

Example_LLH_setup_fixed_dirs

July 27, 2022

```
[1]: import numpy as np
from astropy.io import fits
from astropy.table import Table, vstack
from astropy.wcs import WCS
import os
from scipy import optimize, stats, interpolate
from scipy.integrate import quad
import argparse
import time
import multiprocessing as mp
import matplotlib.pyplot as plt
from matplotlib.colors import LogNorm
from matplotlib import cm
import sys
import pandas as pd
pd.options.display.max_columns = 250
pd.options.display.max_rows = 250
import healpy as hp
from copy import copy, deepcopy
# sys.path.append('BatML/')
import logging, traceback
import sys
#logging.basicConfig(stream=sys.stdout, level=logging.DEBUG)
```

WARNING: version mismatch between CFITSIO header (v4.000999999999999) and linked library (v4).

WARNING: version mismatch between CFITSIO header (v4.000999999999999) and linked library (v4).

WARNING: version mismatch between CFITSIO header (v4.000999999999999) and linked library (v4).

```
#NITRATES_path = '/Users/tparsota/Documents/BAT_SCRIPTS/NITRATES/NITRATES_GIT/NITRATE
```

```
#!/%cd {NITRATES_path}
```

The example notebooks need access to the F646018360 directory. also need to untar the F646018360_results.tar.gz file.

first untar the file and then cp the files in the F646018360 directory to the untarred directory. Then the user has to run this notebook first and then the example_reading_results notebook.

```
[2]: # if the response directories are all put in the same directory
# and given as an env variable (like in this cell), then config.py will
# be able to find them correctly as long as they keep the original names
# (Ex: ray trace directotry is still called 'ray_traces_detapp_npy')
# These paths can also just be hardcoded into config.py
# most of these paths can also be given directly as an argument to
# some analysis objects, like Source_Model_InOutFoV()
os.environ['NITRATES_RESP_DIR'] = '/Users/tparsota/Documents/BAT_SCRIPTS/
↳NITRATES_BAT_RSP_FILES/'
```

```
[3]: #from config import rt_dir
#from ray_trace_funcs import RayTraces
#from event2dpi_funcs import det2dpis, mask_detxy
#from flux_models import Cutoff_Plav_Flux, Plav_Flux, get_eflux_from_model
#from models import Source_Model_InOutFoV, Bkg_Model_wFlatA, \
#      CompoundModel, Point_Source_Model_Binned_Rates, im_dist
#from ray_trace_funcs import RayTraces
#from sqlite_funcs import get_conn
#from dbread_funcs import get_info_tab, get_twinds_tab
#from hp_funcs import ang_sep
#from coord_conv_funcs import theta_phi2imxy, imxy2theta_phi, \
↳convert_imxy2radec, \
#      convert_radec2thetaphi, convert_radec2imxy
#from do_llh_inFoV4realtime2 import parse_bkg_csv
#from LLH import LLH_webins
#from minimizers import NLLH_ScipyMinimize_Wjacob
```

```
[4]: #%matplotlib inline
import nitrates
from nitrates.config import rt_dir, solid_angle_dpi_fname
from nitrates.lib import get_conn, det2dpi, mask_detxy, get_info_tab, \
↳get_twinds_tab, ang_sep, theta_phi2imxy, \
    imxy2theta_phi, convert_imxy2radec, convert_radec2thetaphi, \
↳convert_radec2imxy
from nitrates.response import RayTraces
from nitrates.models import Cutoff_Plav_Flux, Plav_Flux, get_eflux_from_model, \
↳Source_Model_InOutFoV, \
    Bkg_Model_wFlatA, CompoundModel, Point_Source_Model_Binned_Rates, im_dist
from nitrates.llh_analysis import parse_bkg_csv, LLH_webins, \
↳NLLH_ScipyMinimize_Wjacob
```

```
print(nitrates.config.NITRATES_RESP_DIR)
```

```
sh: https://heasarc.gsfc.nasa.gov/FTP/caldb/caldbinit.sh: No such file or directory
```

```
No module named 'ROOT'
```

```
Please install the Python ROOT package to be able to run the full forward modeling calculations.
```

```
16
```

```
16
```

```
[array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14]), array([19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32]), array([37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50]), array([55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68]), array([73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86]), array([ 91,  92,  93,  94,  95,  96,  97,  98,  99, 100, 101, 102, 103, 104]), array([109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122]), array([127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140]), array([145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158]), array([163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176]), array([181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194]), array([199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212]), array([217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230]), array([235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248]), array([253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266]), array([271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284])]
```

```
(173, 286) (173, 286)
```

```
285 172
```

```
16
```

```
16
```

```
[array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14]), array([19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32]), array([37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50]), array([55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68]), array([73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86]), array([ 91,  92,  93,  94,  95,  96,  97,  98,  99, 100, 101, 102, 103, 104]), array([109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122]), array([127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138,
```

```

139,
    140]), array([145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156,
157,
    158]), array([163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174,
175,
    176]), array([181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192,
193,
    194]), array([199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210,
211,
    212]), array([217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228,
229,
    230]), array([235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246,
247,
    248]), array([253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264,
265,
    266]), array([271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282,
283,
    284])]
(173, 286) (173, 286)
285 172
16
16
[array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14]), array([19, 20,
21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32]), array([37, 38, 39, 40, 41, 42,
43, 44, 45, 46, 47, 48, 49, 50]), array([55, 56, 57, 58, 59, 60, 61, 62, 63, 64,
65, 66, 67, 68]), array([73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85,
86]), array([ 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103,
    104]), array([109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120,
121,
    122]), array([127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138,
139,
    140]), array([145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156,
157,
    158]), array([163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174,
175,
    176]), array([181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192,
193,
    194]), array([199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210,
211,
    212]), array([217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228,
229,
    230]), array([235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246,
247,
    248]), array([253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264,
265,
    266]), array([271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282,
283,
    284])]

