feeclkratio=0.966*, *tec2fdead=5.52e-06*, *tstep=1.0*)

Given a dict() containing 't', a list of global event times, computes the deadtime using an empirical formula based on global count rates, put into bins of depth equal to *tstep* seconds and averaged.

Parameters

- **data** (*dict*) – The event times and detector positions from which to calculate the deadtime.

- **t0** (*float*) – The mininum of the time range to calculate the deadtime within.
- **t1** (*float*) – The maximum of the time range to calculate the deadtime within.
- **band** (*str*) – The band to return the constants for, either 'FUV' or 'NUV'. Not actually used in this function, but retained to make this function definition consistent with deadtime_method[02].
- **feeclkratio** (*float*) – Ratio of Front End Electronics clock rates.
- **tec2fdead** (*float*) – The nominal amount of time following an event that the detector is unable to detect another event.
- **tstep** (*float*) – Bin size (in seconds) to use when reporting the deadtime.

Returns numpy.ndarray – The deadtimes within the specified time range, split into the requested bin size.

gPhoton.CalUtils.**deadtime_method2**(*data, t0, t1, band, refrate=79.0, feeclkratio=0.966, tstep=1.0, refrange=[0.4, 2.0]*)

Given a list of global event times, computes the deadtime through direct comparison of the stim rate to the reference rate in bins of depth *tstep* seconds, and trimmed of outliers. This is close to the deadtime method used by the mission pipeline.

Parameters

- **data** (*dict*) – The event times and 'x' and 'y' detector positions from which to calculate the deadtime.
- **t0** (*float*) – The mininum of the time range to calculate the deadtime within.
- **t1** (*float*) – The maximum of the time range to calculate the deadtime within.
- **band** (*str*) – The band to return the constants for, either 'FUV' or 'NUV'.
- **refrate** (*float*) – The reference rate of stims in counts/sec.
- **feeclkratio** (*float*) – Ratio of Front End Electronics clock rates.
- **tstep** (*float*) – Bin size (in seconds) to use when reporting the deadtime.
- **refrange** (*list*) – minimum and maximum multiplicative of the reference stim rate that will be considered a legitimate / valid measurement

Returns numpy.ndarray – The deadtimes within the specified time range, split into the requested bin size.

gPhoton.CalUtils.**find_fuv_offset**(*scstfile*)

Computes NUV->FUV center offset based on a lookup table.

Parameters **scstfile** (*str*) – Name of the spacecraft state (-scst) FITS file.

Returns tuple – A two-element tuple containing the x and y offsets.

gPhoton.CalUtils.**find_stims**(*t*, *x*, *y*, *band*, *eclipse*)

Returns t,x,y and identity (1-4) of likely stims in the input arrays.

Parameters

- **t** (*numpy.ndarray*) – Photon event times within which to search for stim events.
- **x** (*numpy.ndarray*) – Photon event detector x positions within which to search for stim events.
- **y** (*numpy.ndarray*) – Photon event detector y positions within which to search for stim events.

- **band** (*str*) – The band to return the constants for, either 'FUV' or 'NUV'.
- **eclipse** (*int*) – The eclipse number to return the constants for.

Returns tuple – A four-element tuple containing arrays of the times, detector x, detector y, and stim ID (1, 2, 3 or 4) of the likely stim events.

gPhoton.CalUtils.**find_stims_index**(*x*, *y*, *band*, *eclipse*, *margin=90.001*)

Given a list of detector x,y positions of events, returns four arrays that contain the indices of likely stim events for that stim, i.e., the first array contains positions for stim1, the second array has positions of stim2, etc.

Example of how the return indexes are used: x[index1], y[index1] would give all of the event positions for stim1.

Parameters

- **x** (*list*) – Detector 'x' positions to identify likely stim events from.
- **y** (*list*) – Detector 'y' positions to identify likely stim events from.
- **band** (*str*) – The band to return the constants for, either 'FUV' or 'NUV'.
- **eclipse** (*int*) – The eclipse number to return the average stim positions for.
- **margin** (*float*) – +/- extent in arcseconds defining search box

Returns tuple – A four-element tuple containing arrays of indexes that correpond to the event positions for stim1, stim2, stim3, and stim4.

gPhoton.CalUtils.**get_stim_coefs**(*ssdfile*)

Computes the stim scaling coefficients based on a ssdfile (which contains information on the on-detector spatial separation of stims as a function of time).

Parameters **ssdfile** (*str*) – The name of the Stim Separatation Data (SSD) file.

Returns tuple – A two-element tuple containing the stim scaling coefficients.

gPhoton.CalUtils.**post_csp_caldata**()

Loads the calibration data for after the CSP event.

Returns tuple – A six-element tuple containing the wiggle, walk, and clock corrections. See the calibration paper for details.

gPhoton.CalUtils.**raw6_to_stims**(*raw6file*, *band*, *eclipse*, *margin=90.001*)

Extracts stim events from a raw6 file.

Parameters

- **raw6file** (*str*) – The name of the raw6 FITS file to read.
- **band** (*str*) – The band to return the stim data for, either 'FUV' or 'NUV'.
- **eclipse** (*int*) – The eclipse number to return the stim data for.
- **margin** (*float*) – +/- extent in arcseconds defining stim search box

Returns tuple – A four-element tuple containing data from each stim. The data from each stim are stored in dicts.

gPhoton.CalUtils.**rtaph_yac**(*yactbl*, *ya*, *yb*, *yamc*, *eclipse*)

Compute a the YAmC (yac) correction to the Y detector FEE position.

Parameters

- **yactbl** (*numpy.ndarry*) – yac correction lookup table
- **ya** (*numpy.ndarray*) – Y axis wiggle.
- **yb** (*numpy.ndarray*) – Y axis coarse clock.
- **yamc** (*numpy.ndarray*) – Raw Y detector position in FEE pixels.
- **eclipse** (*int*) – The eclipse number to return the constants for.

Returns numpy.ndarray

gPhoton.CalUtils.**rtaph_yac2**(*q*, *xb*, *yb*, *ya*, *y*, *aspum*, *wig2*, *wig2data*, *wlk2*, *wlk2data*, *clk2*, *clk2data*)

Compute a secondary correction to the YAmC (yac) detector FEE Y position.

Parameters

- **q** (*numpy.ndarray*) – Detected pulse height
- **xb** (*numpy.ndarray*) – X axis coarse clock.
- **yb** (*int*) – Y axis coarse clock.
- **ya** (*int*) – Y axis wiggle.
- **y** (*numpy.ndarray*) – Detector y position.
- **aspum** (*float*) – Detector arcseconds per micrometer.
- **wig2** (*numpy.ndarray*) – Secondary wiggle correction lookup table.
- **wig2data** (*dict*) – Secondary wiggle reference values.
- **wlk2** (*numpy.ndarray*) – Secondary walk correction lookup table.
- **wlk2data** (*dict*) – Secondary walk correction reference values.
- **clk2** (*numpy.ndarray*) – Secondary clock correction lookup table.
- **clk2data** (*dict*) – Secondary clock correction reference values.

Returns numpy.ndarray – Secondary YAmC corrections.

gPhoton.CalUtils.**rtaph_yap**(*ya*, *yb*, *yamc*)

For post-CSP data, 'wrap' the YA value for YA in [0,1]. From rtaph.c.

Parameters

- **ya** (*numpy.ndarray*) – Y axis wiggle.
- **yb** (*numpy.ndarray*) – Y axis coarse clock.
- **yamc** (*numpy.ndarray*) – Raw Y detector position in FEE pixels.

Returns numpy.ndarray

gPhoton.CalUtils.**stimcount**(*data*, *band*, *t0*, *t1*, *margin=90.001*)

Given a dict() that contains a list of 't' event times and 'x' and 'y' detector positions, returns the total number of stim events within a time range.

Parameters

- **data** (*dict*) – The event times and detector positions from which to count the total number of stim events.
- **band** (*str*) – The band to return the constants for, either 'FUV' or 'NUV'.
- **t0** (*float*) – The mininum of the time range to report stim counts within.

- **t1** (*float*) – The maximum of the time range to report stim counts within.
- **margin** (*float*) – +/- extent in arcseconds defining search box

Returns int – The total number of stim events.

gPhoton.CalUtils.**totalcount**(*data*, *t0*, *t1*)

Given a dict() containg 't', a list of global even times, return the total number of events within the time range.

Parameters

- **data** (*dict*) – The event times and detector positions from which to count the total number of photon events.
- **t0** (*float*) – The mininum of the time range to report total counts within.
- **t1** (*float*) – The maximum of the time range to report total counts within.

Returns int – The total number of events within the given time range.

gPhoton.CalUtils.**xieta2colrow**(*xi*, *eta*, *detsize*, *fill*, *npixx*, *npixy*)

Convert detector xi and eta values to image column and row.

Parameters

- **xi** (*numpy.ndarray*) – One coordinate of the unprojected detector position.
- **eta** (*numpy.ndarray*) – Second coordinate of the unprojected detector position.
- **fill** (*float*) – Ration of the detector extent to the image extent.
- **npixx** (*int*) – Number of pixels in the x dimension of an image.
- **nipxy** – Number of pixels in the y dimension of an image.

Returns tuple – A tuple containing the image column and row values.

CalibrationTools

gPhoton.CalibrationTools.**compute_deadtime**(*t, x, y, band, eclipse, trange=[[], []]*)

Uses multiple methods to estimate the detector deadtime correction. The deadtime is an estimate of the fraction of time that the detector was unable to register new events because it was in the middle of readout. Deadtime should, therefore, not count as true exposure time.

Parameters

- **t** (*numpy.ndarray*) – Set of photon event times.
- **x** (*numpy.ndarray*) – Set of photon event x positions.
- **y** (*numpy.ndarray*) – Set of photon event y positions.
- **band** (*str*) – The band being used, either 'FUV' or 'NUV'.
- **eclipse** (*int*) – The eclipse number the data are taken from.
- **trange** (*list*) – A 2xN list of minimum and maximum times that define the ranges in which to calculate the dead time. If not supplied, the range is defined as the minimum and maximum times of the photon events.

Returns numpy.ndarray – The dead time (dt), as a percentage (0<dt<1), during the entire observation or per time bin (if trange is defined).

gPhoton.CalibrationTools.**compute_exposure**(*t, x, y, flags, band, eclipse, trange=[[], []]*)

Computes the effective, or 'true', exposure for the given data.

Parameters

- **t** (*numpy.ndarray*) – Set of photon event times.
- **x** (*numpy.ndarray*) – Set of photon event x positions.
- **y** (*numpy.ndarray*) – Set of photon event y positions.
- **flags** (*numpy.ndarray*) – Set of flags for each photon event.
- **band** (*str*) – The band being used, either 'FUV' or 'NUV'.
- **eclipse** (*int*) – The eclipse number the data are taken from.
- **trange** (*list*) – A 2xN list of minimum and maximum times that define the ranges in which to calculate the dead time. If not supplied, the range is defined as the minimum and maximum times of the photon events.

Returns numpy.ndarray – The effective exposure time, in seconds, during the specified time range(s).

gPhoton.CalibrationTools.**compute_shutter**(*t, trange=[[], []]*)

Computes the detector shutter correction. The shutter correction accounts for short periods of time when no events were registered by the detector for any number of reasons. Any gap of longer than 0.05 seconds does not count as true exposure time.

