"Tutorial of the adaptive-binning method for Fermi-LAT lightcurves"

Benoit Lott, CENBG, lott@cenbg.in23.fr

June 17, 2020

This document describes the usage of the scripts of the adaptive-binning method presented in Lott et al., 2012, A&A 544, 6L (http://adsabs.harvard.edu/abs/2012A&A...544A...6L).

Cautionary foreword

- The adaptive-binning method is not meant to provide a sensitive test of variability. If you want to test as to whether a source is variable (as opposed to steady), you should consider using other methods.
- The current script is not optimized to be a "flare-finder" per say.
- This script does not handle source confusion at all. If the source of interest sits next to a bright source, the results are likely to be incorrect (i.e., the criterion won't be properly satisfied in some or all bins).
- The script has not been tested for high, non-smooth background regions. Unexpected results may occur under these conditions.

Description of the procedure

The files needed to generate the bins with constant relative flux uncertainty/significance are included in this tar file (https://www.slac.stanford.edu/∼lott/ABM_mult_P8.tar.gz). The Fermi Science Tools (ST) must be installed as some scripts call ST libraries (e.g., pyIrfLoader).

The procedure includes 5 steps (the first four correspond to "Step 1" in the Lott et al. paper, and the fifth one to "Step 2") :

1. Create the input file

   The input parameters are contained in ascii file, an example of which (input_J1246.7-2546.txt) is given here (https://www.slac.stanford.edu/∼lott/input_J1246.7-2546.txt).

parameter	comment
ft2file = ft2_uptodate.fits	name of the spacecraft file covering the whole time range
ft1file = ft1_J1246.7-2546_filt.fits	name of the gtselected photon file[1].
MacroC = TS_estimate_P7.C	leave unchanged
rspfunc= P8R3_SOURCE_V2	must match the selection of the FT1 file[2] .
critval = 20.0	criterion: target relative flux uncertainty (here 20%) if crit=1 or target TS if crit=0
crit = 1	0: bin widths based on constant TS, 1: bin widths based on constant relative flux uncertainty
reverse = 1	time arrow. 0: normal, 1: reversed
expfile = exp_J1246.7-2546.txt	name of the output exposure file
anafile= times_J1246.7-2546.txt	name of the output file containing the bin times
SRC_RA = 191.68	source ra
SRC_DEC = -25.78	source dec
Index = 2.10	source photon spectral index [3]
Flux = 1.43e-07	source seed flux above 100 MeV (ph $cm^{-2}s^{-1}$)[4]
Elowerl = 269.566	lower energy limit of the reported integral flux [5]
normeg=1	normalization of EG background [6]
normgal=1	normalization of GAL background [6]
indexgal=0	spectral index of GAL background [6]

[1] typically centered on the source location with a 10 deg. radius.

[2] Use only P8R3_SOURCE_V2 with this release.

[3] Usually taken from the latest FGL catalog. It should be a realistic value for the source considered.

[4] This flux must be taken from the latest FGL catalog if the optimum energy is to be computed (see [5]). Otherwise any other choice will do.

[5] It is recommended to use the optimum energy as the lower limit (Elowerl) of the integral flux, but other values will do as well. The optimum energy can be computed as follows (from

the flux and index listed in the input file) > python comp_E1.py input_file.txt
In order to obtain "optimum" results, the value of the "Flux" parameter should be made close to the average flux over the considered time period.

[6] leave unchanged, unless you know what you are doing.

2. Create the ROI ft1 file, using gtselect+gtmktime; The photon list must be time ordered, this is not always so in the FSSC files. Use: fsort ft1_file_name TIME method = "heap"

3. Compute exposure > python comp_exposure_phi.py input_file.txt

   The exposure file (exp_J1246.7-2546.txt in the above example) is created during this step.

4. Compute the time bins

   python time_estimate_v3.py input_file.txt

5. Perform the standard gtlike analysis using the time bins obtained in step 4 to obtain publishable fluxes.

Option

In order to improve the accuracy of the method in case another bright source lies close to the source of interest, the flux of the confusing source must be taken into account. In the current implementation, a maximum of two confusing sources can be included. This is done by adding two parameters to the input file: "sources_in_roi" and "dir_sources_in_roi" pointing to a file listing the confusing sources and to the directory including that file. The file can be automatically generated with the script comp_confusion_4LAC.py, which for a set of sources listed in the file select_mult.txt, look up the 4FGL file gll_psc_v22.fit for overlapping sources. See sources_in_roi_J0221.0+3555.txt for an example.