```

(173, 286) (173, 286)

285 172

16

16

```
[array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14]), array([19, 20,
21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32]), array([37, 38, 39, 40, 41, 42,
43, 44, 45, 46, 47, 48, 49, 50]), array([55, 56, 57, 58, 59, 60, 61, 62, 63, 64,
65, 66, 67, 68]), array([73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85,
86]), array([ 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103,
104]), array([109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120,
121,
122]), array([127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138,
139,
140]), array([145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156,
157,
158]), array([163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174,
175,
176]), array([181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192,
193,
194]), array([199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210,
211,
212]), array([217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228,
229,
230]), array([235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246,
247,
248]), array([253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264,
265,
266]), array([271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282,
283,
284])]
```

(173, 286) (173, 286)

285 172

16

16

```
[array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10, 11, 12, 13, 14]), array([19, 20,
21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32]), array([37, 38, 39, 40, 41, 42,
43, 44, 45, 46, 47, 48, 49, 50]), array([55, 56, 57, 58, 59, 60, 61, 62, 63, 64,
65, 66, 67, 68]), array([73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85,
86]), array([ 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103,
104]), array([109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120,
121,
122]), array([127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138,
139,
140]), array([145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156,
157,
158]), array([163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174,
175,
176]), array([181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192,
```

```

193,
    194]), array([199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210,
211,
    212]), array([217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228,
229,
    230]), array([235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246,
247,
    248]), array([253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264,
265,
    266]), array([271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282,
283,
    284])]
(173, 286) (173, 286)
285 172
No module named 'ROOT'
Please install the Python ROOT package to be able to run the full forward
modeling calculations.
/Users/tparsota/Documents/BAT_SCRIPTS/NITRATES_BAT_RSP_FILES/

```

```
[5]: os.system(f'cp -r ./F646018360 {nitrates.config.NITRATES_RESP_DIR}')
```

```
[5]: 0
```

```
[6]: ebins0 = np.array([15.0, 24.0, 35.0, 48.0, 64.0])
ebins0 = np.append(ebins0, np.logspace(np.log10(84.0), np.log10(500.0), 5+1))[:
    ↪-1]
ebins0 = np.round(ebins0, decimals=1)[: -1]
ebins1 = np.append(ebins0[1:], [350.0])
nebins = len(ebins0)
print("Nebins: ", nebins)
print("ebins0: ", ebins0)
print("ebins1: ", ebins1)

# work_dir = '/storage/work/jjd330/local/bat_data/realtime_workdir/F646018360/'
NITRATES_path=nitrates.config.NITRATES_RESP_DIR
work_dir = os.path.join(NITRATES_path, 'F646018360')
conn = get_conn(os.path.join(work_dir, 'results.db'))
info_tab = get_info_tab(conn)
trigger_time = info_tab['trigtimeMET'][0]
print('trigger_time: ', trigger_time)

evfname = os.path.join(work_dir, 'filter_evdata.fits')
ev_data = fits.open(evfname)[1].data
GTI_PNT = Table.read(evfname, hdu='GTI_POINTING')
GTI_SLEW = Table.read(evfname, hdu='GTI_SLEW')
dmask = fits.open(os.path.join(work_dir, 'detmask.fits'))[0].data
attfile = fits.open(os.path.join(work_dir, 'attitude.fits'))[1].data