Parameters

- **t** (*numpy.ndarray*) – Set of photon event times.
- **trange** (*list*) – A 2xN list of minimum and maximum times that define the ranges in which to calculate the dead time. If not supplied, the range is defined as the minimum and maximum times of the photon events.

Returns float – The total time lost due to shutter, i.e., the sum of all time periods >=0.05 seconds during which no events are registered by the detector.

gPhoton.CalibrationTools.**create_rr**(*csvfile*, *band*, *eclipse*, *aspfile=None*, *expstart=None*, *expend=None*, *retries=20*, *detsize=1.25*, *pltscl=68.754932*)

DEPRECATED: Creates a relative response map for an eclipse, given a photon list.

Parameters

- **csvfile** (*str*) – Name of CSV file containing photon events.
- **band** (*str*) – The band being used, either 'FUV' or 'NUV'.
- **eclipse** (*int*) – The eclipse number the data are taken from.
- **aspfile** (*str*) – The aspect file to use with the CSV file.
- **expstart** (*float*) – Start of the exposure, in seconds.
- **expend** (*float*) – End of the exposure, in seconds.
- **retries** (*int*) – Number of query retries before giving up.
- **detsize** (*float*) – Effective size of the detector, in degrees.

- **pltscl** (*float*) – The plate scale in arcseconds.

Returns tuple – A two-element tuple containing the relative response and the effective exposure time. The relative response is a 2D array of values.

gPhoton.CalibrationTools.**load_txy**(*csvfile*)
Loads just the t,x,y columns from a photon CSV file.

Parameters **csvfile** (*str*) – Name of the photon event CSV file to read.

Returns tuple – A four-element tuple containing arrays with the times, x position, y position, and flags from the photon event CSV file.

gPhoton.CalibrationTools.**write_int**(*cntfile*, *rrhrfile*, *oldint*, *outfile*)

Writes out an intensity (int) map given a count (cnt) and a high resolution relative response (rrhr).

Parameters

- **cntfile** (*str*) – Name of count map (-cnt) FITS file.
- **rrhrfile** (*str*) – Name of the High Resolution Relative Response FITS file.
- **oldint** (*str*) – Name of Intensity file (nominally from the mission pipeline) from which to pull header information. (for comparisons)
- **outfile** (*str*) – Name of the output FITS file to create.

gPhoton.CalibrationTools.**write_rr**(*csvfile*, *band*, *eclipse*, *rrfile*, *outfile*, *aspfile=None*, *expstart=None*, *expend=None*, *retries=20*)

Creates a relative response map for an eclipse, given a photon list file, and writes it to a FITS file.

Parameters

- **csvfile** (*str*) – Name of CSV file containing photon events.
- **band** (*str*) – The band being used, either 'FUV' or 'NUV'.
- **eclipse** (*int*) – The eclipse number the data are taken from.
- **rrfile** (*str*) – The name of an existing rrhr FITS frile from which to steal header information about exposure time. (For comparing outputs.)
- **outfile** (*str*) – The name of the output relative response FITS file to make.
- **aspfile** (*str*) – The aspect file to use with the CSV file.
- **expstart** (*float*) – Start of the exposure, in seconds.
- **expend** (*float*) – End of the exposure, in seconds.
- **retries** (*int*) – Number of query retries before giving up.

gPhoton.CalibrationTools.**write_rrhr**(*rrfile*, *rrhrfile*, *outfile*)

Turns a relative response (rr) into a high resolution relative response (rrhr) file with interpolation.

Parameters

- **rrfile** (*str*) – Relative Response FITS file to get header information from.
- **rrhrfile** (*str*) – High Resolution Relative Response FITS file to get header information from.
- **outfile** (*str*) – Name of the output FITS file to create.

FileUtils

gPhoton.FileUtils.**create_ssd_filename**(*band*, *eclipse*)
Returns the Stim Separation Data (SSD) calibration file name.

Parameters

- **band** (*str*) – The band to create the SSD for, either 'FUV' or 'NUV'.
- **eclipse** (*int*) – The eclipse number to create the SSD file for.

Returns str – The name of the SSD file to create.

gPhoton.FileUtils.**load_aspect**(*aspfile*)
Loads a set of aspect files into a bunch of arrays.

Parameters **aspfile** (*list*) – List of aspect files (+paths) to read.

Returns tuple – Returns a six-element tuple containing the RA, DEC, twist (roll), time, header, and aspect flags. Each of these EXCEPT for header, are returned as numpy.ndarrays. The header is returned as a dict containing the RA, DEC, and roll from the headers of the aspec files in numpy.ndarrays.

gPhoton.FileUtils.**load_raw6**(*raw6file*)
Reads a raw6 file. Just wraps some pyfits I/O commands.

Parameters **raw6file** (*str*) – Name of the raw6 file to read.

Returns tuple – A two-element tuple containing the header of the first extension, and the HDUList object returned from astropy.io.fits.

gPhoton.FileUtils.**web_query_aspect**(*eclipse*, *retries=20*)
Grabs the aspect data from MAST databases based on eclipse.

Parameters

- **eclipse** (*int*) – The number of the eclipse to retrieve aspect files for.
- **retries** (*int*) – The number of times to retry a query before giving up.

Returns tuple – Returns a six-element tuple containing the RA, DEC, twist (roll), time, header, and aspect flags. Each of these EXCEPT for header, are returned as numpy.ndarrays. The header is returned as a dict containing the RA, DEC, and roll from the headers of the aspec files in numpy.ndarrays.

MCUtils

gPhoton.MCUtils.**angularSeparation**(*ra1*, *dec1*, *ra2*, *dec2*)

Compute angular separation in degrees of points on the sky. It is important, especially for small angular separations, that the values for ra[01] and dec[01] have precision of float64 or better. Now uses the haversine formula which is stable for small angles.

Parameters

- **ra1** (*float*) – The right ascension of the first coordinate.
- **dec1** (*float*) – The declination of the first coordinate.
- **ra2** (*float*) – The right ascension of the second coordinate.
- **dec2** (*float*) – The declination of the second coordinate.

Returns float – The angular separation, in degrees, on the sky.

gPhoton.MCUtils.**area**(*radius*)
Returns the area of a circle with a given radius.

Parameters **radius** (*float*) – The radius of the cricle.

Returns float – The area of the circle.

gPhoton.MCUtils.**distance**(*a*, *b*, *c*, *d*)
Computes Euclidean distance between [a,b] and [c,d].

Parameters

- **a** (*float*) – x-coordinate of first data point.
- **b** (*float*) – y-coordinate of first data point.
- **c** (*float*) – x-coordinate of second data point.
- **d** (*float*) – y-coordinate of second data point.

Returns float – The Euclidean distance between the two points.

gPhoton.MCUtils.**find_nearest_lower**(*array*, *value*)

Finds the index of the value in the array that is closest without going

over. This method assumes that:

1. 'value' is within the range of 'array'.
2. 'array' is ordered.
3. 'array' has no gaps.

Parameters

- **array** (*numpy.ndarray*) – Array of values to search.
- **value** (*float*) – Value to find the closest match without going over.

Returns int – The index of the array element closest to value without going over.

gPhoton.MCUtils.**get_fits_data**(*filename*, *dim=0*, *verbose=0*)
Reads FITS data. A wrapper for common pyfits commands.

Parameters

- **filename** (*str*) – The name of the FITS file to retrieve the data from.
- **dim** (*int*) – The extension to retrieve the data from, 0=Primary, 1=First Extension, etc.
- **verbose** (*int*) – If > 0, print messages to STDOUT.

Returns Data instance – The data from the 'dim' HDU.

gPhoton.MCUtils.**get_fits_header**(*filename*)
Reads a FITS header. A wrapper for common astropy.io.fits commands.

Parameters **filename** (*str*) – The name of the FITS file to retrieve the header from.

Returns Header instance – The header from the primary HDU.

gPhoton.MCUtils.**get_tbl_data**(*filename*, *comment='|'*)
Reads data from a table into a numpy array.

Parameters

- **filename** (*str*) – The name of the FITS file to read.
- **comment** (*str*) – The symbol that represents a comment.

Returns numpy.ndarray – The table data.

gPhoton.MCUtils.**manage_requests2**(*query*, *maxcnt=100*, *wait=1*, *timeout=60.0*, *verbose=0*)
Make simple 'requests' calls more robust against network issues.

Parameters

- **query** (*str*) – The URL containing the query.
- **maxcnt** (*int*) – The maximum number of attempts to make before failure.
- **wait** (*int*) – The length of time to wait before attempting the query again. Currently a placeholder.
- **timeout** (*float*) – The length of time to wait for the server to send data before giving up, specified in seconds.
- **verbose** (*int*) – If > 0, print additional messages to STDOUT. Higher value represents more verbosity.

Returns requests.Response or None – The response from the server. If the query does not receive a response, returns None.

gPhoton.MCUtils.**print_inline**(*text*, *blanks=60*)
For updating text in place without a carriage return.

Parameters

- **text** (*str*) – Message to print to standard out.
- **blanks** (*int*) – Number of white spaces to prepend to message.

gPhoton.MCUtils.**rms**(*data*)
Return the root-mean-square of the set of values.

Parameters **data** (*list*) – The set of values from which to calculate the root-mean-squared.

Returns float – The root-mean-square of the set of values.

gPhoton.MCUtils.**rotvec**(*vector*, *theta*)
Rotate vectors clockwise by theta degrees.

Parameters

- **vector** (*numpy.ndarray*) – The vector to rotate. Must have (2,n) shape.
- **theta** (*numpy.ndarray*) – Angle to rotate the vector, in degrees. Must have (n,) shape.

Returns numpy.ndarray – The rotated vector with (2,n) shape.

PhotonPipe

gPhoton.PhotonPipe.**photonpipe**(*raw6file*, *scstfile*, *band*, *outbase*, *aspfile=None*, *ssdfile=None*, *nullfile=None*, *verbose=0*, *retries=20*)

Apply static and sky calibrations to -raw6 GALEX data, producing fully aspect-corrected and time-tagged photon list files.

Parameters

- **raw6file** (*str*) – Name of the raw6 file to use.
- **scstfile** (*str*) – Spacecraft state file to use.
- **band** (*str*) – Name of the band to use, either 'FUV' or 'NUV'.
- **outbase** (*str*) – Base of the output file names.
- **aspfile** (*int*) – Name of aspect file to use.
- **ssdfile** (*int*) – Name of Stim Separation Data file to use.
- **nullfile** (*int*) – Name of output file to record NULL lines.
- **verbose** (*int*) – Note used. Included for consistency with other tools.
- **retries** (*int*) – Number of query retries to attempt before giving up.

curvetools

gPhoton.curvetools.**aperture_error**(*counts*, *expt*, *bgcounts=0*)

The estimated error in the countrate within the aperture, by adding together the counting error within the aperture and background (if provided) in quadrature.

Parameters

- **counts** (*int*) – Total counts within the aperture.
- **expt** – The exposure time in seconds.
- **bgcounts** (*int*) – The total background counts within the aperture.

Returns float – The error in the counts within the aperture.

gPhoton.curvetools.**caiwarning**(*band*, *bin_ix*, *events*, *verbose=0*)

Test whether a bin contains data from the first 3 legs of an FUV observation as part of the calibration (CAI) survey.

Parameters

- **band** (*str*) – The band being used, either 'FUV' or 'NUV'.
- **bin_ix** (*numpy.ndarray*) – Array indices designating which events are in the time bin of interest.
- **events** (*dict*) – Set of photon events to check if they are near a masked detector region.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.

gPhoton.curvetools.**detedgewarning**(*bin_ix*, *events*, *verbose=0*, *valid_detrad=0.5*)

Assigns warning flags if any of the events of interest are adjacent to the detector edge as defined by a radius of valid_detrad in degrees.

Parameters

- **bin_ix** (*numpy.ndarray*) – Array indices designating which events are in the time bin of interest.
- **events** (*dict*) – Set of photon events to check if they are near the detector edge.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.

- **valid_detrad** (*float*) – The radius, in degrees, beyond which an edge warning is raised.

Returns bool – Returns True/False whether a given set of events are too close to the edge of the detector.

gPhoton.curvetools.**exptimewarning**(*bin_ix*, *events*, *verbose=0*, *ratio=0.5*)

Passes a warning if the effective exposure time within a bin is significantly less than the raw exposure time, which might produce anomalous values due to counting statistics or be a symptom of a problem in the exposure time correction for this bin.

Parameters

- **bin_ix** (*numpy.ndarray*) – Array indices designating which events are in the time bin of interest.
- **events** (*dict*) – Set of photon events to check if there is an effective exposure time warning.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **ratio** (*float*) – The ratio of effective to raw exposure time in a bin below which the bin will be flagged.

Returns bool – Returns True/False whether a bin has a low exposure.

gPhoton.curvetools.**get_curve**(*band*, *ra0*, *dec0*, *radius*, *annulus=None*, *stepsz=None*, *trange=None*, *tranges=None*, *verbose=0*, *coadd=False*, *minexp=1.0*, *maxgap=1.0*, *detsize=1.1*)

Wraps quickmag() to make it ensure some proper parameter formatting and therefore make it slightly more user friendly.

Parameters

- **band** (*str*) – The band being used, either 'FUV' or 'NUV'.
- **ra0** (*float*) – Right ascension, in degrees, of the target position.
- **dec0** (*float*) – Declination, in degrees, of the target position.
- **radius** (*float*) – The radius of the photometric aperture, in degrees.
- **annulus** (*list*) – Radii of the inner and outer extents of the background annulus, in degrees.
- **stepsz** (*float*) – The size (depth) of the time bins to use, in seconds.
- **trange** (*list*) – Minimum and maximum time range to make a light curve, in GALEX time seconds.
- **tranges** (*list*) – Set of time ranges to query within in GALEX time seconds.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **coadd** (*bool*) – Set to True if calculating a total flux instead of flux from each time bin.
- **minexp** (*float*) – Minimum gap size, in seconds, for data to be considered contiguous.
- **maxgap** (*float*) – Maximum gap size, in seconds, for data to be considered contiguous.
- **detsize** (*float*) – Effective diameter, in degrees, of the field-of-view.

Returns dict – The light curve, including input parameters.

gPhoton.curvetools.**getflags**(*band*, *bin_ix*, *events*, *verbose=0*)

Pass flags if data meets conditions that are likely to create misleading photometry. The flags are binary, with bins set as follows: 1 - 'hotspot' - aperture events in pixels contiguous to a masked hotspot 2 - 'mask edge' - aperture events in pixels contiguous to the detector edge 4 - 'exptime' - bin contains < 50% exposure time coverage 8 - 'respose' - events weighted with response < 0.7 16 - 'nonlinearity' - local count-rate exceeds 10% response dropoff 32 - 'detector edge' - events outside of 0.5 degrees of detector center 64 - 'bg hotspot' - annulus events in pixels contiguous to a masked hotspot 128 - 'bg mask' - annulus events in pixels contiguous to detector edge

Parameters

- **band** (*str*) – The band being used, either 'FUV' or 'NUV'.
- **bin_ix** (*numpy.ndarray*) – Array indices designating which events are in the time bin of interest.
- **events** (*dict*) – Set of photon events to check for warning flags.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.

Returns numpy.ndarray – The array of flags for each photon event.

gPhoton.curvetools.**gphot_params**(*band*, *skypos*, *radius*, *annulus=None*, *verbose=0*, *detsize=1.25*, *stepsz=None*, *trange=None*)

Populate a dict() with parameters that are constant over all bins.

Parameters

- **band** (*str*) – The band being used, either 'FUV' or 'NUV'.
- **skypos** (*list*) – A two-element list containing the RA and DEC.
- **radius** (*float*) – Radius of the photometric aperture in degrees.
- **annulus** (*list*) – A two-element list containing the inner and outer radius to use for background subtraction during aperture photometry, in degrees.
- **verbose** (*int*) – Level of verbosity, 0 = minimum verbosity.
- **detsize** (*float*) – Effective diameter, in degrees, of the field-of-view.
- **stepsz** (*float*) – Size of time bin to use, in seconds.
- **trange** (*list*) – The start and end timestamps to consider, in GALEX time

Returns dict – The set of parameters that are constant across all bins.

gPhoton.curvetools.**hashresponse**(*band*, *events*, *verbose=0*)

Given detector xi, eta, return the response at each position.

Parameters

- **band** (*str*) – The band being used, either 'FUV' or 'NUV'.
- **events** (*dict*) – The set of photon events.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.

Returns dict – The photon event properties, updated with the response at each position.

gPhoton.curvetools.**lowresponsewarning**(*bin_ix*, *events*, *verbose=0*, *ratio=0.7*)

Checks for anomalously low response values in the data of interest, which could indicate data on poorly characterized or behaved detector regions.

Parameters

- **bin_ix** (*numpy.ndarray*) – Array indices designating which events are in the time bin of interest.
- **events** (*dict*) – Set of photon events to check if there is a low response.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **ratio** (*float*) – The value that defines a low response, between 0 and 1. (Where a response value of 1 indicates a "perfect" response value w/ no correction.)

Returns bool – Returns True/False whether a given set of events contain any on a low response region of the detector.

gPhoton.curvetools.**maskwarning**(*band*, *bin_ix*, *events*, *verbose=0*, *mapkey='H'*, *mode=None*)

Test if any given events are near a masked detector region.

Parameters

- **band** (*str*) – The band being used, either 'FUV' or 'NUV'.
- **bin_ix** (*numpy.ndarray*) – Array indices designating which events are in the time bin of interest.
- **events** (*dict*) – Set of photon events to check if they are near a masked detector region.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **mapkey** (*str*) – Text code indicating whether to use the hotspot mask ("H") or the flat (edge) mask ("E").

Returns bool – Returns True/False whether a given set of events are near a masked detector region.

gPhoton.curvetools.**nonlinearitywarning**(*band*, *bin_ix*, *events*, *verbose=0*)

Flag count rates above the 10% local nonlinearty dropoff, per the calibration paper.

Parameters

- **band** (*str*) – The band that is being used, either 'FUV' or 'NUV'.
- **bin_ix** (*numpy.ndarray*) – Array indices designating which events are in the time bin of interest.
- **events** (*dict*) – Set of photon events to check if they are in the non-linearity regime.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.

Returns bool – Returns True/False whether a given set of events are at the non-linearity regime.

gPhoton.curvetools.**pullphotons**(*band*, *ra0*, *dec0*, *tranges*, *radius*, *verbose=0*, *flag=0*, *photonfile=None*, *detsize=1.25*)

Reads photon list data from the MAST database using a cone search.

Parameters

- **band** (*str*) – Name of the band being used, either 'FUV' or 'NUV'.
- **ra0** (*float*) – Right ascension of the targeted sky position, in degrees.
- **dec0** (*float*) – Declination of the targeted sky position, in degrees.
- **tranges** (*list*) – Set of time ranges from which to retrieve photon events, in GALEX time units
- **radius** (*float*) – The radius, in degrees, defining a cone on the sky that is centered on ra0 and dec0, from which to extract photons.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.

- **photonfile** (*str*) – Name of photon event file to use (if reading from disk).
- **flag** (*int*) – Photon list flag value upon which to select. Default of 0 corresponds to nominally corrected data with no issues. NOTE: 'Flag' is not a reliable way to parse data at this time. You should compare timestamps against the aspect file.

Returns dict – The set of photon events and their properties.

gPhoton.curvetools.**query_photons**(*band*, *ra0*, *dec0*, *tranges*, *radius*, *verbose=0*, *flag=0*, *detsize=1.25*)

Retrieve photons within an aperture from the database.

Parameters

- **band** (*str*) – Name of the band being used, either 'FUV' or 'NUV'.
- **ra0** (*float*) – Right ascension of the targeted sky position, in degrees.
- **dec0** (*float*) – Declination of the targeted sky position, in degrees.
- **tranges** (*list*) – Set of time ranges from which to retrieve photon events, in GALEX time units
- **radius** (*float*) – The radius, in degrees, defining a cone on the sky that is centered on ra0 and dec0, from which to extract photons.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **flag** (*int*) – Photon list flag value upon which to select. Default of 0 corresponds to nominally corrected data with no issues. NOTE: 'Flag' is not a reliable way to parse data at this time. You should compare timestamps against the aspect file.

Returns dict – The set of photon events with their properties.

gPhoton.curvetools.**quickmag**(*band*, *ra0*, *dec0*, *tranges*, *radius*, *annulus=None*, *stepsz=None*, *verbose=0*, *detsize=1.25*, *coadd=False*)

Primary wrapper function for generating and synthesizing all of the parameters and calculations necessary to create light curves.

Parameters

- **band** (*str*) – The band being used, either 'FUV' or 'NUV'.
- **ra0** (*float*) – Right ascension, in degrees, of the target position.
- **dec0** (*float*) – Declination, in degrees, of the target position.
- **tranges** (*list*) – Set of time ranges to query within in GALEX time seconds.
- **radius** (*float*) – The radius of the photometric aperture, in degrees.
- **annulus** (*list*) – Radii of the inner and outer extents of the background annulus, in degrees.
- **stepsz** (*float*) – The size of the time bins to use, in seconds.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **detsize** – Effective diameter, in degrees, of the field-of-view.
- **coadd** (*bool*) – Set to True if calculating a total flux instead of flux from each time bin.

Returns dict – The light curve, including input parameters.

gPhoton.curvetools.**read_photons**(*photonfile, ra0, dec0, tranges, radius, verbose=0, colnames=['t', 'x', 'y', 'xa', 'ya', 'q', 'xi', 'eta', 'ra', 'dec', 'flags']*)

Read a photon list file and return a python dict() with the expected format.

Parameters

- **photonfile** (*str*) – Name of the photon event file to use.
- **ra0** (*float*) – Right ascension of the targeted sky position, in degrees.
- **dec0** (*float*) – Declination of the targeted sky position, in degrees.
- **tranges** (*list*) – Set of time ranges from which to retrieve photon events, in GALEX time units
- **radius** (*float*) – The radius, in degrees, defining a cone on the sky that is centered on ra0 and dec0, from which to extract photons.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **colnames** (*list*) – Labels of the columns found in the photon event file.