```

```

att_ind = np.argmin(np.abs(attfile['TIME'] - trigger_time))
att_quat = attfile['QPARAM'][att_ind]
print('attitude quaternion: ', att_quat)

ndets = np.sum(dmask==0)
print("Ndets: ", np.sum(dmask==0))

t_end = trigger_time + 1e3
t_start = trigger_time - 1e3
mask_vals = mask_detxy(dmask, ev_data)
bl_dmask = (dmask==0.)

bl_ev = (ev_data['EVENT_FLAGS']<1)&\
        (ev_data['ENERGY']<=500.)&(ev_data['ENERGY']>=14.)&\
        (mask_vals==0.)&(ev_data['TIME']<=t_end)&\
        (ev_data['TIME']>=t_start)

print("Nevents: ",np.sum(bl_ev))
ev_data0 = ev_data[bl_ev]

```

```

Nebins: 9
ebins0: [ 15.  24.  35.  48.  64.  84. 120. 171.5 245. ]
ebins1: [ 24.  35.  48.  64.  84. 120. 171.5 245. 350. ]
trigger_time: 646018383.1787
attitude quaternion: [-0.03597053  0.2345147 -0.64420835  0.72712074]
Ndets: 14932
Nevents: 1367885

```

```

[7]: ra, dec = 233.117, -26.213
      theta, phi = convert_radec2thetaphi(ra, dec, att_quat)
      print(theta, phi)
      imx, imy = convert_radec2imxy(ra, dec, att_quat)
      print(imx, imy)

```

```

38.54132137017975 137.65241966813443
-0.5887551341212709 -0.5366203642198198

```

```

[8]: flux_params = {'A':1.0, 'gamma':0.5, 'Epeak':1e2}
      flux_mod = Cutoff_Plaw_Flux(E0=100.0)

```

```

[9]: # use rt_dir from config.py or
      # rt_dir = '/path/to/ray_traces_detapp_npy/'
      rt_obj = RayTraces(rt_dir)

```

```

[10]: rt = rt_obj.get_intp_rt(imx, imy)
       print(np.shape(rt))

```

```

(173, 286)

```

```
[11]: %%time
# will by default use the resp directories from config.py
sig_mod = Source_Model_InOutFoV(flux_mod, [ebins0,ebins1], bl_dmask,\
                                rt_obj, use_deriv=True)

# or the paths can be given
# resp_tab_dname = '/path/to/resp_tabs_ebins/'
# hp_flor_resp_dname = '/path/to/hp_flor_resps/'
# comp_flor_resp_dname = '/path/to/comp_flor_resps/'
# sig_mod = Source_Model_InOutFoV(flux_mod, [ebins0,ebins1], bl_dmask,\
#                                 rt_obj, use_deriv=True,\
#                                 resp_tab_dname=resp_tab_dname,\
#                                 hp_flor_resp_dname=hp_flor_resp_dname,\
#                                 comp_flor_resp_dname=comp_flor_resp_dname)
```

CPU times: user 1.36 ms, sys: 1.26 ms, total: 2.61 ms

Wall time: 1.46 ms

```
[12]: %%time
sig_mod.set_theta_phi(theta, phi)
```

WARNING: hdu= was not specified but multiple tables are present, reading in first available table (hdu=1) [astropy.io.fits.connect]

WARNING:astropy:hdu= was not specified but multiple tables are present, reading in first available table (hdu=1)

(0.0, 30.48, -14.117)

(0.0, 30.48, -14.117)