Returns dict – The set of photon events and their properties.

gPhoton.curvetools.**recoverywarning**(*band*, *bin_ix*, *events*, *verbose=0*)

Test whether the bin contains data that was collected during a spacecraft recovery period (e.g. FUV cycling) as defined by the lookup table in galextools.recovery_tranges().

Parameters

- **band** (*str*) – The band being used, either 'FUV' or 'NUV'.
- **bin_ix** (*numpy.ndarray*) – Array indices designating which events are in the time bin of interest.
- **events** (*dict*) – Set of photon events to check if they are near a masked detector region.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.

gPhoton.curvetools.**reduce_lcurve**(*bin_ix*, *region_ix*, *data*, *function*, *dtype='float64'*)

Produces light curve columns by iteratively applying 'function' to 'data' within 'region_ix' over 'bin_ix'.

Parameters

- **bin_ix** (*numpy.ndarray*) – Array indices designating which events are in the time bin of interest.
- **region_ix** (*numpy.ndarray*) – Array indices designating which events are in the spatial region of interest (e.g. the photometric aperture).
- **data** (*numpy.ndarray*) – The data to apply the function on.
- **function** (*function*) – The function to apply to the data.
- **dtype** (*str*) – The data type.

Returns numpy.ndarray – The light curve columns.

gPhoton.curvetools.**write_curve**(*band*, *ra0*, *dec0*, *radius*, *csvfile=None*, *annulus=None*, *stepsz=None*, *trange=None*, *tranges=None*, *verbose=0*, *coadd=False*, *iocode='w'*, *detsize=1.1*, *overwrite=False*, *minexp=1.0*, *maxgap=1.0*, *minimal_output=False*, *photoncsvfile=None*)

Generates a lightcurve and optionally writes the data to a CSV file.

Parameters

- **band** (*str*) – The band being used, either 'FUV' or 'NUV'.
- **ra0** (*float*) – Right ascension, in degrees, of the target position.
- **dec0** (*float*) – Declination, in degrees, of the target position.
- **radius** (*float*) – The radius of the photometric aperture, in degrees.
- **csvfile** (*str*) – Name of the photon event CSV file to use for the lightcurve.
- **annulus** (*list*) – Radii of the inner and outer extents of the background annulus, in degrees.
- **stepsz** (*float*) – The size of the time bins to use, in seconds.
- **trange** (*list*) – Minimum and maximum timew within which to make a lightcurve, in GALEX time seconds.
- **tranges** (*list*) – Set of time ranges to query within in GALEX time seconds.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **coadd** (*bool*) – Set to True if calculating a total flux instead of flux from each time bin.
- **iocode** (*str*) – The code to use when writing the output file.
- **detsize** (*float*) – Effective diameter, in degrees, of the field-of-view.
- **overwrite** (*bool*) – If True, overwite an existing output file.
- **minexp** (*float*) – Minimum gap size, in seconds, for data to be considered contiguous.
- **maxgap** (*float*) – Maximum gap size, in seconds, for data to be considered contiguous.
- **minimal_output** (*bool*) – If True, produce an output file with a minimum number of columns.
- **photoncsvfile** (*str*) – Name of the photon event CSV file to write the photon list data to.

Returns dict – The light curve, including input parameters.

gPhoton.curvetools.**xieta2colrow**(*xi*, *eta*, *band*, *detsize=1.25*)

Convert detector xi and eta into col and row.

Parameters

- **xi** (*numpy.ndarray*) – Sky-projected event "x" positions *in detetor coordinates*.
- **eta** (*numpy.ndarray*) – Sky-projected event "y" positions *in detetor coordinates*.
- **band** (*str*) – The band that is being used, either 'FUV' or 'NUV'.
- **detsize** (*float*) – Effective diameter, in degrees, of the field-of-view.

Returns tuple – A two-element tuple containing the column and row values.

dbasetools

gPhoton.dbasetools.**avg_sources**(*band*, *skypos*, *radius=0.001*, *maglimit=20.0*, *verbose=0*, *catalog='MCAT'*)

Return the mean position of sources within the search radius.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.

- **skypos** (*list*) – The right ascension and declination, in degrees.
- **radius** (*float*) – The radius within which to search for MCAT sources, in degrees?
- **maglimit** (*float*) – The NUV faint limit to return unique sources for.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **catalog** (*str*) – The name of the catalog to query.

Returns tuple – A three-element tuple containing the mean RA, mean DEC, and mean FWHM of sources within the search radius.

gPhoton.dbasetools.**compute_exptime**(*band*, *tr*, *verbose=0*, *skypos=None*, *detsize=1.25*, *coadd=False*)

Compute the total effective exposure time, in seconds, accounting for shutter and deadtime _and_ detector size (i.e. effective coverage).

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **tr** (*list*) – Pairs of minimum and maximum times (in GALEX time) to consider.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **skypos** (*list*) – The right ascension and declination of interest, in degrees.
- **detsize** (*float*) – The effective detector diameter, in degrees.
- **coadd** (*bool*) – Should the effective exposure time be calculated across all time ranges, e.g., a coadded effective exposure time.

Returns list

gPhoton.dbasetools.**compute_shutter**(*band*, *trange*, *verbose=0*, *shutgap=0.05*, *timestamplist=False*)

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **trange** (*list*) – Minimum and maximum time (in GALEX time) to consider.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **shutgap** (*float*) – Amount of time, in seconds, that defines the minimum gap in observation that corresponds to a 'shutter' (not a true exposure time).
- **timestamplist** (*list*) – Global event time stamps.

Returns numpy.ndarray – The total shutter time, in seconds, during the specified time range.

gPhoton.dbasetools.**distinct_tranges**(*times*, *maxgap=1.0*)

Produces a list of pairs of start / stop times delimiting distinct unique time ranges, given that gaps of >maxgap initiate a new time period.

Parameters

- **times** (*list*) – A set of time stamps to extract unique time ranges from.
- **maxgap** (*float*) – Maximum gap size, in seconds, for data to be considered contiguous.

Returns list – The set of distinct time ranges.

gPhoton.dbasetools.**empirical_deadtime**(*band*, *trange*, *verbose=0*, *feeclkratio=0.966*, *timestamplist=False*)

Calculate empirical deadtime (per global count rate) using revised formulas. Restricts integration of global counts to non-shuttered time periods.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **trange** (*list*) – Minimum and maximum time (in GALEX time) to consider.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **feeclkratio** (*float*) – A scaling parameter for the Front End Electronics clock.
- **timestamplist** (*list*) – Global even time stamps.

Returns float – The empirical deadtime ratio.

gPhoton.dbasetools.**exp_from_objid**(*objid*)

Return the effective exposure time, start time, and end time from the MCAT for a given GALEX object ID.

Parameters **objid** (*int*) – GALEX object ID.

Returns dict – The FUV and NUV effective exposure time and start/stop times.

gPhoton.dbasetools.**exposure**(*band*, *trange*, *verbose=0*)

Calculate the effective exposure time in a period, in seconds, accounting for shutter and deadtime. Does not account for actual sky coverage of the telescope during the time period queried (see: compute_exptime() below).

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **trange** (*list*) – Minimum and maximum time (in GALEX time) to consider.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.

Returns float – The effective exposure time, in seconds.

gPhoton.dbasetools.**fGetTimeRanges**(*band*, *skypos*, *trange=None*, *detsize=1.1*, *verbose=0*, *maxgap=1.0*, *minexp=1.0*, *skyrange=None*, *maxgap_override=False*)

Find the contiguous time ranges within a time range at a specific location.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **skypos** (*list*) – The right ascension and declination, in degrees.
- **trange** (*list*) – Minimum and maximum time (in GALEX time) to consider.
- **detsize** (*float*) – The effective detector diameter, in degrees.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **maxgap** (*float*) – Maximum gap size, in seconds, for data to be considered contiguous.
- **minexp** (*float*) – Minimum gap size, in seconds, for data to be considered contiguous.
- **skyrange** (*list*) – Values in degrees RA and Dec of a box around skypos that defines the extent of the region of interest.

- **maxgap_override** (*bool*) – Enables an experimental feature where maxgap can be less than one second.

Returns numpy.ndarray – A valid set of time ranges, accounting for minimum exposure lengths and maximum gaps.

gPhoton.dbasetools.**find_nearest_mcat**(*band*, *skypos*, *radius*, *maglimit=30.0*)

Given a sky position and a search radius, find the nearest MCAT source and return its position and magnitude in specified band.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **skypos** (*array*) – Two element array of RA and Dec in decimal degrees.
- **radius** (*float*) – Search radius in decimal degrees.
- **maglimit** (*float*) – The NUV faint limit to return MCAT sources for.

gPhoton.dbasetools.**find_unique_sources**(*band*, *ra0*, *dec0*, *searchradius*, *maglimit=20.0*, *margin=0.001*, *verbose=0*)

Locates nominally unique (via crossmatch) GALEX sources in the MCAT near a sky position of interest.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **ra0** (*float*) – The right ascension, in degrees, around which to search.
- **dec0** (*float*) – The declination, in degrees, around which to search.
- **searchradius** (*float*) – The size of the radius to search for unique sources, in degrees.
- **maglimit** (*float*) – The NUV faint limit to return unique sources for.
- **margin** (*float*) – Buffer size when determining matches, in degrees.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.

Returns numpy.ndarray – The set of unique sources.

gPhoton.dbasetools.**get_aspect**(*band, skypos, trange=[600000000.0, 1100000000.0], verbose=0, detsize=1.25*)

Get aspect solution in a dict() for the given time range.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **skypos** (*list*) – The right ascension and declination, in degrees.
- **trange** (*list*) – Minimum and maximum time (in GALEX time) to consider.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **detsize** (*float*) – The effective detector diameter, in degrees.

Returns dict – The aspect solution parameters.

gPhoton.dbasetools.**get_mags**(*band*, *ra0*, *dec0*, *radius*, *maglimit*, *mode='coadd'*, *zpmag={'NUV': 20.08*, *'FUV': 18.82}*, *verbose=0*)

Given RA, Dec and search radius, searches the coadd MCAT for sources. Returns a dict() which contains magnitudes for all of the APER settings. Note: Visit mode returns a lot more sources, more slowly than coadd mode given the same search parameters. You should probably use smaller search radii in visit

mode. If you're just trying to find unique sources in a large region, use coadd mode and then pass the result through the parse_unique_sources() function contained in this module.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **ra0** (*float*) – The right ascension, in degrees, around which to search.
- **dec0** (*float*) – The declination, in degrees, around which to search.
- **radius** (*float*) – The size of the search radius for MCAT sources in degrees.
- **maglimit** (*float*) – The NUV faint limit to return MCAT sources for.
- **mode** (*str*) – Specify whether to return MCAT sources from the 'visit' or 'coadd' catalog.
- **zpmag** (*dict*) – The zero-point magnitude in FUV and NUV.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.

Returns dict – The set of magnitudes from different apertures for sources in the MCAT, centered around the specified coordinate.

gPhoton.dbasetools.**get_mcat_data**(*skypos*, *rad*)

Return visit-level MCAT sources and their catalog values within a give radius of the specified sky position.

Parameters

- **skypos** (*list*) – The right ascension and declination, in degrees, around which to search for MCAT sources.
- **rad** (*float*) – The radius within which to search for MCAT sources, in degrees.

Returns dict – The MCAT sources within the specified radius.

gPhoton.dbasetools.**get_valid_times**(*band*, *skypos*, *trange=None*, *detsize=1.1*, *verbose=0*, *skyrange=None*)

Given a sky position and (optional) extent, return all of the times periods containing spatially intersecting observations.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **skypos** (*list*) – The right ascension and declination, in degrees.
- **trange** (*list*) – Minimum and maximum time (in GALEX time) to consider.
- **detsize** (*float*) – The effective detector diameter, in degrees.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **skyrange** (*list*) – Values in degrees RA and Dec of a box around skypos that defines the extent of the region of interest.

Returns numpy.ndarray – A sorted set of unique time stamps.

gPhoton.dbasetools.**globalcount_shuttered**(*band*, *trange*, *verbose=0*, *timestamplist=False*)

Global event counts over the time range, exluding shuttered periods (due to no non-NULL data).

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.

- **trange** (*list*) – Minimum and maximum time (in GALEX time) to consider.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **timestamplist** (*list*) – Global event time stamps.

Returns int – Total global counts excluding shuttered periods.

gPhoton.dbasetools.**legnum**(*t*, *obsdata=None*)
Returns the lef number.

gPhoton.dbasetools.**mcat_skybg**(*band*, *skypos*, *radius*, *verbose=0*, *trange=None*, *mcat=None*, *searchradius=0.1*)
Estimate the sky background using the MCAT 'skybg' for nearby sources.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **skypos** (*list*) – The right ascension and declination, in degrees.
- **radius** (*float*) – The radius in which to search for MCAT sources in degrees.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **trange** (*list*) – Minimum and maximum time (in GALEX time) to consider.

Returns float – The estimated sky background in the photometric aperture, in counts per second.

gPhoton.dbasetools.**nearest_distinct_source**(*band*, *skypos*, *radius=0.1*, *maglimit=20.0*, *verbose=0*, *catalog='MCAT'*)
Return parameters for the nearest non-targeted source.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **skypos** (*list*) – The right ascension and declination, in degrees.
- **radius** (*float*) – The radius within which to search for the nearest MCAT source, in degrees.
- **maglimit** (*float*) – The NUV faint limit to return unique sources for.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **catalog** (*str*) – The name of the catalog to query.

Returns numpy.ndarray – Catalog values for the nearest non-targeted source.

gPhoton.dbasetools.**nearest_source**(*band*, *skypos*, *radius=0.01*, *maglimit=20.0*, *verbose=0*, *catalog='MCAT'*)
Return targeting parameters for the nearest MCAT source to a position.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **skypos** (*list*) – The right ascension and declination, in degrees.
- **radius** (*float*) – The radius within which to search for the nearest MCAT source, in degrees.
- **maglimit** (*float*) – The NUV faint limit to return unique sources for.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **catalog** (*str*) – The name of the catalog to query.

Returns tuple – A three-element tuple containing the mean RA, mean DEC, and mean FWHM of the nearest sources within the search radius.

gPhoton.dbasetools.**obstype**(*t*, *obsdata=None*)
Determines the type of dither pattern.

gPhoton.dbasetools.**obstype_from_objid**(*objid*)
Return the number of legs and petal value for a given GALEX object ID.

Parameters **objid** (*int*) – GALEX object ID.

Returns tuple – A two-element tuple containing the number of legs and the petal value, which can be used to infer the observation type/strategy.

gPhoton.dbasetools.**optimize_annulus**(*optrad*, *outann*, *verbose=0*)
Suggest optiumum annulus dimensions.

Parameters

- **optrad** (*float*) – The optimal photometric aperture radius, in degrees.
- **outann** (*float*) – The outer annulus to test, in degrees.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.

Returns float - The suggested outer annulus for background calculation.

gPhoton.dbasetools.**parse_unique_sources**(*ras*, *decs*, *margin=0.001*)

Iteratively returns unique sources based upon a margin within which two sources should be considered the same sources. Is a little bit sensitive to the first entry and could probably be written to be more robust, but works well enough.

Parameters

- **ras** (*numpy.ndarray*) – Set of right ascensions, in degrees.
- **decs** (*numpy.ndarray*) – Set of declinations, in degrees.
- **margin** (*float*) – Buffer size when determining matches, in degrees.

Returns list – The indices corresponding to unique sources.

gPhoton.dbasetools.**stimcount_shuttered**(*band*, *trange*, *verbose=0*, *timestamplist=False*)

Returns the stim count over a time range, excluding periods that the detector is considered shuttered (because of no non-NULL data).

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **trange** (*list*) – Minimum and maximum time (in GALEX time) to consider.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **timestamplist** (*list*) – Global detector event timestamps.

Returns int – Total stim counts excluding shuttered time ranges.

gPhoton.dbasetools.**suggest_bg_radius**(*band*, *skypos*, *radius=0.1*, *maglimit=20.0*, *verbose=0*, *catalog='MCAT'*)

Returns a recommended background radius based upon the positions and FWHM of nearby sources in the MCAT.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **skypos** (*list*) – The right ascension and declination, in degrees.
- **radius** (*float*) – The radius within which to search for MCAT sources, in degrees?
- **maglimit** (*float*) – The NUV faint limit to return unique sources for.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **catalog** (*str*) – The name of the catalog to query.

Returns float – The suggested radius to define the background with.

gPhoton.dbasetools.**suggest_parameters**(*band*, *skypos*, *verbose=0*)

Provide suggested coordinates and photometric apertures for a source given the location of known MCAT sources nearby.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **skypos** (*list*) – The right ascension and declination, in degrees.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.

Returns tuple – A five-element tuple containing the suggested right ascension, declination, photometric aperture, inner annulus, and outer annulus, all in degrees.

gCalrun

gPhoton.gCalrun.**calrun**(*outfile, band, nsamples=10, seed=323, rarange=[0.0, 360.0], decrange=[-90.0, 90.0], exprange=[0.0, 5000.0], maglimit=24.0, verbose=0, radius=0.0016666666666666668, annulus=[0.0083, 0.025]*)

Generate a bunch of magnitudes with comparisons against MCAT values for random points on the sky within given legal ranges. Write it to a CSV.

Parameters

- **outfile** (*str*) – Name of the output file.
- **band** (*str*) – The band being used, either 'FUV' or 'NUV'.
- **nsamples** (*int*) – Number of random positions to sample.
- **seed** (*int*) – The seed to use when generating the random sample.
- **rarange** (*list*) – The minimum and maximum RA range to sample from.
- **decrange** (*list*) – The minimum and maximum DEC range to sample from.
- **exprange** (*list*) – The minimum and maximum exposure time to sample, in seconds.
- **maglimit** (*float*) – The faintest source to consider.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **radius** (*float*) – Photometric aperture radius, in degrees.
- **annulus** (*list*) – Inner and outer extent of background annulus, in degrees.

gPhoton.gCalrun.**check_annulus**(*args*)

Checks and formats the annulus values.

Parameters **args** (*argparse.ArgumentParser Namespace*) – The command-line arguments.

Returns argparse.ArgumentParser Namespace – The updated command-line arguments.

gPhoton.gCalrun.**check_args**(*args*)

Ensures that the input RA is between 0. and 360., and the DEC is between -90 and 90.

Parameters **args** (*argparse.ArgumentParser Namespace*) – The command-line arguments.

Returns argparse.ArgumentParser Namespace – The command-line arguments. Included for function usage consistency.

gPhoton.gCalrun.**find_random_positions**(*rarange=[0.0, 360.0], decrange=[-90.0, 90.0], nsamples=10, seed=323*)

Generate 'nsamples' random positions uniformly selected from the specified RA and DEC ranges.

Parameters

- **rarange** (*list*) – The minimum and maximum RA range to sample from.
- **decrange** (*list*) – The minimum and maximum DEC range to sample from.
- **nsamples** (*int*) – The number of random positions to return.
- **seed** (*int*) – The seed to use when generating the random sample.

Returns tuple – A two-element tuple containing 'nsamples' of RA and DEC values.

gPhoton.gCalrun.**setup_parser**()
Setup input arguments.

gQuery

gPhoton.gQuery.**allphotons**(*band*, *ra0*, *dec0*, *t0*, *t1*, *radius*, *flag=0*)
Grab the major columns for all events within an aperture.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **ra0** (*float*) – The right ascension, in degrees, around which to search.
- **dec0** (*float*) – The declination, in degrees, around which to search.
- **t0** (*long*) – The minimum time stamp to search.
- **t1** (*long*) – The maximum time stamp to search.
- **radius** (*float*) – The radius within which to search, in degrees.
- **flag** (*int*) – Only return times with this flag value. Zero is nominal. NOTE: 'Flag' is not a reliable way to parse data at this time. You should compare event timestamps against the aspect file.

Returns str – The query to submit to the database.

gPhoton.gQuery.**alltimes**(*band*, *t0*, *t1*)
Return the time stamps of every detector event within a time range.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **t0** (*long*) – The minimum time stamp to search.
- **t1** (*long*) – The maximum time stamp to search.

Returns str – The query to submit to the database.

gPhoton.gQuery.**aperture**(*band*, *ra0*, *dec0*, *t0*, *t1*, *radius*)
Integrate counts over an aperture at a position.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **ra0** (*float*) – The right ascension, in degrees, around which to search.
- **dec0** (*float*) – The declination, in degrees, around which to search.
- **t0** (*long*) – The minimum time stamp to search.
- **t1** (*long*) – The maximum time stamp to search.
- **radius** (*float*) – The radius within which to integrate counts, in degrees.

Returns str – The query to submit to the database.

gPhoton.gQuery.**aspect**(*t0*, *t1*)
Return the aspect information based on a time range.

Parameters

- **t0** (*long*) – The minimum time stamp to search.
- **t1** (*long*) – The maximum time stamp to search.

Returns str – The query to submit to the database.

gPhoton.gQuery.**aspect_ecl**(*eclipse*)
Return the aspect information based upon an eclipse number.

Parameters **eclipse** (*int*) – The eclipse to return aspect information for.

Returns str – The query to submit to the database.

gPhoton.gQuery.**aspect_skypos**(*ra*, *dec*, *detsize=1.25*)
Return the aspect information based upon sky position and det radius.

Parameters

- **ra** (*float*) – The right ascension to search on, in degrees.
- **dec** (*float*) – The declination to search on, in degrees.
- **detsize** (*float*) – Effective diameter, in degrees, of the field-of-view.

Returns str – The query to submit to the database.

gPhoton.gQuery.**box**(*band*, *ra0*, *dec0*, *t0*, *t1*, *radius*, *flag=0*)

Return data within a box centered on ra0, dec0 with sides of length 2*radius.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **ra0** (*float*) – The right ascension, in degrees, around which to search.
- **dec0** (*float*) – The declination, in degrees, around which to search.

- **t0** (*long*) – The minimum time stamp to search.
- **t1** (*long*) – The maximum time stamp to search.
- **radius** (*float*) – The radius within which to search, in degrees.
- **flag** (*int*) – If true, only return flag=0 data. Else return all non-NULL.

Returns str – The query to submit to the database.

gPhoton.gQuery.**boxcentroid**(*band*, *t0*, *t1*, *xr*, *yr*)

Find the mean position of events inside of a box in detector space.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **t0** (*long*) – The minimum time stamp to search.
- **t1** (*long*) – The maximum time stamp to search.
- **xr** (*list*) – The minimum and maximum x-values that define the box.
- **yr** (*list*) – The minimum and maximum y-values that define the box.

Returns str – The query to submit to the database.

gPhoton.gQuery.**boxcount**(*band*, *t0*, *t1*, *xr*, *yr*)

Find the number of events inside of a box defined by [xy] range in detector space coordinates. This is useful for pulling out stim events.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **t0** (*long*) – The minimum time stamp to search.
- **t1** (*long*) – The maximum time stamp to search.
- **xr** (*list*) – The minimum and maximum x-values that define the box.
- **yr** (*list*) – The minimum and maximum y-values that define the box.

Returns str – The query to submit to the database.

gPhoton.gQuery.**boxtimes**(*band*, *t0*, *t1*, *xr*, *yr*)

Get the list of times for events inside of a box in detector space.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **t0** (*long*) – The minimum time stamp to search.
- **t1** (*long*) – The maximum time stamp to search.
- **xr** (*list*) – The minimum and maximum x-values that define the box.
- **yr** (*list*) – The minimum and maximum y-values that define the box.

Returns str – The query to submit to the database.

gPhoton.gQuery.**deadtime**(*band*, *t0*, *t1*, *feeclkratio=0.966*, *tec2fdead=5.52e-06*)

Return the emperically determined deadtime correction based upon the global count rate.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **t0** (*long*) – The minimum time stamp to search.
- **t1** (*long*) – The maximum time stamp to search.
- **feeclkratio** (*float*) – Ratio of Front End Electronics clock rates.
- **tec2fdead** (*float*) – The nominal amount of time following an event that the detector is unable to detect another event.

Returns str – The query to submit to the database.

gPhoton.gQuery.**deadtime1**(*band*, *t0*, *t1*, *flag=False*)
Return the global counts of non-NULL data.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **t0** (*long*) – The minimum time stamp to search.
- **t1** (*long*) – The maximum time stamp to search.
- **flag** (*bool*) – If true, return only flag=0 data, else return all non-NULL.

Returns str – The query to submit to the database.

gPhoton.gQuery.**deadtime2**(*band*, *t0*, *t1*)
Return the global counts for NULL data.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **t0** (*long*) – The minimum time stamp to search.
- **t1** (*long*) – The maximum time stamp to search.

Returns str – The query to submit to the database.

gPhoton.gQuery.**detbox**(*band*, *t0*, *t1*, *xr*, *yr*)

Return all events inside a box defined in detector space by [xy] range. Created as a sanity check for stim events.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **t0** (*long*) – The minimum time stamp to search.
- **t1** (*long*) – The maximum time stamp to search.
- **xr** (*list*) – The minimum and maximum x-values that define the box.
- **yr** (*list*) – The minimum and maximum y-values that define the box.

Returns str – The query to submit to the database.

gPhoton.gQuery.**exposure_range**(*band*, *ra0*, *dec0*, *t0=1*, *t1=10000000000000*)
Find time ranges for which data exists at a given position.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **ra0** (*float*) – The right ascension, in degrees, around which to search.

- **dec0** (*float*) – The declination, in degrees, around which to search.
- **t0** (*long*) – The minimum time stamp to search for exposure ranges.
- **t1** (*long*) – The maximum time stamp to search for exposure ranges.

Returns str – The query to submit to the database.

gPhoton.gQuery.**exposure_ranges**(*band*, *ra0*, *dec0*, *t0=1*, *t1=10000000000000*, *detsize=1.25*, *epsilon=0.001*)

Returns a list of times (in one second increments) where data exists with an aspect solution within detsize of [ra0,dec0].

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **ra0** (*float*) – The right ascension, in degrees, around which to search.
- **dec0** (*float*) – The declination, in degrees, around which to search.
- **t0** (*long*) – The minimum time stamp to search for exposure ranges.
- **t1** (*long*) – The maximum time stamp to search for exposure ranges.
- **detsize** (*float*) – Effective diameter, in degrees, of the field-of-view.
- **epsilon** (*float*) – Buffer on t1 to avoid missing the end value in the search.

Returns str – The query to submit to the database.

gPhoton.gQuery.**getArray**(*query*, *verbose=0*, *retries=100*)

Manage a database call which returns an array of values.

Parameters

- **query** (*str*) – The query to run.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **retries** (*int*) – Number of query retries to attempt before giving up.

Returns requests.Response or None – The response from the server. If the query does not receive a response, returns None.

gPhoton.gQuery.**getValue**(*query*, *verbose=0*, *retries=100*)

Manage a database call which returns a single value.

Parameters

- **query** (*str*) – The query to run.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **retries** (*int*) – Number of query retries to attempt before giving up.

Returns requests.Response or None – The response from the server. If the query does not receive a response, returns None.

gPhoton.gQuery.**globalcounts**(*band*, *t0*, *t1*, *flag=False*)

Return the total number of detector events within a time range.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **t0** (*long*) – The minimum time stamp to search.

- **t1** (*long*) – The maximum time stamp to search.
- **flag** (*bool*) – If true, return only flag=0 data. Else return all non-NULL.

Returns str – The query to submit to the database.

gPhoton.gQuery.**hasNaN**(*query*)

Check if there is NaN in a query (or any string) and, if so, raise an exception because that probably indicates that something has gone wrong.

Parameters **query** (*str*) – The query string to check.

gPhoton.gQuery.**mcat_objid_search**(*objid*)

Return a bunch of observation data for a visit level objid (ggoid). Doing the same for coadd level data is not yet supported.

Parameters **objid** (*long*) – The MCAT Object ID to return the observation data from.

Returns str – The query to submit to the database.

gPhoton.gQuery.**mcat_sources**(*band*, *ra0*, *dec0*, *radius*, *maglimit=20.0*)

Return the MCAT coadd sources given sky position and search radius (and optional lower magnitude limit). Columns are: [0,RA],[1,Dec],[2,NUV_mag],[3,FUV_mag],[4,FoV_radius],[5,NUV_skybg], [6,FUV_skybg],[7,NUV_FWHM_world],[8,FUV_FWHM_world], [9:15,FUV_mag_aper_1:7],[16:22,NUV_mag_aper_1:7 [23:29,FUV_magerr_aper_1:7],[30:36,NUV_magerr_aper1:7]

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **ra0** (*float*) – The right ascension, in degrees, around which to search.
- **dec0** (*float*) – The declination, in degrees, around which to search.
- **radius** (*float*) – The radius within which to search for MCAT sources, in degrees.
- **maglimit** (*float*) – The NUV faint limit to return MCAT sources for.

Returns str – The query to submit to the database.

gPhoton.gQuery.**mcat_visit_sources**(*ra0*, *dec0*, *radius*)

Return the MCAT per-visit sources given sky position and search radius. The columns are as follows: [0,objid],[1,ra],[2,dec],[3,NUV_mag],[4,FUV_mag],[5,FoV_radius], [6,NUV_skybg],[7,FUV_skybg],[8,NUV_FWHM],[9,FUV_FWHM],[10,FUV_expt], [11,NUV_expt],[12:18,FUV_mag_aper_1:7],[19:25,NUV_mag_aper_1:7], [26:32,FUV_magerr_aper_1:7],[33:39,NUV_magerr_aper_1:7],[40,Nobssecs], [41,Fobssecs],[42,NUV_artifact],[43,FUV_artifact],[44,FUV_obstart], [45,FUV_obsend],[46,NUV_obstart],[47,NUV_obsend], [48,FUV_ALPHA_J2000],[49,FUV_DELTA_J2000], [50,NUV_ALPHA_J2000],[51,NUV_DELTA_J2000]

Parameters

- **ra0** (*float*) – The right ascension, in degrees, around which to search.
- **dec0** (*float*) – The declination, in degrees, around which to search.
- **radius** (*float*) – The radius within which to search for MCAT sources, in degrees.

Returns str – The query to submit to the database.

gPhoton.gQuery.**obstype**(*objid*)
Get the dither pattern type based on the object id.

Parameters **objid** (*long*) – The MCAT Object ID to return the observation type data from.

Returns str – The query to submit to the database.

gPhoton.gQuery.**obstype_from_t**(*t*)
Get the dither pattern type based on the time stamp.

gPhoton.gQuery.**shutter**(*band*, *t0*, *t1*)
Get shutter correction, i.e., number of 0.05-sec gaps in data.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **t0** (*long*) – The minimum time stamp to search.
- **t1** (*long*) – The maximum time stamp to search.

Returns str – The query to submit to the database.

gPhoton.gQuery.**skyrect**(*band*, *ra0*, *dec0*, *t0*, *t1*, *ra*, *dec*, *flag=0*)
Extract photon data falling within a specified rectangle on the sky.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **ra0** (*float*) – The right ascension, in degrees, around which to search.
- **dec0** (*float*) – The declination, in degrees, around which to search.
- **t0** (*long*) – The minimum time stamp to search.
- **t1** (*long*) – The maximum time stamp to search.
- **ra** (*float*) – The length in degrees along RA describing the region of interest.
- **dec** (*float*) – The length in degrees along Dec describing the region of interest.
- **flag** (*int*) – Flag value of non-NULL data to return. Zero is nominal.

Returns str – The query to submit to the database.

gPhoton.gQuery.**stimcount**(*band, t0, t1, margin=[90.01, 90.01], aspum=0.06875493199999999, eclipse=None, null=True*)
Return stim counts.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **t0** (*long*) – The minimum time stamp to search.
- **t1** (*long*) – The maximum time stamp to search.
- **margin** (*list*) – X and Y lengths, in arcseconds, of a box within which to search for stim events.
- **aspum** (*float*) – Arcseconds per micrometer (of detector).
- **eclipse** (*int*) – The eclipse to return stim counts for.
- **null** (*bool*) – If true, query the NULL table.

Returns str – The query to submit to the database.

gPhoton.gQuery.**stimtimes**(*band, t0, t1, margin=[90.01, 90.01], aspum=0.06875493199999999, eclipse=None*)
Return stim counts.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **t0** (*long*) – The minimum time stamp to search.
- **t1** (*long*) – The maximum time stamp to search.
- **margin** (*list*) – X and Y lengths, in arcseconds, of a box in which to search for stim values.
- **aspum** (*float*) – Arcseconds per micrometer (on detector).
- **eclipse** (*int*) – The eclipse to return stim counts for.

Returns str – The query to submit to the database.

gPhoton.gQuery.**uniquetimes**(*band*, *t0*, *t1*, *flag=False*, *null=False*)
Return the unique timestamps for events within trange.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **t0** (*long*) – The minimum time stamp to search.
- **t1** (*long*) – The maximum time stamp to search.
- **flag** (*bool*) – If true, only return flag=0 data. Else return all non-NULL.
- **null** – If true, query the null table.

Returns str – The query to submit to the database.

galextools

gPhoton.galextools.**apcorrect1**(*radius*, *band*)

Compute an apeture correction. First way. Uses the table data in Figure 4 from Morissey, et al., 2007

Parameters

- **radius** (*float*) – The photometric radius, in degrees.
- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.