/Users/tparsota/Documents/BAT_SCRIPTS/NITRATES/NITRATES_GIT/NITRATES/nitrates/response/response.py:1499: RuntimeWarning: invalid value encountered in true_divide

```
self.E_A0s = (self.orig_photonEs[self.Einds1] - self.photonEs) /\
```

(36.0, 45.0)

2.652419668134428

42.34758033186557

max rt: 0.8604

WARNING: hdu= was not specified but multiple tables are present, reading in first available table (hdu=1) [astropy.io.fits.connect]

WARNING:astropy:hdu= was not specified but multiple tables are present, reading in first available table (hdu=1)

initing ResponseDPI, with fname

/Users/tparsota/Documents/BAT_SCRIPTS/NITRATES_BAT_RSP_FILES/resp_tabs_ebins/drm_theta_36.0_phi_30.0.fits

initing ResponseDPI, with fname

/Users/tparsota/Documents/BAT_SCRIPTS/NITRATES_BAT_RSP_FILES/resp_tabs_ebins/drm_theta_36.0_phi_45.0.fits

initing ResponseDPI, with fname

/Users/tparsota/Documents/BAT_SCRIPTS/NITRATES_BAT_RSP_FILES/resp_tabs_ebins/drm


```

_theta_45.0_phi_30.0_.fits
initing ResponseDPI, with fname
/Users/tparsota/Documents/BAT_SCRIPTS/NITRATES_BAT_RSP_FILES/resp_tabs_ebins/drm
_theta_45.0_phi_45.0_.fits
CPU times: user 5.12 s, sys: 2.54 s, total: 7.65 s
Wall time: 7.44 s

```

```

[13]: #from nitrates.config import solid_angle_dpi_fname
# or give the path directly
# solid_angle_dpi_fname = '/path/to/solid_angle_dpi.npy'
bkg_fname = os.path.join(work_dir, 'bkg_estimation.csv')
solid_ang_dpi = np.load(solid_angle_dpi_fname)
bkg_df, bkg_name, PSnames, bkg_mod, ps_mods = parse_bkg_csv(bkg_fname, \
↳solid_ang_dpi, \
                                ebins0, ebins1, bl_dmask, rt_dir)
bkg_mod.has_deriv = False
bkg_mod_list = [bkg_mod]
Nsracs = len(ps_mods)
if Nsracs > 0:
    bkg_mod_list += ps_mods
    for ps_mod in ps_mods:
        ps_mod.has_deriv = False
    bkg_mod = CompoundModel(bkg_mod_list)
tmid = trigger_time
bkg_row = bkg_df.iloc[np.argmin(np.abs(tmid - bkg_df['time']))]
bkg_params = {pname:bkg_row[pname] for pname in \
                bkg_mod.param_names}
bkg_name = bkg_mod.name
bkg_params

```

```
['4U 1700-377', 'GR0 J1655-40', 'GX 339-4', 'Sco X-1']
```

```

[13]: {'Background_bkg_rate_0': 0.0913703220701183,
'Background_bkg_rate_1': 0.0661578002239374,
'Background_bkg_rate_2': 0.0400898569026105,
'Background_bkg_rate_3': 0.0394919934207801,
'Background_bkg_rate_4': 0.0346780266835281,
'Background_bkg_rate_5': 0.0351076584048365,
'Background_bkg_rate_6': 0.033602492003955,
'Background_bkg_rate_7': 0.0248610826847874,
'Background_bkg_rate_8': 0.0176399476145076,
'Background_flat_0': 0.0,
'Background_flat_1': 0.0,
'Background_flat_2': 0.1859632649428977,
'Background_flat_3': 0.0766344244533236,
'Background_flat_4': 0.3020855890115875,
'Background_flat_5': 0.7611510180823338,
'Background_flat_6': 0.8091430254293075,