Returns float – The aperture correction.

gPhoton.galextools.**apcorrect2**(*radius*, *band*)

Compute an aperture correction in mag based upon an aperture radius in degrees. Second way. Uses the data in Table 1 from http://www.galex.caltech.edu/researcher/techdoc-ch5.html

Parameters

- **radius** (*float*) – The photometric radius, in degrees.
- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.

Returns float – The aperture correction.

gPhoton.galextools.**aper2deg**(*apercode*)
Convert SExtractor APER numbers to decimal degrees radii.

Parameters **apercode** (*int*) – The SExtractor APER number to convert.

Returns float – The APER radii in decimal degrees.

gPhoton.galextools.**compute_flat_scale**(*t*, *band*, *verbose=0*)

Return the flat scale factor for a given time. These are empirically determined linear scales to the flat field as a function of time due to diminished detector response. They were determined by Tom Barlow and are in the original GALEX pipeline but there is no published source of which I am aware.

Parameters

- **t** (*numpy.ndarray*) – Time stamp(s) to retrieve the scale factor for.
- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.

Returns numpy.ndarray

gPhoton.galextools.**counts2flux**(*cps*, *band*)

Converts GALEX counts per second to flux (erg sec^-1 cm^-2 A^-1). See: http://asd.gsfc.nasa.gov/archive/galex/FAQ/counts_background.html

Parameters

- **cps** (*float*) – The flux in counts per second.
- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.

Returns float – The converted flux in erg sec^-1 cm^-2 A^-1.

gPhoton.galextools.**counts2mag**(*cps*, *band*)

Converts GALEX counts per second to AB magnitudes. See: http://asd.gsfc.nasa.gov/archive/galex/FAQ/counts_background.

Parameters

- **cps** (*float*) – The flux in counts per second.
- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.

Returns float – The converted flux in AB magnitudes.

gPhoton.galextools.**deg2pix**(*skypos*, *skyrange*, *pixsz=0.000416666666666667*)

Converts degrees to GALEX pixels rounded up to the nearest pixel so that the number of degrees specified will fully fit into the frame.

Parameters

- **skypos** (*list*) – The right ascension and declination, in degrees.
- **skyrange** (*list*) – Values in degrees RA and Dec of a box around skypos that defines the extent of the region of interest.
- **pixsz** (*float*) – Width of a GALEX pixel, in degrees.

Returns float – The converted number of pixels.

gPhoton.galextools.**detbg**(*area*, *band*)

Nominal background in counts per second per 1.5" pixel. See: http://asd.gsfc.nasa.gov/archive/galex/FAQ/counts_backgroun

Parameters

- **area** (*float*) – The area to calculate the background in square degrees.

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.

Returns float – The nominal background for the given band, in counts per second.

gPhoton.galextools.**flat_scale_parameters**(*band*)
Return the flat scaling parameters for a given band.

Parameters **band** (*str*) – The band to use, either 'FUV' or 'NUV'.

Returns tuple – A three-element tuple containing the flat scaling parameters.

gPhoton.galextools.**isPostCSP**(*t*, *switch=961986575.0*)

Given a GALEX time stamp, return TRUE if it corresponds to a "post-CSP" eclipse. The actual CSP was on eclipse 37423, but the clock change (which matters more for calibration purposes) occured on 38268 (t~=961986575.)

Parameters

- **t** (*float*) – The time stamp to test.
- **switch** (*float*) – The GALEX time stamp that defines pre- and post-CSP.

Returns bool – Does this time correspond to a post-CSP eclipse?

gPhoton.galextools.**local_nl_correction**(*mr*, *band*)

Measured counts per second to predicted counts per sectond. Attempts to correct a measured count rate for nonlinearity per the formula given in Fig. 8 of Morrissey 2007.

gPhoton.galextools.**mag2counts**(*mag*, *band*)

Converts AB magnitudes to GALEX counts per second. See: http://asd.gsfc.nasa.gov/archive/galex/FAQ/counts_background.

Parameters

- **mag** (*float*) – The AB magnitude to convert.
- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.

Returns float – The converted flux in counts per second.

gPhoton.galextools.**photometric_repeatability**(*cps*, *expt*, *band*)

Estimate the photometric repeatability vs. magnitude . See: http://asd.gsfc.nasa.gov/archive/galex/FAQ/counts_background.h

Parameters

- **cps** (*float*) – Source flux in counts per second.
- **expt** (*float*) – Effective exposure time, in seconds.
- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.

Returns float – Estimated photometric repeatability based on flux.

gPhoton.galextools.**recovery_tranges**()

Defines and returns an array of time ranges during which the spacecraft was in some sort of recovery mode (e.g. FUV cycling or CSP) and therefore any data from these periods should be viewed skeptically (because of things like observing while not at HVNOM).

gPhoton.galextools.**zpmag**(*band*)
Define the zero point magnitude offset for the APER MCAT values.

Parameters **band** (*str*) – The band to use, either 'FUV' or 'NUV'.

Returns float – The zero point magnitude offset.

gnomonic

gPhoton.gnomonic.**gnomfwd_simple**(*ra*, *dec*, *ra0*, *dec0*, *crota*, *cdelt*, *cenpix*)
A forward gnomonic projection.

Parameters

- **ra** (*float numpy.ndarray*) – The right ascension of the event, in degrees.
- **dec** (*numpy.ndarray*) – The declination of the event, in degrees.
- **ra0** (*float*) – The right ascension of the boresight.
- **dec0** (*float*) – The declination of the boresight.
- **crota** (*float*) – Rotation of the FOV.
- **cdelt** (*float*) – Resolution at the center of the FOV.
- **cenpix** (*float*) – Coordinates of the center pixel.

Returns tuple – A two-element tuple containing the detector coordinates.

gPhoton.gnomonic.**gnomrev_simple**(*xi*, *eta*, *ra0*, *dec0*, *crota*, *cdelt*, *cenpix*)
A reverse gnomonic projection.

Parameters

- **xi** (*numpy.ndarray*) – Detector-space coordinte.
- **eta** (*numpy.ndarray*) – Detector-space coordinate.
- **ra0** (*float*) – The right ascension of the boresite, in degrees.
- **dec0** (*float*) – The declination of the boresite, in degrees.
- **crota** (*float*) – The field rotation.
- **cdelt** (*float*) – The resolution per pixel element at the field center.
- **cenpix** (*float*) – The coordinates of the center pixel.

Returns tuple – A two-element tuple containing the right ascension and declination, in degrees.

gpoton_args

gPhoton.gphoton_args.**check_common_args**(*args, function_name, valid_functions=['gaperture', 'gmap', 'gfind'], allow_no_coords=False*)

Checks validity of some command line arguments used in gFind, gAperture, gMap, etc. Returns the appropriate arguments as variables back to the calling procedure.

Parameters

- **args** (*argparse.ArgumentParser Namespace*) – Command-line options to check and modify.
- **function_name** (*str*) – Name of the function being called.
- **valid_functions** (*list*) – List of known/valid functions.

- **allow_no_coords** (*bool*) – Allow function to be called without coordinates?

Returns argparse.ArgumentParser Namespace – The updated command-line arguments.

gPhoton.gphoton_args.**common_args**(*parser, function_name, valid_functions=['gaperture', 'gmap', 'gfind']*)

Defines the arguments and options for the parser object when called from the command line. Accepts a string used to determine which arguments to add to the Parser object. Valid function names are "gfind", "gaperture", or "gmap" (all case insensitive).

Parameters

- **parser** (*argparse.ArgumentParser Namespace*) – Command-line options to check and modify.
- **function_name** (*str*) – Name of the function being called.
- **valid_functions** (*list*) – List of known/valid functions.

Returns argparse.ArgumentParser Namespace – The updated command-line arguments.

exception gPhoton.gphoton_args.**gPhotonArgsError**(*value*)
Bases: Exception

Exception class specific to gphoton_args.

gphoton_utils

gPhoton.gphoton_utils.**calculate_caldat**(*galex_time*)
Calculates a Gregorian calendar date given a GALEX time, in the UTC time standard.

Parameters **galex_time** (*float*) – A GALEX timestamp.

Returns float – The time converted to a Gregorian calendar date, in the UTC time standard.

gPhoton.gphoton_utils.**calculate_jd**(*galex_time*)
Calculates the Julian date, in the TDB time standard, given a GALEX time.

Parameters **galex_time** (*float*) – A GALEX timestamp.

Returns float – The time converted to a Julian date, in the TDB time standard.

gPhoton.gphoton_utils.**calculate_jd_tai**(*galex_time*)
Calculates the Julian date, in the TAI time standard, given a GALEX time.

Parameters **galex_time** (*float*) – A GALEX timestamp.

Returns float – The time converted to a Julian date, in the TAI time standard.

gPhoton.gphoton_utils.**calculate_jd_utc**(*galex_time*)
Calculates the Julian date, in the UTC time standard, given a GALEX time.

Parameters **galex_time** (*float*) – A GALEX timestamp.

Returns float – The time converted to a Julian date, in the UTC time standard.

gPhoton.gphoton_utils.**data_errors**(*catmag, band, t, sigma=3.0, mode='mag'*)
Given an array (of counts or mags), return an array of n-sigma error values.

Parameters

- **catmag** (*float*) – Nominal AB magnitude of the source.

- **t** (*list @CHASE - is this scalar or list? Also, consider trange instead of t to match first method?@*) – Set of integration times to compute the bounds on, in seconds.
- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **sigma** (*float*) – How many sigma out to set the bounds.
- **mode** (*str*) – Units in which to report bounds. Either 'cps' or 'mag'.

Returns tuple – A two-element tuple containing the lower and upper uncertainty, respectively.

gPhoton.gphoton_utils.**dmag_errors**(*t*, *band*, *sigma=3.0*, *mode='mag'*, *mags=array([13.*, *13.1*, *13.2*, *13.3*, *13.4*, *13.5*, *13.6*, *13.7*, *13.8*, *13.9*, *14.*, *14.1*, *14.2*, *14.3*, *14.4*, *14.5*, *14.6*, *14.7*, *14.8*, *14.9*, *15.*, *15.1*, *15.2*, *15.3*, *15.4*, *15.5*, *15.6*, *15.7*, *15.8*, *15.9*, *16.*, *16.1*, *16.2*, *16.3*, *16.4*, *16.5*, *16.6*, *16.7*, *16.8*, *16.9*, *17.*, *17.1*, *17.2*, *17.3*, *17.4*, *17.5*, *17.6*, *17.7*, *17.8*, *17.9*, *18.*, *18.1*, *18.2*, *18.3*, *18.4*, *18.5*, *18.6*, *18.7*, *18.8*, *18.9*, *19.*, *19.1*, *19.2*, *19.3*, *19.4*, *19.5*, *19.6*, *19.7*, *19.8*, *19.9*, *20.*, *20.1*, *20.2*, *20.3*, *20.4*, *20.5*, *20.6*, *20.7*, *20.8*, *20.9*, *21.*, *21.1*, *21.2*, *21.3*, *21.4*, *21.5*, *21.6*, *21.7*, *21.8*, *21.9*, *22.*, *22.1*, *22.2*, *22.3*, *22.4*, *22.5*, *22.6*, *22.7*, *22.8*, *22.9*, *23.*, *23.1*, *23.2*, *23.3*, *23.4*, *23.5*, *23.6*, *23.7*, *23.8*, *23.9]*)

Given an exposure time, give dmag error bars at a range of magnitudes.