```

```

'Background_flat_7': 1.0,
'Background_flat_8': 1.0,
'4U 1700-377_imx': -0.0981485305770971,
'4U 1700-377_imy': -0.4742076074486664,
'4U 1700-377_rate_0': 0.0124901359218975,
'4U 1700-377_rate_1': 0.0072425264886238,
'4U 1700-377_rate_2': 0.0045295645864628,
'4U 1700-377_rate_3': 0.0016876913514016,
'4U 1700-377_rate_4': 0.0013255091059552,
'4U 1700-377_rate_5': 0.0012956105576278,
'4U 1700-377_rate_6': 0.0005834459169539,
'4U 1700-377_rate_7': 5e-08,
'4U 1700-377_rate_8': 0.0001401320031663,
'GR0 J1655-40_imx': -0.1134159678653529,
'GR0 J1655-40_imy': -0.5359750413588411,
'GR0 J1655-40_rate_0': 0.0052771851384718,
'GR0 J1655-40_rate_1': 0.0022036634261929,
'GR0 J1655-40_rate_2': 0.0003149730275822,
'GR0 J1655-40_rate_3': 0.0017640644311276,
'GR0 J1655-40_rate_4': 5e-08,
'GR0 J1655-40_rate_5': 9.114980149845126e-05,
'GR0 J1655-40_rate_6': 3.88633504826533e-05,
'GR0 J1655-40_rate_7': 5e-08,
'GR0 J1655-40_rate_8': 0.0007082358100819,
'GX 339-4_imx': 0.0087293008614054,
'GX 339-4_imy': -0.6947927093434689,
'GX 339-4_rate_0': 0.0037893354510156,
'GX 339-4_rate_1': 0.0032148894308596,
'GX 339-4_rate_2': 0.0017605414459212,
'GX 339-4_rate_3': 0.0015331679372228,
'GX 339-4_rate_4': 0.0015777295548874,
'GX 339-4_rate_5': 0.0014172765864478,
'GX 339-4_rate_6': 0.0008599988841135,
'GX 339-4_rate_7': 0.0013329459515836,
'GX 339-4_rate_8': 0.0005777085477427,
'Sco X-1_imx': -0.4801793393603647,
'Sco X-1_imy': -0.2001566229807942,
'Sco X-1_rate_0': 0.0497219623187451,
'Sco X-1_rate_1': 0.0059213556150606,
'Sco X-1_rate_2': 5e-08,
'Sco X-1_rate_3': 0.000359259928782,
'Sco X-1_rate_4': 0.0004049393144718,
'Sco X-1_rate_5': 5e-08,
'Sco X-1_rate_6': 5e-08,
'Sco X-1_rate_7': 5e-08,
'Sco X-1_rate_8': 6.897789976904059e-05}

```

```
[14]: pars_ = {}
pars_['Signal_theta'] = theta
pars_['Signal_phi'] = phi
for pname,val in list(bkg_params.items()):
    pars_[bkg_name+'_'+pname] = val
for pname,val in list(flux_params.items()):
    pars_['Signal_'+pname] = val
pars_
```

```
[14]: {'Signal_theta': 38.54132137017975,
'Signal_phi': 137.65241966813443,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_Background_bkg_rate_0':
0.0913703220701183,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_Background_bkg_rate_1':
0.0661578002239374,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_Background_bkg_rate_2':
0.0400898569026105,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_Background_bkg_rate_3':
0.0394919934207801,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_Background_bkg_rate_4':
0.0346780266835281,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_Background_bkg_rate_5':
0.0351076584048365,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_Background_bkg_rate_6':
0.033602492003955,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_Background_bkg_rate_7':
0.0248610826847874,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_Background_bkg_rate_8':
0.0176399476145076,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_Background_flat_0': 0.0,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_Background_flat_1': 0.0,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_Background_flat_2':
0.1859632649428977,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_Background_flat_3':
0.0766344244533236,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_Background_flat_4':
0.3020855890115875,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_Background_flat_5':
0.7611510180823338,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_Background_flat_6':
0.8091430254293075,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_Background_flat_7': 1.0,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_Background_flat_8': 1.0,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_4U 1700-377_imx':
-0.0981485305770971,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_4U 1700-377_imy':
-0.4742076074486664,
```