Parameters

- **t** (*list @CHASE - is this scalar or list? Also, consider trange instead of t to match first method?@*) – Set of integration times to compute the bounds on, in seconds.
- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **sigma** (*float*) – How many sigma out to set the bounds.
- **mode** (*str*) – Units in which to report bounds. Either 'cps' or 'mag'.
- **mags** (*numpy.ndarray*) – Set of magnitudes to compute uncertainties on.

Returns tuple – A three-element tuple containing the magnitudes and their lower and upper uncertainties, respectively.

gPhoton.gphoton_utils.**model_errors**(*catmag, band, sigma=3.0, mode='mag', trange=[1, 1600]*)

Give upper and lower expected bounds as a function of the nominal magnitude of a source. Very useful for identifying outliers.

Parameters

- **catmag** (*float*) – Nominal AB magnitude of the source.
- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **sigma** (*float*) – How many sigma out to set the bounds.
- **mode** (*str*) – Units in which to report bounds. Either 'cps' or 'mag'.
- **trange** (*list*) – Set of integration times to compute the bounds on, in seconds.

Returns tuple – A two-element tuple containing the lower and upper bounds, respectively.

gPhoton.gphoton_utils.**plot_lc**(*data_frame*)

Plots a lightcurve from a CSV file data_frame - pandas DataFrame from read_lc()

gPhoton.gphoton_utils.**read_lc**(*csvfile*, *comment='|'*)
Read a light curve csv file from gAperture.

Parameters

- **csvfile** (*str*) – The name of the csv file to read.
- **comment** (*str*) – The character used to denote a comment row.

Returns pandas DataFrame – The contents of the csv file.

imagetools

gPhoton.imagetools.**create_image**(*band*, *skypos*, *tranges*, *skyrange*, *framesz=0*, *verbose=0*, *memlight=None*, *coadd=False*, *response=False*, *hdu=None*, *retries=100*, *detsize=1.1*)

Generate count or intensity images or movies at a given sky position and across given time ranges.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **skypos** (*list*) – The right ascension and declination, in degrees.
- **tranges** (*list*) – Set of time ranges to retrieve the photon events, in GALEX time seconds.
- **skyrange** (*list*) – RA and Dec extents of the region of interest in degrees.
- **framesz** (*float*) – The time bin size (depth) to use per frame, in seconds.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **memlight** (*float*) – Reduce memory usage by breaking query into smaller segments of this size in seconds.
- **coadd** (*bool*) – Integrate across all time ranges. (i.e. make a coadd)
- **response** (*bool*) – Apply the relative response correction.
- **hdu** (*bool*) – An existing HDU to modify.
- **retries** (*int*) – Number of query retries to attempt before giving up.
- **detsize** (*float*) – Effective diameter, in degrees, of the field-of-view.

Returns numpy.ndarray – The image file.

gPhoton.imagetools.**define_wcs**(*skypos*, *skyrange*, *verbose=0*, *pixsz=0.000416666666666667*)
Define the world coordinate system (WCS).

Parameters

- **skypos** (*list*) – The right ascension and declination, in degrees.
- **skyrange** (*list*) – Extent of the region of interest, in degrees.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **pixsz** (*float*) – Size of a GALEX pixel, in degrees.

Returns astropy.wcs WCS Object – The WCS information.

gPhoton.imagetools.**fits_header**(*band*, *skypos*, *tranges*, *skyrange*, *verbose=0*, *hdu=None*, *retries=100*)

Populate a FITS header.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **skypos** (*list*) – The right ascension and declination, in degrees.
- **tranges** (*list*) – Set of time ranges to retrieve the photon events, in GALEX time seconds.
- **skyrange** (*list*) – Extent of the region of interest, in degrees.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **hdu** (*bool*) – An existing HDU to modify.
- **retries** (*int*) – Number of query retries to attempt before giving up.

Returns astropy.fits.BinTableHDU object – The modified HDU header.

gPhoton.imagetools.**integrate_map**(*band*, *skypos*, *tranges*, *skyrange*, *verbose=0*, *memlight=None*, *hdu=None*, *retries=100*, *response=False*, *detsize=1.1*)

Integrate an image over some number of time ranges. Use a reduced memory optimization (at the expense of more web queries) if requested.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **skypos** (*list*) – The right ascension and declination, in degrees.
- **tranges** (*list*) – Set of time ranges to retrieve the photon events, in GALEX time seconds.
- **skyrange** (*list*) – RA and Dec extents of the region of interest in degrees.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **memlight** (*float*) – Reduce memory usage by breaking query into smaller segments of this size in seconds.
- **hdu** (*bool*) – An existing HDU to modify.
- **retries** (*int*) – Number of query retries to attempt before giving up.
- **response** (*bool*) – Apply the response correction.
- **detsize** (*float*) – Effective diameter, in degrees, of the field-of-view.

Returns numpy.ndarray - The integrated image.

gPhoton.imagetools.**makemap**(*band*, *skypos*, *trange*, *skyrange*, *response=False*, *verbose=0*, *detsize=1.1*)

Generate a single image frame.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **skypos** (*list*) – The right ascension and declination, in degrees.
- **trange** (*list*) – Minimum and maximum time to use, in GALEX time seconds.
- **skyrange** (*list*) – RA and Dec extent of the region of interest in degrees.

- **response** (*bool*) – Apply the response correction.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **detsize** (*float*) – Effective diameter, in degrees, of the field-of-view.

Returns numpy.ndarray - The bi-dimensional histogram of ra and dec.

gPhoton.imagetools.**movie**(*band*, *skypos*, *tranges*, *skyrange*, *framesz=0*, *verbose=0*, *memlight=None*, *coadd=False*, *response=False*, *hdu=None*, *retries=100*, *detsize=1.1*)

Generate a movie (mov) file.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **skypos** (*list*) – The right ascension and declination, in degrees.
- **tranges** (*list*) – Set of time ranges to retrieve the photon events, in GALEX time seconds.
- **skyrange** (*list*) – RA and Dec extents of the region of interest in degrees.
- **framesz** (*float*) – The time bin size (depth) to use per frame, in seconds.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **memlight** (*float*) – Reduce memory usage by breaking query into smaller segments of this depth in seconds.
- **coadd** (*bool*) – Integrated across all time ranges. (i.e. create a coadd)
- **response** (*bool*) – Apply the response correction.
- **hdu** (*bool*) – An existing HDU to modify.
- **retries** (*int*) – Number of query retries to attempt before giving up.
- **detsize** (*float*) – Effective diameter, in degrees, of the field-of-view.

Returns numpy.ndarray – The movie file.

gPhoton.imagetools.**movie_tbl**(*band*, *tranges*, *verbose=0*, *framesz=0.0*, *retries=100*)

Initialize a FITS table to contain movie frame information.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **tranges** (*list*) – Set of time ranges to retrieve the photon events, in GALEX time seconds.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **framesz** (*float*) – The time bin size (depth)to use per frame, in seconds.
- **retries** (*int*) – Number of query retries to attempt before giving up.

Returns astropy.fits.BinTableHDU object – The set of frames as an HDU object.

gPhoton.imagetools.**write_images**(*band*, *skypos*, *tranges*, *skyrange*, *write_cnt=None*, *write_int=None*, *framesz=0*, *verbose=0*, *memlight=None*, *coadd=False*, *overwrite=None*, *retries=100*, *write_cnt_coadd=False*, *write_int_coadd=False*, *detsize=1.1*)

Generate a write various maps to files.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.

- **skypos** (*list*) – The right ascension and declination, in degrees.
- **tranges** (*list*) – Set of time ranges to retrieve the photon events, in GALEX time seconds.
- **skyrange** (*list*) – RA and Dec extents of the region of interest in degrees.
- **write_cnt** (*bool*) – Make count image?
- **write_int** (*bool*) – Make intensity image?
- **framesz** (*float*) – The time bin size (depth) to use per frame, in seconds
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **memlight** (*float*) – Reduce memory usage by breaking query into smaller segments of this size in seconds.
- **coadd** (*bool*) – Integrate across all time ranges (i.e. create a coadd)
- **overwrite** (*bool*) – Overwrite existing output files?
- **retries** (*int*) – Number of query retries to attempt before giving up.
- **write_cnt_coadd** (*bool*) – Make count coadd image?
- **write_int_coadd** (*bool*) – Make intensity coadd image?
- **detsize** (*float*) – Effective diameter, in degrees, of the field-of-view.

gPhoton.imagetools.**write_jpeg**(*filename*, *band*, *skypos*, *tranges*, *skyrange*, *stepsz=1.0*, *overwrite=None*, *verbose=0*, *retries=100*)

Write a 'preview' jpeg image from a count map.

Parameters

- **filename** (*str*) – Name of output jpeg to make.
- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **skypos** (*list*) – The right ascension and declination, in degrees.
- **tranges** (*list*) – Set of time ranges to retrieve the photon events, in GALEX time seconds.
- **skyrange** (*list*) – RA and Dec extent of the region of interest in degrees.
- **stepsz** (*float*) – Time bin size to use, in seconds.
- **overwrite** (*bool*) – Overwrite existing output files?
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.
- **retries** (*int*) – Number of query retries to attempt before giving up.

regtestutils

gPhoton.regtestutils.**construct_row**(*i*, *band*, *objid*, *mcat*, *data*)

Assemble gAperture and MCAT data into a CSV row.

Parameters

- **i** (*int*) – The index of the row to collect values for.
- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.

- **objid** (*long*) – The GALEX MCAT object ID.
- **mcat** (*dict*) – Object containing MCAT data.
- **data** (*dict*) – Object containing gAperture data.

Returns tuple – The CSV row to output.

gPhoton.regtestutils.**datamaker**(*band, skypos, outfile, maglimit=20.0, margin=0.005, searchradius=0.1, radius=0.0016666666666666668, annulus=[0.0083, 0.025], verbose=0*)

Generate gAperture photometry for MCAT sources within a specified region.

Parameters

- **band** (*str*) – The band to use, either 'FUV' or 'NUV'.
- **skypos** (*list*) – The right ascension and declination, in degrees.
- **outfile** (*str*) – Name of output file to make.
- **maglimit** (*float*) – Faint limit to use, in AB Mag.
- **margin** (*float*) – The margin within which two sources are consider "the same," in degrees.
- **searchradius** (*float*) – The radius within which to search for sources, degrees.
- **radius** (*float*) – The size of the aperture to measure fluxes with, in degrees.
- **annulus** (*float*) – The inner and outer radii of the background annulus in degrees.
- **verbose** (*int*) – Verbosity level, a value of 0 is minimum verbosity.

gPhoton.regtestutils.**file_setup**(*outfile*)

Checks for a CSV file in which to put all the data and initializes it if it hasn't already been created. Loads already processed data, if any, in order to continue interrupted runs.

Parameters **outfile** (*str*) – Name of output file to make.

Returns numpy.ndarray – Set of object IDs that are already processed.

CHAPTER 6

Indices and tables

- genindex
- modindex
- search

c

d

f

g

i

m

p

r

C

D

F

G

I

M

P

R