'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_4U 1700-377_rate_0':
0.0124901359218975,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_4U 1700-377_rate_1':
0.0072425264886238,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_4U 1700-377_rate_2':
0.0045295645864628,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_4U 1700-377_rate_3':
0.0016876913514016,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_4U 1700-377_rate_4':
0.0013255091059552,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_4U 1700-377_rate_5':
0.0012956105576278,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_4U 1700-377_rate_6':
0.0005834459169539,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_4U 1700-377_rate_7':
5e-08,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_4U 1700-377_rate_8':
0.0001401320031663,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_GRO J1655-40_imx':
-0.1134159678653529,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_GRO J1655-40_imy':
-0.5359750413588411,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_GRO J1655-40_rate_0':
0.0052771851384718,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_GRO J1655-40_rate_1':
0.0022036634261929,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_GRO J1655-40_rate_2':
0.0003149730275822,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_GRO J1655-40_rate_3':
0.0017640644311276,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_GRO J1655-40_rate_4':
5e-08,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_GRO J1655-40_rate_5':
9.114980149845126e-05,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_GRO J1655-40_rate_6':
3.88633504826533e-05,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_GRO J1655-40_rate_7':
5e-08,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_GRO J1655-40_rate_8':
0.0007082358100819,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_GX 339-4_imx':
0.0087293008614054,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_GX 339-4_imy':
-0.6947927093434689,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_GX 339-4_rate_0':
0.0037893354510156,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_GX 339-4_rate_1':

```

0.0032148894308596,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_GX 339-4_rate_2':
0.0017605414459212,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_GX 339-4_rate_3':
0.0015331679372228,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_GX 339-4_rate_4':
0.0015777295548874,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_GX 339-4_rate_5':
0.0014172765864478,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_GX 339-4_rate_6':
0.0008599988841135,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_GX 339-4_rate_7':
0.0013329459515836,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_GX 339-4_rate_8':
0.0005777085477427,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_Sco X-1_imx':
-0.4801793393603647,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_Sco X-1_imy':
-0.2001566229807942,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_Sco X-1_rate_0':
0.0497219623187451,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_Sco X-1_rate_1':
0.0059213556150606,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_Sco X-1_rate_2': 5e-08,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_Sco X-1_rate_3':
0.000359259928782,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_Sco X-1_rate_4':
0.0004049393144718,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_Sco X-1_rate_5': 5e-08,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_Sco X-1_rate_6': 5e-08,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_Sco X-1_rate_7': 5e-08,
'Background+4U 1700-377+GR0 J1655-40+GX 339-4+Sco X-1_Sco X-1_rate_8':
6.897789976904059e-05,
'Signal_A': 1.0,
'Signal_gamma': 0.5,
'Signal_Epeak': 100.0}

```

```

[15]: comp_mod = CompoundModel([bkg_mod, sig_mod])

sig_miner = NLLH_ScipyMinimize_Wjacob('')

sig_llh_obj = LLH_webins(ev_data0, ebins0, ebins1, bl_dmask, has_err=True)

sig_llh_obj.set_model(comp_mod)

sig_miner.set_llh(sig_llh_obj)

```

```
[16]: fixed_pnames = list(pars_.keys())
fixed_vals = list(pars_.values())
trans = [None for i in range(len(fixed_pnames))]
sig_miner.set_trans(fixed_pnames, trans)
sig_miner.set_fixed_params(fixed_pnames, values=fixed_vals)
sig_miner.set_fixed_params(['Signal_A'], fixed=False)
```

```
[17]: %%time
flux_params['gamma'] = 0.8
flux_params['Epeak'] = 350.0
sig_mod.set_flux_params(flux_params)
```

```
[18]: t0 = trigger_time - 0.512
t1 = t0 + 2.048
sig_llh_obj.set_time(t0, t1)
```

```
[19]: %%time
pars, nllh, res = sig_miner.minimize()
```

```
[20]: print(res)
print(nllh)
print(pars)
```

```
[
  fun: 45894.070681000456
 hess_inv: <1x1 LbfgsInvHessProduct with dtype=float64>
  jac: array([0.00239975])
 message: 'CONVERGENCE: REL_REDUCTION_OF_F_<= _FACTR*EPSMCH'
  nfev: 8
  nit: 5
  njev: 8
  status: 0
  success: True
   x: array([0.006875])]
[45894.070681000456]
[[0.006875003655224625]]
```

```
[21]: %%time
pars_['Signal_A'] = 1e-10
bkg_nllh = -sig_llh_obj.get_logprob(pars_)
print (bkg_nllh)
sqrtTS = np.sqrt(2.*(bkg_nllh - nllh[0]))
print (sqrtTS)
```

```
46038.743327207194
17.01015262757734
CPU times: user 6.4 ms, sys: 2.14 ms, total: 8.54 ms
Wall time: 7.57 ms
```

[]:

[]:

[]:

[]: