Fermipy tutorial - PKS2155-304

This tutorial shows how to perform Fermi-LAT analysis with the Fermipy Python package. Many parts of this tutorial are taken directly from the documentation page of Fermipy: fermipy.readthedocs. I suggest to visit the documentation page to find further informations.

Fermipy is a python package created by Mattew Wood and maintained by a wide community of people. Fermipy facilitates analysis of data from the Large Area Telescope (LAT) with the Fermi Science Tools. The Fermipy package is built on the pyLikelihood interface of the Fermi Science Tools and provides a set of high-level tools for performing common analysis tasks:

- Data and model preparation with the gt-tools (gtselect, gtmktime, etc.).
- Extracting a spectral energy distribution (SED) of a source.
- Generating TS and residual maps for a region of interest.
- Finding new source candidates.
- Localizing a source or fitting its spatial extension.
- Perfoming phase analysis
- Calculating light curves
- Deriving the sensitivity of gamma-ray sources with a specific SED and spatial morphology.

Installation

Instruction on how to install in SLAC machines or in your laptop Fermipy are available at this page: fermipy.installation. Fermipy is only compatible with Science Tools v10r0p5 or later. If you are using an earlier version, you will need to download and install the latest version from the FSSC. Note that it is recommended to use the non-ROOT binary distributions of the Science Tools. These instructions assume that you want to run Fermipy on the SLAC machines.

With these instructions you will create your own Conda installation and you will install all the packages needed to use FermiPy and Science Tools. Using your own Conda installation avoids conflicts with package versions because you build your own environment.

First grab the installation and setup scripts from the fermipy github repository:

In [1]:
```
# ! curl -OL https://raw.githubusercontent.com/fermiPy/fermipy/master/condainsta
# ! curl -OL https://raw.githubusercontent.com/fermiPy/fermipy/master/slacsetup.
```

Now choose an installation path. This should be a new directory (e.g. $HOME/anaconda) that has at least 2-4 GB available. We will assign this location to the CONDABASE environment variable which is used by the setup script to find the location of your python installation. To avoid setting this every time you log in it's recommended to set CONDABASE into your .bashrc file.

Now run the following commands to install anaconda and fermipy. This will take about 5-10 minutes.

In [2]:
```
# export CONDABASE=<path to install directory>
# bash condainstall.sh $CONDABASE
```

Once anaconda is installed you will initialize your python and ST environment by running the slacsetup function in slacsetup.sh. This function will set the appropriate environment variables needed to run the STs and python.

In [3]:
```
# source slacsetup.sh
# slacsetup
```

For convenience you can also copy this function into your .bashrc file so that it will automatically be available when you launch a new shell session. By default the function will setup your environment to point to a recent version of the STs and the installation of python in CONDABASE. If CONDABASE is not defined then it will use the installation of python that is packaged with a given release of the STs. The slacsetup function takes two optional arguments which can be used to override the ST version or python installation path.

In [4]:
```
# Use ST 10-00-05
# slacsetup 10-00-05
# Use ST 11-01-01 and python distribution located at <PATH>
# slacsetup 11-01-01 <PATH>
```

The installation script only installs packages that are required by fermipy and the STs. Once you've initialized your shell environment you are free to install additional python packages with the conda package manager tool with conda install . Packages that are not available on conda can also be installed with pip.

conda install fermipy

conda can also be used to upgrade packages. For instance you can upgrade fermipy to the newest version with the conda update command:

In [5]:
```
# conda update fermipy
```

If you want to make development of Fermipy you should get the github version making:

In [6]:
```
# git clone https://github.com/fermiPy/fermipy.git
```

Then you should create a branch using:

In [7]:
```
# git checkout -b mattia-dev
```

In the branch you can make your development of Fermipy and then use the git commands to merge them to the master repository.

More informations on how to install Fermipy are available here: fermipy-install

Download the raw files

The files can be obtained from the Fermi SSC website. The following data selection choices were made:

- Object name or coordinates: 302.25,-44.37
- Coordinate system: GAL
- Search radius (degrees): 12
- Observation dates: 239557417,512994417
- Time system: MET
- Energy range (MeV): 100,500000
- LAT data type: Photon
- Spacecraft data: yes

The raw data with these selections can be downloaded using wget.

In [8]:
```
# # Set up a subdirectory for the data
# !mkdir -p SMC_data/PH
# !mkdir -p SMC_data/SC

# # Download the data and place into the subdirectory
# !wget -P ./SMC_data/PH/ https://fermi.gsfc.nasa.gov/FTP/fermi/data/lat/queries
# !wget -P ./SMC_data/PH/ https://fermi.gsfc.nasa.gov/FTP/fermi/data/lat/queries
# !wget -P ./SMC_data/PH/ https://fermi.gsfc.nasa.gov/FTP/fermi/data/lat/queries
# !wget -P ./SMC_data/PH/ https://fermi.gsfc.nasa.gov/FTP/fermi/data/lat/queries
# !wget -P ./SMC_data/PH/ https://fermi.gsfc.nasa.gov/FTP/fermi/data/lat/queries
# !wget -P ./SMC_data/PH/ https://fermi.gsfc.nasa.gov/FTP/fermi/data/lat/queries
# !wget -P ./SMC_data/PH/ https://fermi.gsfc.nasa.gov/FTP/fermi/data/lat/queries

# !wget -P ./SMC_data/SC/ https://fermi.gsfc.nasa.gov/FTP/fermi/data/lat/queries

# # Make a file list
# !ls ./SMC_data/PH/*PH*.fits > ./SMC_data/PH.txt
```

Note that processing the raw photon and spacecraft files can take multiple hours. Alternatively, you can download the pre-processed data from github.

In [9]:
```
#import os
#if os.path.isfile('./../data/SMC_data.tar.gz'):
#    !tar xzf ./../data/SMC_data.tar.gz
#else:
#    !wget -P ./../data/ https://raw.githubusercontent.com/fermiPy/fermipy-extra
#    !tar xzf ./../data/SMC_data.tar.gz
```

Configuration file:

The first step is to compose a configuration file that defines the data selection and analysis parameters. Fermipy uses YAML files to read and write its configuration in a persistent format. The configuration file has a hierarchical structure that groups parameters into dictionaries that are keyed to a section name (data, binning, etc.). Below I report a sample of configuration applied for an analysis of the SMC:

In [24]:
```
import os
import numpy as np
from fermipy.gtanalysis import GTAnalysis
from fermipy.plotting import ROIPlotter, SEDPlotter
import matplotlib.pyplot as plt
import matplotlib
```

In [25]:
```
!cat ./PKS2155_data/config.yaml
```


```
data:
  evfile : './PKS2155_data/PKS2155.lst'
  scfile : './PKS2155_data/L211025154921445A2D9929_SC00.fits'
#  ltcube : 'ltcube_00.fits'

binning:
  roiwidth   : 12.0
  binsz      : 0.08
  binsperdec : 8

selection :
  emin : 100
  emax : 300000
  tmin : 241401601
  tmax : 257385601
  zmax    : 105
  evclass : 128
  evtype  : 3
  target : 'PKS 2155-304'

gtlike:
  edisp : True
  irfs : 'P8R2_SOURCE_V6'
  edisp_disable : ['isodiff','galdiff']

model:
  src_roiwidth : 12.0
  galdiff  : '$FERMI_DIFFUSE_DIR/gll_iem_v07.fits'
  isodiff  : '$FERMI_DIFFUSE_DIR/iso_P8R2_SOURCE_V6_v06.txt'
  catalogs: gll_psc_v16.fit

fileio:
  usescratch: False
  outdir : PKS2155_data
```


The configuration file has the same structure as the configuration dictionary such that one can read/write configurations using the load/dump methods of the yaml module:

In [26]:
```
import yaml
# Load a configuration
config = yaml.load(open('./PKS2155_data/config.yaml'), Loader=yaml.FullLoader)
# Update a parameter and write a new configuration
```

```
config['selection']['emin'] = 1000.
yaml.dump(config, open('new_PKS_config.yaml','w'))
```

The data section defines the input data set and spacecraft file for the analysis. Here evfile points to a list of FT1 files that encompass the chosen ROI, energy range, and time selection. The parameters in the binning section define the dimensions of the ROI and the spatial and energy bin size. The selection section defines parameters related to the data selection (energy range, zmax cut, and event class/type). The target parameter in this section defines the ROI center to have the same coordinates as the given source. The model section defines parameters related to the ROI model definition (diffuse templates, point sources). Fermipy gives the user the option to combine multiple data selections into a joint likelihood with the components section. For more informations on this visit: http://fermipy.readthedocs.io/en/latest/quickstart.html

Note that the setup for a joint analysis is identical to the above except for the modification to the components section. The following example shows the components configuration one would use to define a joint analysis with the four PSF event types:

```
components:
  - { selection : { evtype : 4  } }
  - { selection : { evtype : 8  } }
  - { selection : { evtype : 16 } }
  - { selection : { evtype : 32 } }
```

Start your run

First of all you need to load the configuration file, create the object gta and run the tool gta.setup that implements the ST gtselect, gtmktime, gtbin, gtexpcube, gtsrcmap tools

Begin the setup routine. As most of the files are included, only the source-map should be created. Will take ~5min.

In [27]:

```
gta = GTAnalysis('new_PKS_config.yaml')
gta.setup()
```

```
2024-09-25 13:57:01 INFO    GTAnalysis.__init__():
--------------------------------------------------------------------------------
fermipy version 1.3.0
ScienceTools version 2.2.0
2024-09-25 13:57:01 INFO    GTAnalysis.setup(): Running setup.
2024-09-25 13:57:01 INFO    GTBinnedAnalysis.setup(): Running setup for component
00
2024-09-25 13:57:01 INFO    GTBinnedAnalysis.run_gtapp(): Running gtselect.
2024-09-25 13:57:01 INFO    GTBinnedAnalysis.run_gtapp(): time -p gtselect infile
=/home/chen/PHYS2015A_fermipy_working/notebooks/PKS2155_data/evfile_00.txt outfil
e=/home/chen/PHYS2015A_fermipy_working/notebooks/PKS2155_data/ft1_00.fits ra=329.
7203063964844 dec=-30.2268009185791 rad=8.985281374238571 tmin=241401601.0 tmax=2
57385601.0 emin=1000.0 emax=300000.0 zmin=0.0 zmax=105.0 evclass=128 evtype=3 con
vtype=-1 phasemin=0.0 phasemax=1.0 evtable="EVENTS" chatter=3 clobber=yes debug=n
o gui=no mode="ql"
2024-09-25 13:57:01 INFO    GTBinnedAnalysis.run_gtapp(): This is gtselect versio
n HEAD
```

```
{'Prefactor': 0, 'Index1': 1, 'Scale': 2, 'Cutoff': 3, 'Index2': 4}
{'Prefactor': 0, 'Index1': 1, 'Scale': 2, 'Cutoff': 3, 'Index2': 4}
```

```
2024-09-25 13:57:02 INFO    GTBinnedAnalysis.run_gtapp(): Applying filter string:
((EVENT_CLASS&o200) != o0) && angsep(RA,DEC,329.7203064,-30.22680092) < 8.9852813
74 && ((EVENT_TYPE&o3) != o0) && 1000 < ENERGY && ENERGY <= 300000 && 241401601 <
TIME && TIME <= 257385601 && 0 < ZENITH_ANGLE && ZENITH_ANGLE <= 105 && gtifilter
()
2024-09-25 13:57:02 INFO    GTBinnedAnalysis.run_gtapp(): Done.
2024-09-25 13:57:02 INFO    GTBinnedAnalysis.run_gtapp(): Finished gtselect. Exec
ution time: 0.17 s
2024-09-25 13:57:02 INFO    GTBinnedAnalysis.run_gtapp(): Running gtltcube.
2024-09-25 13:57:02 INFO    GTBinnedAnalysis.run_gtapp(): time -p gtltcube evfile
="/home/chen/PHYS2015A_fermipy_working/notebooks/PKS2155_data/ft1_00.fits" evtabl
e="EVENTS" scfile=./PKS2155_data/L211025154921445A2D9929_SC00.fits sctable="SC_DA
TA" outfile=/home/chen/PHYS2015A_fermipy_working/notebooks/PKS2155_data/ltcube_0
0.fits dcostheta=0.025 binsz=1.0 phibins=0 tmin=0.0 tmax=0.0 file_version="1" zmi
n=0.0 zmax=105.0 chatter=2 clobber=yes debug=no gui=no mode="ql"
2024-09-25 13:57:02 INFO    GTBinnedAnalysis.run_gtapp(): Working on file ./PKS21
55_data/L211025154921445A2D9929_SC00.fits
2024-09-25 13:59:16 INFO    GTBinnedAnalysis.run_gtapp(): .....................!
2024-09-25 13:59:16 INFO    GTBinnedAnalysis.run_gtapp(): Finished gtltcube. Exec
ution time: 134.72 s
2024-09-25 13:59:17 INFO    GTBinnedAnalysis.run_gtapp(): Running gtbin.
2024-09-25 13:59:17 INFO    GTBinnedAnalysis.run_gtapp(): time -p gtbin evfile=/h
ome/chen/PHYS2015A_fermipy_working/notebooks/PKS2155_data/ft1_00.fits scfile=./PK
S2155_data/L211025154921445A2D9929_SC00.fits outfile=/home/chen/PHYS2015A_fermipy
_working/notebooks/PKS2155_data/ccube_00.fits algorithm="ccube" ebinalg="LOG" emi
n=1000.0 emax=300000.0 enumbins=20 ebinfile=NONE tbinalg="LIN" tbinfile=NONE nxpi
x=150 nypix=150 binsz=0.08 coordsys="CEL" xref=329.7203063964844 yref=-30.2268009
185791 axisrot=0.0 rafield="RA" decfield="DEC" proj="AIT" hpx_ordering_scheme="RI
NG" hpx_order=3 hpx_ebin=yes hpx_region="" evtable="EVENTS" sctable="SC_DATA" efi
eld="ENERGY" tfield="TIME" chatter=3 clobber=yes debug=no gui=no mode="ql"
2024-09-25 13:59:17 INFO    GTBinnedAnalysis.run_gtapp(): This is gtbin version H
EAD
2024-09-25 13:59:17 INFO    GTBinnedAnalysis.run_gtapp(): Finished gtbin. Executi
on time: 0.14 s
2024-09-25 13:59:17 INFO    GTBinnedAnalysis.run_gtapp(): Running gtexpcube2.
2024-09-25 13:59:17 INFO    GTBinnedAnalysis.run_gtapp(): time -p gtexpcube2 infi
le=/home/chen/PHYS2015A_fermipy_working/notebooks/PKS2155_data/ltcube_00.fits cma
p=none outfile=/home/chen/PHYS2015A_fermipy_working/notebooks/PKS2155_data/bexpma
p_00.fits irfs="P8R2_SOURCE_V6" evtype=3 edisp_bins=-1 nxpix=360 nypix=180 binsz=
1.0 coordsys="CEL" xref=0.0 yref=0.0 axisrot=0.0 proj="CAR" ebinalg="LOG" emin=10
00.0 emax=300000.0 enumbins=20 ebinfile="NONE" hpx_ordering_scheme="RING" hpx_ord
er=6 bincalc="EDGE" ignorephi=no thmax=180.0 thmin=0.0 table="EXPOSURE" chatter=3
clobber=yes debug=no mode="ql"
2024-09-25 13:59:17 INFO    GTBinnedAnalysis.run_gtapp(): This is gtexpcube2 vers
ion HEAD
2024-09-25 13:59:17 INFO    GTBinnedAnalysis.run_gtapp(): Using evtype=3 (i.e., F
RONT/BACK irfs)
2024-09-25 13:59:17 INFO    GTBinnedAnalysis.run_gtapp(): ResponseFunctions::loa
d: IRF used: P8R2_SOURCE_V6
2024-09-25 13:59:17 INFO    GTBinnedAnalysis.run_gtapp(): event_types:  0  1
2024-09-25 13:59:19 INFO    GTBinnedAnalysis.run_gtapp(): Computing binned exposu
re map....................!
2024-09-25 13:59:19 INFO    GTBinnedAnalysis.run_gtapp(): Finished gtexpcube2. Ex
ecution time: 2.12 s
2024-09-25 13:59:19 INFO    GTBinnedAnalysis.run_gtapp(): Running gtexpcube2.
2024-09-25 13:59:19 INFO    GTBinnedAnalysis.run_gtapp(): time -p gtexpcube2 infi
le=/home/chen/PHYS2015A_fermipy_working/notebooks/PKS2155_data/ltcube_00.fits cma
p=none outfile=/home/chen/PHYS2015A_fermipy_working/notebooks/PKS2155_data/bexpma
p_roi_00.fits irfs="P8R2_SOURCE_V6" evtype=3 edisp_bins=0 nxpix=150 nypix=150 bin
sz=0.08 coordsys="CEL" xref=329.7203063964844 yref=-30.2268009185791 axisrot=0.0
```

```
proj="CAR" ebinalg="LOG" emin=1000.0 emax=300000.0 enumbins=20 ebinfile="NONE" hp
x_ordering_scheme="RING" hpx_order=6 bincalc="EDGE" ignorephi=no thmax=180.0 thmi
n=0.0 table="EXPOSURE" chatter=3 clobber=yes debug=no mode="ql"
2024-09-25 13:59:19 INFO    GTBinnedAnalysis.run_gtapp(): This is gtexpcube2 vers
ion HEAD
2024-09-25 13:59:19 INFO    GTBinnedAnalysis.run_gtapp(): Using evtype=3 (i.e., F
RONT/BACK irfs)
2024-09-25 13:59:19 INFO    GTBinnedAnalysis.run_gtapp(): ResponseFunctions::loa
d: IRF used: P8R2_SOURCE_V6
2024-09-25 13:59:19 INFO    GTBinnedAnalysis.run_gtapp(): event_types:  0  1
2024-09-25 13:59:20 INFO    GTBinnedAnalysis.run_gtapp(): Computing binned exposu
re map...................!
2024-09-25 13:59:20 INFO    GTBinnedAnalysis.run_gtapp(): Finished gtexpcube2. Ex
ecution time: 0.83 s
WARNING: FITSFixedWarning: RADECSYS= 'FK5 '
the RADECSYS keyword is deprecated, use RADESYSa. [astropy.wcs.wcs]
WARNING: FITSFixedWarning: 'datfix' made the change 'Set DATEREF to '2001-01-01T0
0:01:04.184' from MJDREF.
Set MJD-OBS to 54704.000000 from DATE-OBS.
Set MJD-END to 54888.999988 from DATE-END'. [astropy.wcs.wcs]
2024-09-25 13:59:20 INFO    GTBinnedAnalysis.run_gtapp(): Running gtsrcmaps.
2024-09-25 13:59:20 INFO    GTBinnedAnalysis.run_gtapp(): time -p gtsrcmaps scfil
e=./PKS2155_data/L211025154921445A2D9929_SC00.fits sctable="SC_DATA" expcube=/hom
e/chen/PHYS2015A_fermipy_working/notebooks/PKS2155_data/ltcube_00.fits cmap=/hom
e/chen/PHYS2015A_fermipy_working/notebooks/PKS2155_data/ccube_00.fits srcmdl=/hom
e/chen/PHYS2015A_fermipy_working/notebooks/PKS2155_data/srcmdl_00.xml bexpmap=/ho
me/chen/PHYS2015A_fermipy_working/notebooks/PKS2155_data/bexpmap_00.fits wmap=non
e outfile=/home/chen/PHYS2015A_fermipy_working/notebooks/PKS2155_data/srcmap_00.f
its irfs="P8R2_SOURCE_V6" evtype=3 convol=yes resample=yes rfactor=2 minbinsz=0.0
5 ptsrc=yes psfcorr=yes emapbnds=no edisp_bins=-1 copyall=no chatter=3 clobber=ye
s debug=no gui=no mode="ql"
2024-09-25 13:59:20 INFO    GTBinnedAnalysis.run_gtapp(): This is gtsrcmaps versi
on HEAD
2024-09-25 13:59:20 INFO    GTBinnedAnalysis.run_gtapp(): Using evtype=3 (i.e., F
RONT/BACK irfs)
2024-09-25 13:59:20 INFO    GTBinnedAnalysis.run_gtapp(): ResponseFunctions::loa
d: IRF used: P8R2_SOURCE_V6
2024-09-25 13:59:20 INFO    GTBinnedAnalysis.run_gtapp(): event_types:  0  1
2024-09-25 13:59:22 INFO    GTBinnedAnalysis.run_gtapp(): Creating source named 3
FGL J2158.8-3013
2024-09-25 13:59:22 INFO    GTBinnedAnalysis.run_gtapp(): Creating source named 3
FGL J2151.8-3025
2024-09-25 13:59:22 INFO    GTBinnedAnalysis.run_gtapp(): Creating source named 3
FGL J2159.2-2841
2024-09-25 13:59:22 INFO    GTBinnedAnalysis.run_gtapp(): Creating source named 3
FGL J2151.6-2744
2024-09-25 13:59:22 INFO    GTBinnedAnalysis.run_gtapp(): Creating source named 3
FGL J2144.9-3356
2024-09-25 13:59:22 INFO    GTBinnedAnalysis.run_gtapp(): Creating source named 3
FGL J2213.1-2532
2024-09-25 13:59:22 INFO    GTBinnedAnalysis.run_gtapp(): Creating source named 3
FGL J2142.2-2546
2024-09-25 13:59:22 INFO    GTBinnedAnalysis.run_gtapp(): Creating source named 3
FGL J2222.3-3500
2024-09-25 13:59:22 INFO    GTBinnedAnalysis.run_gtapp(): Creating source named i
sodiff
2024-09-25 13:59:22 INFO    GTBinnedAnalysis.run_gtapp(): Creating source named g
aldiff
2024-09-25 13:59:22 INFO    GTBinnedAnalysis.run_gtapp(): Generating SourceMap fo
r 3FGL J2142.2-2546 21...................!
```

```
2024-09-25 13:59:22 INFO    GTBinnedAnalysis.run_gtapp(): Generating SourceMap fo
r 3FGL J2144.9-3356 21....................!
2024-09-25 13:59:22 INFO    GTBinnedAnalysis.run_gtapp(): Generating SourceMap fo
r 3FGL J2151.6-2744 21....................!
2024-09-25 13:59:23 INFO    GTBinnedAnalysis.run_gtapp(): Generating SourceMap fo
r 3FGL J2151.8-3025 21....................!
2024-09-25 13:59:23 INFO    GTBinnedAnalysis.run_gtapp(): Generating SourceMap fo
r 3FGL J2158.8-3013 21....................!
2024-09-25 13:59:23 INFO    GTBinnedAnalysis.run_gtapp(): Generating SourceMap fo
r 3FGL J2159.2-2841 21....................!
2024-09-25 13:59:24 INFO    GTBinnedAnalysis.run_gtapp(): Generating SourceMap fo
r 3FGL J2213.1-2532 21....................!
2024-09-25 13:59:24 INFO    GTBinnedAnalysis.run_gtapp(): Generating SourceMap fo
r 3FGL J2222.3-3500 21....................!
2024-09-25 13:59:41 INFO    GTBinnedAnalysis.run_gtapp(): Generating SourceMap fo
r galdiff 21....................!
2024-09-25 13:59:54 INFO    GTBinnedAnalysis.run_gtapp(): Generating SourceMap fo
r isodiff 21....................!
2024-09-25 13:59:54 INFO    GTBinnedAnalysis.run_gtapp(): Finished gtsrcmaps. Exe
cution time: 33.85 s
2024-09-25 13:59:54 INFO    GTBinnedAnalysis.setup(): Finished setup for componen
t 00
2024-09-25 13:59:54 INFO    GTBinnedAnalysis._create_binned_analysis(): Creating
BinnedAnalysis for component 00.
2024-09-25 13:59:56 INFO    GTAnalysis.setup(): Initializing source properties
2024-09-25 13:59:58 INFO    GTAnalysis.setup(): Finished setup.
```

In [28]:
```
gta.print_model()
```

```
2024-09-25 14:00:17 INFO    GTAnalysis.print_model():
sourcename             offset    norm  eflux     index        ts        npred free
--------------------------------------------------------------------------------
3FGL J2158.8-3013       0.000   2.204  0.000138   1.85       nan        467.3
3FGL J2151.8-3025       1.539   2.031   9.5e-07   4.40       nan         13.5
3FGL J2159.2-2841       1.542   0.393  2.72e-06   1.94       nan          9.0
3FGL J2151.6-2744       2.948   1.013  8.19e-07   2.51       nan          6.2
3FGL J2144.9-3356       4.743   2.950  4.25e-06   2.30       nan         24.5
3FGL J2213.1-2532       5.650   0.351  1.32e-06   2.55       nan         10.1
3FGL J2142.2-2546       5.760   0.364  1.68e-06   2.23       nan          9.0
3FGL J2222.3-3500       6.878   0.394  7.72e-07   2.37       nan          4.6
isodiff                   ---   1.000    0.0301   2.12       nan        472.1
galdiff                   ---   1.000      0.13   0.00       nan        878.3
```

In [29]:
```
gta.free_sources()
gta.fit()
```

```
2024-09-25 14:00:18 INFO    GTAnalysis.free_source(): Freeing parameters for 3FGL
J2158.8-3013     : ['norm', 'alpha', 'beta']
2024-09-25 14:00:18 INFO    GTAnalysis.free_source(): Freeing parameters for 3FGL
J2151.8-3025     : ['norm', 'alpha', 'beta']
2024-09-25 14:00:18 INFO    GTAnalysis.free_source(): Freeing parameters for 3FGL
J2159.2-2841     : ['Prefactor', 'Index']
2024-09-25 14:00:18 INFO    GTAnalysis.free_source(): Freeing parameters for 3FGL
J2151.6-2744     : ['Prefactor', 'Index']
2024-09-25 14:00:18 INFO    GTAnalysis.free_source(): Freeing parameters for 3FGL
J2144.9-3356     : ['Prefactor', 'Index']
2024-09-25 14:00:18 INFO    GTAnalysis.free_source(): Freeing parameters for 3FGL
J2213.1-2532     : ['Prefactor', 'Index']
2024-09-25 14:00:18 INFO    GTAnalysis.free_source(): Freeing parameters for 3FGL
J2142.2-2546     : ['Prefactor', 'Index']
2024-09-25 14:00:18 INFO    GTAnalysis.free_source(): Freeing parameters for 3FGL
J2222.3-3500     : ['Prefactor', 'Index']
2024-09-25 14:00:18 INFO    GTAnalysis.free_source(): Freeing parameters for isod
iff               : ['Normalization']
2024-09-25 14:00:18 INFO    GTAnalysis.free_source(): Freeing parameters for gald
iff               : ['Prefactor', 'Index']
2024-09-25 14:00:18 INFO    GTAnalysis.fit(): Starting fit.
2024-09-25 14:00:18 INFO    GTAnalysis.fit(): Fit returned successfully. Quality:
3 Status:   0
2024-09-25 14:00:18 INFO    GTAnalysis.fit(): LogLike:    -8721.929 DeltaLogLike:
24.835
Drm_Cache::update Measured counts < 0 3FGL J2151.8-3025 19 -2.86633e-14 4.66044e-
13
2.65931 1.80298 1.05619 0.536945 0.237817 0.0924885 0.0319272 0.00991039 0.002702
09 0.000648556 0.00013762 2.60488e-05 4.41123e-06 6.54526e-07 8.47181e-08 9.65071
e-09 9.69929e-10 8.59364e-11 6.7331e-12 4.66044e-13
```

```
Out[29]: {'fit_quality': 3,
          'fit_status': 0,
          'fit_success': True,
          'dloglike': 24.83531423709792,
          'edm': 0.00014450939796931605,
          'loglike': -8721.928683972012,
          'covariance': array([[ 1.64517549e-02,  9.98976778e-02,  2.40325744e-03,
                   1.78549944e-05,  6.47085767e-04, -2.67066184e-04,
                   8.30831196e-08, -1.14307600e-05,  2.42433636e-05,
                   1.46352663e-04,  1.36991604e-05, -2.03836407e-07,
                   1.46957003e-04,  1.48324427e-04,  1.57456561e-04,
                   1.85504828e-04,  1.59325310e-04,  7.30971390e-05,
                  -5.86941887e-04,  1.55406169e-04,  5.12244240e-04],
                 [ 9.98976778e-02,  2.21893970e+00,  1.97025691e-02,
                   1.28024601e-03,  4.73259364e-03,  4.91550635e-04,
                   6.16536895e-07, -4.34650389e-05,  3.16358628e-04,
                   1.09160903e-03,  3.43637380e-04, -5.61991924e-05,
                   3.82915910e-04,  7.32792720e-04,  9.92266925e-04,
                   1.65570110e-03,  9.05920776e-04,  1.29155169e-03,
                  -3.71694712e-03, -2.17909316e-03,  1.98008159e-03],
                 [ 2.40325744e-03,  1.97025691e-02,  1.22748594e+01,
                   1.12833651e+00,  9.92676086e-02,  1.80732101e-02,
                   2.59688596e-06, -2.67872902e-04,  1.14009801e-03,
                   6.93353464e-03,  2.11476691e-03, -3.41296794e-04,
                   3.51621637e-03,  5.89362141e-03,  8.49116818e-03,
                   1.25705259e-02,  1.01716131e-02,  1.04228032e-02,
                  -2.16857443e-02, -1.83438022e-02, -2.44359378e-03],
                 [ 1.78549944e-05,  1.28024601e-03,  1.12833651e+00,
                   1.45250169e-01,  6.82301014e-03,  1.84571319e-03,
                   1.91534689e-07, -1.12265633e-05,  1.10913125e-04,
                   4.97916061e-04,  1.91458825e-04, -3.42184097e-05,
                   1.44234400e-04,  3.34513900e-04,  6.10463874e-04,
                   8.84139159e-04,  9.07790532e-04,  8.87637687e-04,
                  -8.13809046e-04, -1.95603981e-03, -1.89187594e-03],
                 [ 6.47085767e-04,  4.73259364e-03,  9.92676086e-02,
                   6.82301014e-03,  7.75229130e+00,  3.26029307e+00,
                  -3.73327093e-07, -1.41316916e-04, -5.53551840e-04,
                  -3.46710705e-03, -1.36715232e-03,  2.73002273e-04,
                  -5.01836473e-03, -6.76742395e-03,  6.18797653e-03,
                   7.89858983e-03,  9.48785074e-03,  7.26126343e-03,
                  -1.04056149e-02, -1.20010946e-02, -1.30267964e-02],
                 [-2.67066184e-04,  4.91550635e-04,  1.80732101e-02,
                   1.84571319e-03,  3.26029307e+00,  2.34778476e+00,
                  -3.20690134e-07, -1.33292952e-05, -2.76793642e-04,
                  -1.04776980e-03, -4.25909577e-04,  8.79270701e-05,
                  -8.52735707e-04, -1.22241019e-03,  1.13012980e-03,
                   1.46819332e-03,  2.10231910e-03,  1.75445799e-03,
                  -3.85148045e-04, -4.02050509e-03, -6.14904609e-03],
                 [ 8.30831196e-08,  6.16536895e-07,  2.59688596e-06,
                   1.91534689e-07, -3.73327093e-07, -3.20690134e-07,
                   1.05425153e-07,  1.72528515e-06,  7.80050918e-05,
                  -4.19701776e-06, -1.41135947e-06,  2.57194235e-07,
                   8.16284844e-08,  1.19095372e-07,  3.35071206e-07,
                   4.35757711e-07,  4.95465016e-07,  3.90417183e-07,
                  -5.22182121e-07, -6.83447220e-07, -7.41297586e-07],
                 [-1.14307600e-05, -4.34650389e-05, -2.67872902e-04,
                  -1.12265633e-05, -1.41316916e-04, -1.33292952e-05,
                   1.72528515e-06,  5.22391048e-04,  3.98184311e-04,
                   2.19572908e-04,  4.81145863e-05, -6.55453309e-06,
                  -1.39313102e-05, -1.79224319e-05, -2.04250055e-05,
```

```
 -2.71563340e-05, -2.33174069e-05, -1.80511452e-05,
  6.04273362e-05,  1.52185914e-05, -2.22851748e-05],
[ 2.42433636e-05,  3.16358628e-04,  1.14009801e-03,
  1.10913125e-04, -5.53551840e-04, -2.76793642e-04,
  7.80050918e-05,  3.98184311e-04,  6.94726926e-02,
 -2.41040618e-03, -8.93504554e-04,  1.69855040e-04,
  2.94191031e-05,  4.77529034e-05,  1.82200754e-04,
  2.34405693e-04,  2.93189587e-04,  2.31258068e-04,
 -1.84327681e-04, -4.59394732e-04, -6.34859601e-04],
[ 1.46352663e-04,  1.09160903e-03,  6.93353464e-03,
  4.97916061e-04, -3.46710705e-03, -1.04776980e-03,
 -4.19701776e-06,  2.19572908e-04, -2.41040618e-03,
  8.12126104e-02,  3.12843338e-02, -6.12971335e-03,
 -3.27527983e-03, -4.52287840e-03,  5.26541532e-04,
  6.91787209e-04,  7.95928974e-04,  6.41730413e-04,
 -9.82540326e-04, -1.17395281e-03, -1.10560873e-03],
[ 1.36991604e-05,  3.43637380e-04,  2.11476691e-03,
  1.91458825e-04, -1.36715232e-03, -4.25909577e-04,
 -1.41135947e-06,  4.81145863e-05, -8.93504554e-04,
  3.12843338e-02,  1.64609438e-02, -3.65102470e-03,
 -9.14946652e-04, -1.34261602e-03,  1.55159903e-04,
  2.21392485e-04,  2.38314882e-04,  2.27503417e-04,
 -2.63137748e-04, -4.99041964e-04, -4.44329654e-04],
[-2.03836407e-07, -5.61991924e-05, -3.41296794e-04,
 -3.42184097e-05,  2.73002273e-04,  8.79270701e-05,
  2.57194235e-07, -6.55453309e-06,  1.69855040e-04,
 -6.12971335e-03, -3.65102470e-03,  9.16560306e-04,
  1.53793348e-04,  2.30412564e-04, -2.63146998e-05,
 -3.70051849e-05, -4.32046310e-05, -4.04247290e-05,
  3.34861483e-05,  9.35797955e-05,  1.01267665e-04],
[ 1.46957003e-04,  3.82915910e-04,  3.51621637e-03,
  1.44234400e-04, -5.01836473e-03, -8.52735707e-04,
  8.16284844e-08, -1.39313102e-05,  2.94191031e-05,
 -3.27527983e-03, -9.14946652e-04,  1.53793348e-04,
  5.37327706e-02,  4.95950148e-02,  2.41286481e-04,
  2.30764487e-04,  4.40493829e-04,  1.98373530e-04,
 -3.27144019e-04, -1.12862682e-04, -6.07247904e-04],
[ 1.48324427e-04,  7.32792720e-04,  5.89362141e-03,
  3.34513900e-04, -6.76742395e-03, -1.22241019e-03,
  1.19095372e-07, -1.79224319e-05,  4.77529034e-05,
 -4.52287840e-03, -1.34261602e-03,  2.30412564e-04,
  4.95950148e-02,  1.30650369e-01,  3.15251368e-04,
  4.23592099e-04,  4.48251204e-04,  3.67962340e-04,
 -8.85560884e-04, -4.90208104e-04, -4.28686603e-05],
[ 1.57456561e-04,  9.92266925e-04,  8.49116818e-03,
  6.10463874e-04,  6.18797653e-03,  1.13012980e-03,
  3.35071206e-07, -2.04250055e-05,  1.82200754e-04,
  5.26541532e-04,  1.55159903e-04, -2.63146998e-05,
  2.41286481e-04,  3.15251368e-04,  5.67803903e-02,
  1.01654156e-01,  9.36683131e-04,  6.38039096e-04,
 -6.01626259e-04, -1.07767798e-03, -1.81623360e-03],
[ 1.85504828e-04,  1.65570110e-03,  1.25705259e-02,
  8.84139159e-04,  7.89858983e-03,  1.46819332e-03,
  4.35757711e-07, -2.71563340e-05,  2.34405693e-04,
  6.91787209e-04,  2.21392485e-04, -3.70051849e-05,
  2.30764487e-04,  4.23592099e-04,  1.01654156e-01,
  2.41832259e-01,  8.09867751e-04,  8.83276182e-04,
 -1.87372933e-03, -1.57465870e-03, -3.16639047e-05],
[ 1.59325310e-04,  9.05920776e-04,  1.01716131e-02,
  9.07790532e-04,  9.48785074e-03,  2.10231910e-03,
```

```
         4.95465016e-07, -2.33174069e-05,  2.93189587e-04,
         7.95928974e-04,  2.38314882e-04, -4.32046310e-05,
         4.40493829e-04,  4.48251204e-04,  9.36683131e-04,
         8.09867751e-04,  1.98167813e-01,  2.32213772e-01,
         1.03260424e-03, -1.77613397e-03, -6.86397574e-03],
       [ 7.30971390e-05,  1.29155169e-03,  1.04228032e-02,
         8.87637687e-04,  7.26126343e-03,  1.75445799e-03,
         3.90417183e-07, -1.80511452e-05,  2.31258068e-04,
         6.41730413e-04,  2.27503417e-04, -4.04247290e-05,
         1.98373530e-04,  3.67962340e-04,  6.38039096e-04,
         8.83276182e-04,  2.32213772e-01,  5.01751534e-01,
        -8.63717729e-04, -1.76733075e-03, -1.84147294e-03],
       [-5.86941887e-04, -3.71694712e-03, -2.16857443e-02,
        -8.13809046e-04, -1.04056149e-02, -3.85148045e-04,
        -5.22182121e-07,  6.04273362e-05, -1.84327681e-04,
        -9.82540326e-04, -2.63137748e-04,  3.34861483e-05,
        -3.27144019e-04, -8.85560884e-04, -6.01626259e-04,
        -1.87372933e-03,  1.03260424e-03, -8.63717729e-04,
         8.34218461e-03,  7.24361963e-04, -1.33668884e-02],
       [ 1.55406169e-04, -2.17909316e-03, -1.83438022e-02,
        -1.95603981e-03, -1.20010946e-02, -4.02050509e-03,
        -6.83447220e-07,  1.52185914e-05, -4.59394732e-04,
        -1.17395281e-03, -4.99041964e-04,  9.35797955e-05,
        -1.12862682e-04, -4.90208104e-04, -1.07767798e-03,
        -1.57465870e-03, -1.77613397e-03, -1.76733075e-03,
         7.24361963e-04,  4.26818889e-03,  5.05872587e-03],
       [ 5.12244240e-04,  1.98008159e-03, -2.44359378e-03,
        -1.89187594e-03, -1.30267964e-02, -6.14904609e-03,
        -7.41297586e-07, -2.22851748e-05, -6.34859601e-04,
        -1.10560873e-03, -4.44329654e-04,  1.01267665e-04,
        -6.07247904e-04, -4.28686603e-05, -1.81623360e-03,
        -3.16639047e-05, -6.86397574e-03, -1.84147294e-03,
        -1.33668884e-02,  5.05872587e-03,  3.61688382e-02]]),
 'correlation': array([[ 1.00000000e+00,  5.22849295e-01,  5.34793249e-03,
         3.65254251e-04,  1.81192697e-03, -1.35888794e-03,
         1.99496169e-03, -3.89916063e-03,  7.17099952e-04,
         4.00389592e-03,  8.32454430e-04, -5.24922414e-05,
         4.94270344e-03,  3.19927258e-03,  5.15175807e-03,
         2.94097942e-03,  2.79037195e-03,  8.04544201e-04,
        -5.01013110e-02,  1.85455759e-02,  2.09992454e-02],
       [ 5.22849295e-01,  1.00000000e+00,  3.77521636e-03,
         2.25508211e-03,  1.14106773e-03,  2.15360659e-04,
         1.27471824e-03, -1.27664297e-03,  8.05749171e-04,
         2.57147693e-03,  1.79804387e-03, -1.24616859e-03,
         1.10894765e-03,  1.36098417e-03,  2.79548122e-03,
         2.26022706e-03,  1.36615794e-03,  1.22403740e-03,
        -2.73195621e-02, -2.23913991e-02,  6.98945086e-03],
       [ 5.34793249e-03,  3.77521636e-03,  1.00000000e+00,
         8.45029861e-01,  1.01761677e-02,  3.36664942e-03,
         2.28282435e-03, -3.34520438e-03,  1.23460167e-03,
         6.94439942e-03,  4.70464521e-03, -3.21768424e-03,
         4.32959938e-03,  4.65391718e-03,  1.01709143e-02,
         7.29605445e-03,  6.52176301e-03,  4.19983204e-03,
        -6.77682346e-02, -8.01418345e-02, -3.66735936e-03],
       [ 3.65254251e-04,  2.25508211e-03,  8.45029861e-01,
         1.00000000e+00,  6.42987681e-03,  3.16065253e-03,
         1.54780768e-03, -1.28881518e-03,  1.10412325e-03,
         4.58443772e-03,  3.91552230e-03, -2.96565739e-03,
         1.63264181e-03,  2.42828985e-03,  6.72206115e-03,
         4.71742699e-03,  5.35070205e-03,  3.28800944e-03,
```

```
 -2.33789118e-02, -7.85592903e-02, -2.61015954e-02],
[ 1.81192697e-03,  1.14106773e-03,  1.01761677e-02,
  6.42987681e-03,  1.00000000e+00,  7.64209548e-01,
 -4.12954638e-04, -2.22065579e-03, -7.54286250e-04,
 -4.36958784e-03, -3.82714130e-03,  3.23869983e-03,
 -7.77548635e-03, -6.72439197e-03,  9.32683791e-03,
  5.76869269e-03,  7.65484222e-03,  3.68173729e-03,
 -4.09178501e-02, -6.59756991e-02, -2.46011441e-02],
[-1.35888794e-03,  2.15360659e-04,  3.36664942e-03,
  3.16065253e-03,  7.64209548e-01,  1.00000000e+00,
 -6.44591153e-04, -3.80609991e-04, -6.85361816e-04,
 -2.39952346e-03, -2.16651101e-03,  1.89545265e-03,
 -2.40085367e-03, -2.20715331e-03,  3.09528042e-03,
  1.94848465e-03,  3.08214408e-03,  1.61647681e-03,
 -2.75206483e-03, -4.01633251e-02, -2.11014029e-02],
[ 1.99496169e-03,  1.27471824e-03,  2.28282435e-03,
  1.54780768e-03, -4.12954638e-04, -6.44591153e-04,
  1.00000000e+00,  2.32482840e-01,  9.11472915e-01,
 -4.53583192e-02, -3.38795910e-02,  2.61642679e-02,
  1.08455104e-03,  1.01476959e-03,  4.33077682e-03,
  2.72907735e-03,  3.42786980e-03,  1.69751042e-03,
 -1.76079980e-02, -3.22189250e-02, -1.20047519e-02],
[-3.89916063e-03, -1.27664297e-03, -3.34520438e-03,
 -1.28881518e-03, -2.22065579e-03, -3.80609991e-04,
  2.32482840e-01,  1.00000000e+00,  6.60966146e-02,
  3.37108075e-02,  1.64078443e-02, -9.47247297e-03,
 -2.62950790e-03, -2.16942057e-03, -3.75029242e-03,
 -2.41610699e-03, -2.29174256e-03, -1.11496836e-03,
  2.89464753e-02,  1.01918912e-02, -5.12685345e-03],
[ 7.17099952e-04,  8.05749171e-04,  1.23460167e-03,
  1.10412325e-03, -7.54286250e-04, -6.85361816e-04,
  9.11472915e-01,  6.60966146e-02,  1.00000000e+00,
 -3.20901267e-02, -2.64217963e-02,  2.12858428e-02,
  4.81507155e-04,  5.01230390e-04,  2.90097453e-03,
  1.80843945e-03,  2.49876214e-03,  1.23864106e-03,
 -7.65673462e-03, -2.66782372e-02, -1.26649461e-02],
[ 4.00389592e-03,  2.57147693e-03,  6.94439942e-03,
  4.58443772e-03, -4.36958784e-03, -2.39952346e-03,
 -4.53583192e-02,  3.37108075e-02, -3.20901267e-02,
  1.00000000e+00,  8.55633561e-01, -7.10473539e-01,
 -4.95812028e-02, -4.39084247e-02,  7.75392705e-03,
  4.93632422e-03,  6.27402176e-03,  3.17904395e-03,
 -3.77484176e-02, -6.30546454e-02, -2.03996372e-02],
[ 8.32454430e-04,  1.79804387e-03,  4.70464521e-03,
  3.91552230e-03, -3.82714130e-03, -2.16651101e-03,
 -3.38795910e-02,  1.64078443e-02, -2.64217963e-02,
  8.55633561e-01,  1.00000000e+00, -9.39954611e-01,
 -3.07644241e-02, -2.89513712e-02,  5.07519749e-03,
  3.50895980e-03,  4.17260468e-03,  2.50331905e-03,
 -2.24551463e-02, -5.95371286e-02, -1.82100297e-02],
[-5.24922414e-05, -1.24616859e-03, -3.21768424e-03,
 -2.96565739e-03,  3.23869983e-03,  1.89545265e-03,
  2.61642679e-02, -9.47247297e-03,  2.12858428e-02,
 -7.10473539e-01, -9.39954611e-01,  1.00000000e+00,
  2.19147950e-02,  2.10557305e-02, -3.64769662e-03,
 -2.48556394e-03, -3.20577634e-03, -1.88504725e-03,
  1.21100176e-02,  4.73129202e-02,  1.75882647e-02],
[ 4.94270344e-03,  1.10894765e-03,  4.32959938e-03,
  1.63264181e-03, -7.77548635e-03, -2.40085367e-03,
  1.08455104e-03, -2.62950790e-03,  4.81507155e-04,
```

```
 -4.95812028e-02, -3.07644241e-02,  2.19147950e-02,
  1.00000000e+00,  5.91920097e-01,  4.36831739e-03,
  2.02438095e-03,  4.26877882e-03,  1.20814692e-03,
 -1.54518139e-02, -7.45261924e-03, -1.37746042e-02],
[ 3.19927258e-03,  1.36098417e-03,  4.65391718e-03,
  2.42828985e-03, -6.72439197e-03, -2.20715331e-03,
  1.01476959e-03, -2.16942057e-03,  5.01230390e-04,
 -4.39084247e-02, -2.89513712e-02,  2.10557305e-02,
  5.91920097e-01,  1.00000000e+00,  3.66018023e-03,
  2.38306240e-03,  2.78579776e-03,  1.43715529e-03,
 -2.68239793e-02, -2.07588528e-02, -6.23616088e-04],
[ 5.15175807e-03,  2.79548122e-03,  1.01709143e-02,
  6.72206115e-03,  9.32683791e-03,  3.09528042e-03,
  4.33077682e-03, -3.75029242e-03,  2.90097453e-03,
  7.75392705e-03,  5.07519749e-03, -3.64769662e-03,
  4.36831739e-03,  3.66018023e-03,  1.00000000e+00,
  8.67498707e-01,  8.83033040e-03,  3.78010401e-03,
 -2.76431607e-02, -6.92258038e-02, -4.00779260e-02],
[ 2.94097942e-03,  2.26022706e-03,  7.29605445e-03,
  4.71742699e-03,  5.76869269e-03,  1.94848465e-03,
  2.72907735e-03, -2.41610699e-03,  1.80843945e-03,
  4.93632422e-03,  3.50895980e-03, -2.48556394e-03,
  2.02438095e-03,  2.38306240e-03,  8.67498707e-01,
  1.00000000e+00,  3.69947785e-03,  2.53568352e-03,
 -4.17166899e-02, -4.90125734e-02, -3.38563451e-04],
[ 2.79037195e-03,  1.36615794e-03,  6.52176301e-03,
  5.35070205e-03,  7.65484222e-03,  3.08214408e-03,
  3.42786980e-03, -2.29174256e-03,  2.49876214e-03,
  6.27402176e-03,  4.17260468e-03, -3.20577634e-03,
  4.26877882e-03,  2.78579776e-03,  8.83033040e-03,
  3.69947785e-03,  1.00000000e+00,  7.36422520e-01,
  2.53967092e-02, -6.10713257e-02, -8.10759231e-02],
[ 8.04544201e-04,  1.22403740e-03,  4.19983204e-03,
  3.28800944e-03,  3.68173729e-03,  1.61647681e-03,
  1.69751042e-03, -1.11496836e-03,  1.23864106e-03,
  3.17904395e-03,  2.50331905e-03, -1.88504725e-03,
  1.20814692e-03,  1.43715529e-03,  3.78010401e-03,
  2.53568352e-03,  7.36422520e-01,  1.00000000e+00,
 -1.33501941e-02, -3.81901763e-02, -1.36695339e-02],
[-5.01013110e-02, -2.73195621e-02, -6.77682346e-02,
 -2.33789118e-02, -4.09178501e-02, -2.75206483e-03,
 -1.76079980e-02,  2.89464753e-02, -7.65673462e-03,
 -3.77484176e-02, -2.24551463e-02,  1.21100176e-02,
 -1.54518139e-02, -2.68239793e-02, -2.76431607e-02,
 -4.17166899e-02,  2.53967092e-02, -1.33501941e-02,
  1.00000000e+00,  1.21393078e-01, -7.69525727e-01],
[ 1.85455759e-02, -2.23913991e-02, -8.01418345e-02,
 -7.85592903e-02, -6.59756991e-02, -4.01633251e-02,
 -3.22189250e-02,  1.01918912e-02, -2.66782372e-02,
 -6.30546454e-02, -5.95371286e-02,  4.73129202e-02,
 -7.45261924e-03, -2.07588528e-02, -6.92258038e-02,
 -4.90125734e-02, -6.10713257e-02, -3.81901763e-02,
  1.21393078e-01,  1.00000000e+00,  4.07147761e-01],
[ 2.09992454e-02,  6.98945086e-03, -3.66735936e-03,
 -2.61015954e-02, -2.46011441e-02, -2.11014029e-02,
 -1.20047519e-02, -5.12685345e-03, -1.26649461e-02,
 -2.03996372e-02, -1.82100297e-02,  1.75882647e-02,
 -1.37746042e-02, -6.23616088e-04, -4.00779260e-02,
 -3.38563451e-04, -8.10759231e-02, -1.36695339e-02,
 -7.69525727e-01,  4.07147761e-01,  1.00000000e+00]]),
```

```
 'values': array([ 3.79269944e-02,  1.96496062e+00,  9.38416756e+00,  2.9917870
2e+00,
        1.87946930e+00,  3.47794136e+00,  2.52660949e-04, -4.99940630e+00,
        1.83706504e+00,  2.79817824e+00,  1.81044083e+00,  2.31978913e-02,
        3.83881395e-01,  1.60820896e+00,  2.58213084e-01,  2.26450453e+00,
        4.51893995e-01,  2.28758167e+00,  9.64456874e-01, -1.15341301e-01,
        5.99638036e-01]),
 'errors': array([1.28264395e-01, 1.48961059e+00, 3.50354954e+00, 3.81117001e-0
1,
       2.78429368e+00, 1.53224827e+00, 3.24692397e-04, 2.28558756e-02,
       2.63576730e-01, 2.84978263e-01, 1.28300209e-01, 3.02747470e-02,
       2.31803302e-01, 3.61455903e-01, 2.38286362e-01, 4.91764435e-01,
       4.45160435e-01, 7.08344220e-01, 9.13355605e-02, 6.53313776e-02,
       1.90181067e-01]),
 'indices': array([ 0,  1,  3,  4,  6,  7,  9, 10, 11, 13, 14, 15, 17, 18, 20,
21, 23,
       24, 26, 27, 29]),
 'is_norm': array([ True, False,  True, False,  True, False,  True, False, Fals
e,
        True, False, False,  True, False,  True, False,  True, False,
        True, False,  True]),
 'src_names': ['3FGL J2142.2-2546',
  '3FGL J2142.2-2546',
  '3FGL J2144.9-3356',
  '3FGL J2144.9-3356',
  '3FGL J2151.6-2744',
  '3FGL J2151.6-2744',
  '3FGL J2151.8-3025',
  '3FGL J2151.8-3025',
  '3FGL J2151.8-3025',
  '3FGL J2158.8-3013',
  '3FGL J2158.8-3013',
  '3FGL J2158.8-3013',
  '3FGL J2159.2-2841',
  '3FGL J2159.2-2841',
  '3FGL J2213.1-2532',
  '3FGL J2213.1-2532',
  '3FGL J2222.3-3500',
  '3FGL J2222.3-3500',
  'galdiff',
  'galdiff',
  'isodiff'],
 'par_names': ['Prefactor',
  'Index',
  'Prefactor',
  'Index',
  'Prefactor',
  'Index',
  'norm',
  'alpha',
  'beta',
  'norm',
  'alpha',
  'beta',
  'Prefactor',
  'Index',
  'Prefactor',
  'Index',
  'Prefactor',
  'Index',
```

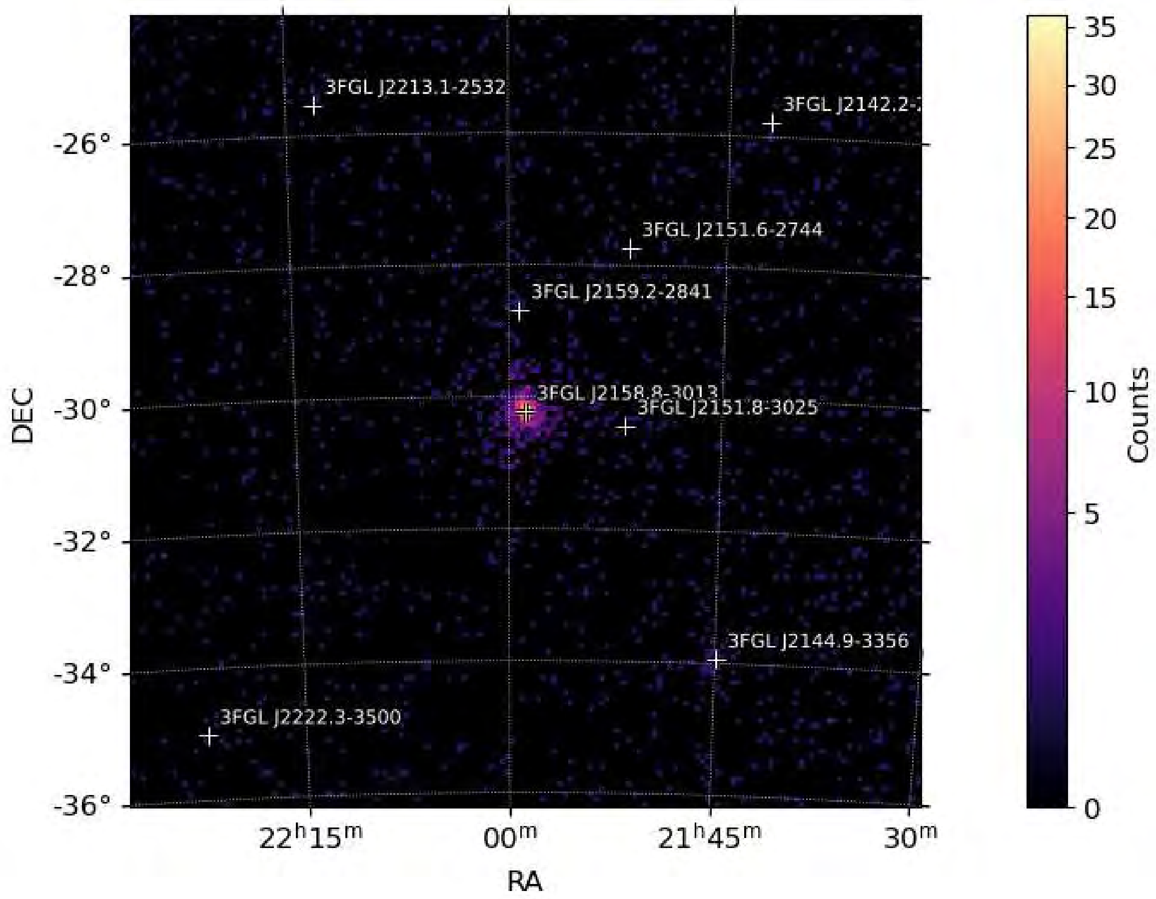
```
  'Prefactor',
  'Index',
  'Normalization'],
 'config': {'optimizer': 'MINUIT',
  'tol': 0.001,
  'max_iter': 100,
  'init_lambda': 0.0001,
  'retries': 3,
  'min_fit_quality': 2,
  'verbosity': 0,
  'covar': True,
  'reoptimize': False},
 'niter': 1}
```

Output files

The current state of the ROI can be written at any point by calling write_roi.

In [30]:
```
gta.write_roi('initial',make_plots=True,save_model_map=True)
plt.show()
```

```
2024-09-25 14:00:22 INFO    GTBinnedAnalysis.write_xml(): Writing /home/chen/PHYS
2015A_fermipy_working/notebooks/PKS2155_data/initial_00.xml...
2024-09-25 14:00:22 INFO    GTAnalysis.write_fits(): Writing /home/chen/PHYS2015A
_fermipy_working/notebooks/PKS2155_data/initial.fits...
WARNING: Format %s cannot be mapped to the accepted TDISPn keyword values.  Forma
t will not be moved into TDISPn keyword. [astropy.io.fits.column]
WARNING: Format %f cannot be mapped to the accepted TDISPn keyword values.  Forma
t will not be moved into TDISPn keyword. [astropy.io.fits.column]
WARNING: Format %s cannot be mapped to the accepted TDISPn keyword values.  Forma
t will not be moved into TDISPn keyword. [astropy.io.fits.column]
2024-09-25 14:00:27 INFO    GTBinnedAnalysis.write_model_map(): Generating model
map for component 00.
2024-09-25 14:00:28 INFO    GTAnalysis.write_roi(): Writing /home/chen/PHYS2015A_
fermipy_working/notebooks/PKS2155_data/initial.npy...
```



The output file will contain all information about the state of the ROI as calculated up to that point in the analysis including model parameters and measured source characteristics (flux, TS, NPred). An XML model file will also be saved for each analysis component.

The output file can be read with load:

In [44]:
```
gta.load_roi('initial')
```

```
2024-09-25 14:04:05 INFO    GTAnalysis.load_roi(): Loading ROI file: /home/chen/PHYS2015A_fermipy_working/notebooks/PKS2155_data/initial.npy
2024-09-25 14:04:05 INFO    GTBinnedAnalysis._create_binned_analysis(): Creating BinnedAnalysis for component 00.
2024-09-25 14:04:07 INFO    GTAnalysis.load_roi(): Finished Loading ROI
```

Using gta.print_model You have an overview of the sources and components present in the ROI.

In [45]:
```
gta.print_model()
```

```
2024-09-25 14:04:08 INFO    GTAnalysis.print_model():
sourcename              offset    norm  eflux      index        ts       npred free
-------------------------------------------------------------------------------------
3FGL J2158.8-3013        0.000   2.798  0.000176    1.87   3439.09       573.6    *
3FGL J2151.8-3025        1.539   0.000  9.83e-07    4.78      8.41        11.9    *
3FGL J2159.2-2841        1.542   0.384  5.58e-06    1.61     21.16         9.1    *
3FGL J2151.6-2744        2.948   1.879  4.05e-07    3.48      1.51         5.1    *
3FGL J2144.9-3356        4.743   9.384  4.04e-06    2.99     73.33        41.2    *
3FGL J2213.1-2532        5.650   0.258  1.94e-06    2.26     10.72        10.9    *
3FGL J2142.2-2546        5.760   0.038  3.34e-07    1.96      0.14         1.2    *
3FGL J2222.3-3500        6.878   0.452  1.06e-06    2.29      4.28         5.7    *
isodiff                    ---   0.600     0.018    2.12     38.40       283.1    *
galdiff                    ---   0.964     0.152   -0.12    817.13       921.3    *
```

Source Dictionary

The sources dictionary contains one element per source keyed to the source name. It is possible to have access to a lot of informations concerning each source of model.

In [46]:
```
print(gta.roi.sources[0].name) #NAME OF THE SOURCE
print(gta.roi[gta.roi.sources[0].name]) #NAME OF THE SOURCE
print(gta.roi[gta.roi.sources[0].name]['glon']) #Longitude OF THE SOURCE
print(gta.roi[gta.roi.sources[0].name]['glat']) #Latitude OF THE SOURCE
print(gta.roi[gta.roi.sources[0].name]['flux']) #Flux OF THE SOURCE
print(gta.roi[gta.roi.sources[0].name]['npred']) #npred OF THE SOURCE
```

```
3FGL J2158.8-3013
Name            : 3FGL J2158.8-3013
Associations    : ['3FGL J2158.8-3013', 'PKS 2155-304', '1FHL J2158.8-3013', '2FGL
J2158.8-3013', '3EG J2158-3023']
RA/DEC          :    329.720/   -30.227
GLON/GLAT       :     17.729/   -52.249
TS              : 3439.09
Npred           : 573.63
Flux            : 2.682e-08 +/- 1.21e-09
EnergyFlux      :  0.000176 +/- 1.98e-05
SpatialModel    : PointSource
SpectrumType    : LogParabola
Spectral Parameters
b'norm'         :  2.798e-11 +/-   2.85e-12
b'alpha'        :       1.81 +/-     0.1283
b'beta'         :     0.0232 +/-    0.03027
b'Eb'           :        904 +/-        nan
17.729041394470283
-52.248875333954594
2.68229777010604e-08
573.6348440388637
```

Other possible outputs are listed here fermipy/sourcedictionary

In [47]:
```
gta.free_shape(gta.roi.sources[0].name,free=False) #Free or fix the index
gta.get_free_source_params(gta.roi.sources[0].name) #Free or fix parameters for
gta.set_parameter(gta.roi.sources[0].name,par='alpha',value=2.0,scale=-1.0,bound
gta.set_parameter(gta.roi.sources[0].name,par='beta',value=0.0,scale=1.0,bounds=
```

```
2024-09-25 14:04:15 INFO    GTAnalysis.free_source(): Fixing parameters for 3FGL
J2158.8-3013     : ['alpha', 'beta']
```

You can always use gta.print_model() to have a summary of you model.

In [48]:
```
gta.print_model()
```

```
2024-09-25 14:04:18 INFO    GTAnalysis.print_model():
sourcename            offset    norm  eflux      index        ts       npred free
--------------------------------------------------------------------------------
3FGL J2158.8-3013      0.000   2.798   0.00013   2.00   3419.36       485.4    *
3FGL J2151.8-3025      1.539   0.000  9.83e-07   4.78      8.41        11.9    *
3FGL J2159.2-2841      1.542   0.384  5.58e-06   1.61     21.16         9.1    *
3FGL J2151.6-2744      2.948   1.879  4.05e-07   3.48      1.51         5.1    *
3FGL J2144.9-3356      4.743   9.384  4.04e-06   2.99     73.33        41.2    *
3FGL J2213.1-2532      5.650   0.258  1.94e-06   2.26     10.72        10.9    *
3FGL J2142.2-2546      5.760   0.038  3.34e-07   1.96      0.14         1.2    *
3FGL J2222.3-3500      6.878   0.452  1.06e-06   2.29      4.28         5.7    *
isodiff                  ---   0.600     0.018   2.12     38.40       283.1    *
galdiff                  ---   0.964     0.152  -0.12    817.13       921.3    *
```

Customizing your model

The ROIModel class is responsible for managing the source and diffuse components in the ROI. Configuration of the model is controlled with the model block of YAML configuration file.

DIFFUSE AND ISOTROPIC TEMPLATES

The simplest configuration uses a single file for the galactic and isotropic diffuse components. By default the galactic diffuse and isotropic components will be named galdiff and isodiff respectively. An alias for each component will also be created with the name of the mapcube or file spectrum. For instance the galactic diffuse can be referred to as galdiff or gll_iem_v06 in the following example.

model: src_roiwidth : 10.0 galdiff : '$FERMI_DIFFUSE_DIR/gll_iem_v06.fits'isodiff:'$
FERMI_DIFFUSE_DIR/isotropic_source_4years_P8V3.txt' catalogs : ['gll_psc_v14.fit']

To define two or more galactic diffuse components you can optionally define the galdiff and isodiff parameters as lists. A separate component will be generated for each element in the list with the name galdiffXX or isodiffXX where XX is an integer position in the list.

model: galdiff : - '$FERMI_DIFFUSE_DIR/diffuse_component0.fits' - '$FERMI_DIFFUSE_DIR/diffuse_component1.fits'

SOURCE COMPONENT

The list of sources for inclusion in the ROI model is set by defining a list of catalogs with the catalogs parameter. Catalog files can be in either XML or FITS format. Sources from the catalogs in this list that satisfy either the src_roiwidth or src_radius selections are added to the ROI model. If a source is defined in multiple catalogs the source definition from the last file in the catalogs list takes precedence.

model: src_radius: 5.0 src_roiwidth: 10.0 catalogs : - 'gll_psc_v16.fit' - 'extra_sources.xml'

Individual sources can also be defined within the configuration file with the sources parameter. This parameter contains a list of dictionaries that defines the spatial and spectral parameters of each source. The keys of the source dictionary map to the spectral and spatial source properties as they would be defined in the XML model file.

model: sources : - { name: 'SourceA', glon : 120.0, glat : -3.0, SpectrumType : 'PowerLaw', Index : 2.0, Scale : 1000, Prefactor : !!float 1e-11, SpatialModel: 'PointSource' } - { name: 'SourceB', glon : 122.0, glat : -3.0, SpectrumType : 'LogParabola', norm : !!float 1E-11, Scale : 1000, beta : 0.0, SpatialModel: 'PointSource' } model: sources : - { name: 'PointSource', glon : 120.0, glat : 0.0, SpectrumType : 'PowerLaw', Index : 2.0, Scale : 1000, Prefactor : !!float 1e-11, SpatialModel: 'PointSource' } - { name: 'DiskSource', glon : 120.0, glat : 0.0, SpectrumType : 'PowerLaw', Index : 2.0, Scale : 1000, Prefactor : !!float 1e-11, SpatialModel: 'RadialDisk', SpatialWidth: 1.0 } - { name: 'GaussSource', glon : 120.0, glat : 0.0, SpectrumType : 'PowerLaw', Index : 2.0, Scale : 1000, Prefactor : !!float 1e-11, SpatialModel: 'RadialGaussian', SpatialWidth: 1.0 } - { name: 'MapSource', glon : 120.0, glat : 0.0, SpectrumType : 'PowerLaw', Index : 2.0, Scale : 1000, Prefactor : !!float 1e-11, SpatialModel: 'SpatialTemplate', Spatial_Filename : 'template.fits' }

Or you can do it while you are running your script with:

In [49]:
```
gta.delete_source(gta.roi.sources[0].name)
glon0 = gta.config['selection']['glon']
glat0 = gta.config['selection']['glat']
gta.add_source('PKS2155-304', dict(glon=glon0, glat=glat0, Index=dict(value=-2.4
gta.print_model()
```

```
2024-09-25 14:04:33 INFO    GTAnalysis.delete_source(): Deleting source 3FGL J215
8.8-3013
2024-09-25 14:04:33 INFO    GTAnalysis.add_source(): Adding source PKS2155-304
2024-09-25 14:04:33 INFO    GTAnalysis.print_model():
sourcename              offset    norm  eflux     index        ts       npred free
--------------------------------------------------------------------------------
PKS2155-304              0.000   1.000  2.24e-07   2.40       nan         1.6    *
3FGL J2151.8-3025        1.539   0.000  9.83e-07   4.78      8.41        11.9    *
3FGL J2159.2-2841        1.542   0.384  5.58e-06   1.61     21.16         9.1    *
3FGL J2151.6-2744        2.948   1.879  4.05e-07   3.48      1.51         5.1    *
3FGL J2144.9-3356        4.743   9.384  4.04e-06   2.99     73.33        41.2    *
3FGL J2213.1-2532        5.650   0.258  1.94e-06   2.26     10.72        10.9    *
3FGL J2142.2-2546        5.760   0.038  3.34e-07   1.96      0.14         1.2    *
3FGL J2222.3-3500        6.878   0.452  1.06e-06   2.29      4.28         5.7    *
isodiff                    ---   0.600     0.018   2.12     38.40       283.1    *
galdiff                    ---   0.964     0.152  -0.12    817.13       921.3    *
```

All sources have nan because we have not done yet a fit do the ROI. Moreover in the model above all sources are fixed. In order to free the parameters of the source it's enough to make gta.free_sources()

In [50]:
```
gta.free_sources()
gta.print_model()
```

```
2024-09-25 14:04:34 INFO    GTAnalysis.free_source(): Freeing parameters for PKS2
155-304          : ['Index']
2024-09-25 14:04:34 INFO    GTAnalysis.print_model():
sourcename             offset   norm  eflux     index        ts       npred  free
--------------------------------------------------------------------------------
PKS2155-304             0.000   1.000  2.24e-07   2.40       nan         1.6    *
3FGL J2151.8-3025       1.539   0.000  9.83e-07   4.78      8.41        11.9    *
3FGL J2159.2-2841       1.542   0.384  5.58e-06   1.61     21.16         9.1    *
3FGL J2151.6-2744       2.948   1.879  4.05e-07   3.48      1.51         5.1    *
3FGL J2144.9-3356       4.743   9.384  4.04e-06   2.99     73.33        41.2    *
3FGL J2213.1-2532       5.650   0.258  1.94e-06   2.26     10.72        10.9    *
3FGL J2142.2-2546       5.760   0.038  3.34e-07   1.96      0.14         1.2    *
3FGL J2222.3-3500       6.878   0.452  1.06e-06   2.29      4.28         5.7    *
isodiff                   ---   0.600     0.018   2.12     38.40       283.1    *
galdiff                   ---   0.964     0.152  -0.12    817.13       921.3    *
```

It is also possible to free only the sources that are at a certain angular distance from a source. For example below we free the sources that are 2 degrees away from 3FGL J0322.5-3721:

In [51]:
```
gta.free_sources(free=False)
gta.free_sources(skydir=gta.roi[gta.roi.sources[0].name].skydir,distance=[3.0],f
gta.print_model()
```

```
2024-09-25 14:04:40 INFO    GTAnalysis.free_source(): Fixing parameters for PKS21
55-304           : ['Prefactor', 'Index']
2024-09-25 14:04:40 INFO    GTAnalysis.free_source(): Fixing parameters for 3FGL
J2151.8-3025     : ['norm', 'alpha', 'beta']
2024-09-25 14:04:40 INFO    GTAnalysis.free_source(): Fixing parameters for 3FGL
J2159.2-2841     : ['Prefactor', 'Index']
2024-09-25 14:04:40 INFO    GTAnalysis.free_source(): Fixing parameters for 3FGL
J2151.6-2744     : ['Prefactor', 'Index']
2024-09-25 14:04:40 INFO    GTAnalysis.free_source(): Fixing parameters for 3FGL
J2144.9-3356     : ['Prefactor', 'Index']
2024-09-25 14:04:40 INFO    GTAnalysis.free_source(): Fixing parameters for 3FGL
J2213.1-2532     : ['Prefactor', 'Index']
2024-09-25 14:04:40 INFO    GTAnalysis.free_source(): Fixing parameters for 3FGL
J2142.2-2546     : ['Prefactor', 'Index']
2024-09-25 14:04:40 INFO    GTAnalysis.free_source(): Fixing parameters for 3FGL
J2222.3-3500     : ['Prefactor', 'Index']
2024-09-25 14:04:40 INFO    GTAnalysis.free_source(): Fixing parameters for isodi
ff               : ['Normalization']
2024-09-25 14:04:40 INFO    GTAnalysis.free_source(): Fixing parameters for galdi
ff               : ['Prefactor', 'Index']
2024-09-25 14:04:40 INFO    GTAnalysis.free_source(): Freeing parameters for PKS2
155-304           : ['Prefactor', 'Index']
2024-09-25 14:04:40 INFO    GTAnalysis.free_source(): Freeing parameters for 3FGL
J2151.8-3025     : ['norm', 'alpha', 'beta']
2024-09-25 14:04:40 INFO    GTAnalysis.free_source(): Freeing parameters for 3FGL
J2159.2-2841     : ['Prefactor', 'Index']
2024-09-25 14:04:40 INFO    GTAnalysis.free_source(): Freeing parameters for 3FGL
J2151.6-2744     : ['Prefactor', 'Index']
2024-09-25 14:04:40 INFO    GTAnalysis.free_source(): Freeing parameters for isod
iff               : ['Normalization']
2024-09-25 14:04:40 INFO    GTAnalysis.free_source(): Freeing parameters for gald
iff               : ['Prefactor', 'Index']
2024-09-25 14:04:40 INFO    GTAnalysis.print_model():
sourcename            offset    norm  eflux     index        ts       npred free
--------------------------------------------------------------------------------
PKS2155-304            0.000   1.000  2.24e-07   2.40       nan         1.6    *
3FGL J2151.8-3025      1.539   0.000  9.83e-07   4.78      8.41        11.9    *
3FGL J2159.2-2841      1.542   0.384  5.58e-06   1.61     21.16         9.1    *
3FGL J2151.6-2744      2.948   1.879  4.05e-07   3.48      1.51         5.1    *
3FGL J2144.9-3356      4.743   9.384  4.04e-06   2.99     73.33        41.2
3FGL J2213.1-2532      5.650   0.258  1.94e-06   2.26     10.72        10.9
3FGL J2142.2-2546      5.760   0.038  3.34e-07   1.96      0.14         1.2
3FGL J2222.3-3500      6.878   0.452  1.06e-06   2.29      4.28         5.7
isodiff                  ---   0.600     0.018   2.12     38.40       283.1    *
galdiff                  ---   0.964     0.152  -0.12    817.13       921.3    *
```

Fit Roi

Source fitting with fermipy is generally performed with the optimize and fit methods. fit is a wrapper on the pyLikelihood fit method and performs a likelihood fit of all free parameters of the model. This method can be used to manually optimize of the model by calling it after freeing one or more source parameters.

In [52]:
```
gta.print_model()
gta.free_sources(free=True)
gta.print_model()
```

```
first_fit=gta.fit()
gta.print_model()
gta.write_roi('PKS_firstfit',make_plots=True,save_model_map=True)
```

```
2024-09-25 14:04:42 INFO    GTAnalysis.print_model():
sourcename              offset    norm  eflux     index        ts        npred free
------------------------------------------------------------------------------------
PKS2155-304              0.000   1.000  2.24e-07   2.40       nan          1.6    *
3FGL J2151.8-3025        1.539   0.000  9.83e-07   4.78      8.41         11.9    *
3FGL J2159.2-2841        1.542   0.384  5.58e-06   1.61     21.16          9.1    *
3FGL J2151.6-2744        2.948   1.879  4.05e-07   3.48      1.51          5.1    *
3FGL J2144.9-3356        4.743   9.384  4.04e-06   2.99     73.33         41.2
3FGL J2213.1-2532        5.650   0.258  1.94e-06   2.26     10.72         10.9
3FGL J2142.2-2546        5.760   0.038  3.34e-07   1.96      0.14          1.2
3FGL J2222.3-3500        6.878   0.452  1.06e-06   2.29      4.28          5.7
isodiff                    ---   0.600     0.018   2.12     38.40        283.1    *
galdiff                    ---   0.964     0.152  -0.12    817.13        921.3    *

2024-09-25 14:04:42 INFO    GTAnalysis.free_source(): Freeing parameters for 3FGL
J2144.9-3356     : ['Prefactor', 'Index']
2024-09-25 14:04:42 INFO    GTAnalysis.free_source(): Freeing parameters for 3FGL
J2213.1-2532     : ['Prefactor', 'Index']
2024-09-25 14:04:42 INFO    GTAnalysis.free_source(): Freeing parameters for 3FGL
J2142.2-2546     : ['Prefactor', 'Index']
2024-09-25 14:04:42 INFO    GTAnalysis.free_source(): Freeing parameters for 3FGL
J2222.3-3500     : ['Prefactor', 'Index']
2024-09-25 14:04:42 INFO    GTAnalysis.print_model():
sourcename              offset    norm  eflux     index        ts        npred free
------------------------------------------------------------------------------------
PKS2155-304              0.000   1.000  2.24e-07   2.40       nan          1.6    *
3FGL J2151.8-3025        1.539   0.000  9.83e-07   4.78      8.41         11.9    *
3FGL J2159.2-2841        1.542   0.384  5.58e-06   1.61     21.16          9.1    *
3FGL J2151.6-2744        2.948   1.879  4.05e-07   3.48      1.51          5.1    *
3FGL J2144.9-3356        4.743   9.384  4.04e-06   2.99     73.33         41.2    *
3FGL J2213.1-2532        5.650   0.258  1.94e-06   2.26     10.72         10.9    *
3FGL J2142.2-2546        5.760   0.038  3.34e-07   1.96      0.14          1.2    *
3FGL J2222.3-3500        6.878   0.452  1.06e-06   2.29      4.28          5.7    *
isodiff                    ---   0.600     0.018   2.12     38.40        283.1    *
galdiff                    ---   0.964     0.152  -0.12    817.13        921.3    *

2024-09-25 14:04:42 INFO    GTAnalysis.fit(): Starting fit.
2024-09-25 14:04:42 INFO    GTAnalysis.fit(): Fit returned successfully. Quality:
3 Status:   0
2024-09-25 14:04:42 INFO    GTAnalysis.fit(): LogLike:    -8722.425 DeltaLogLike:
1647.900
2024-09-25 14:04:42 INFO    GTAnalysis.print_model():
sourcename              offset    norm  eflux     index        ts        npred free
------------------------------------------------------------------------------------
PKS2155-304              0.000 232.686  0.000182   1.89   3437.83        574.2    *
3FGL J2151.8-3025        1.539   0.000  9.76e-07   4.77      8.30         11.8    *
3FGL J2159.2-2841        1.542   0.381  5.61e-06   1.60     21.12          9.0    *
3FGL J2151.6-2744        2.948   1.879  4.05e-07   3.48      1.51          5.1    *
3FGL J2144.9-3356        4.743   9.416  4.04e-06   3.00     73.37         41.2    *
3FGL J2213.1-2532        5.650   0.260  1.94e-06   2.27     10.73         10.9    *
3FGL J2142.2-2546        5.760   0.038  3.18e-07   1.99      0.14          1.2    *
3FGL J2222.3-3500        6.878   0.452  1.06e-06   2.29      4.29          5.7    *
isodiff                    ---   0.598     0.018   2.12     38.17        282.4    *
galdiff                    ---   0.963     0.153  -0.12    819.19        921.6    *

2024-09-25 14:04:42 INFO    GTBinnedAnalysis.write_xml(): Writing /home/chen/PHYS
2015A_fermipy_working/notebooks/PKS2155_data/PKS_firstfit_00.xml...
2024-09-25 14:04:42 INFO    GTAnalysis.write_fits(): Writing /home/chen/PHYS2015A
_fermipy_working/notebooks/PKS2155_data/PKS_firstfit.fits...
WARNING: Format %s cannot be mapped to the accepted TDISPn keyword values.  Forma
```

```
t will not be moved into TDISPn keyword. [astropy.io.fits.column]
WARNING: Format %f cannot be mapped to the accepted TDISPn keyword values.  Forma
t will not be moved into TDISPn keyword. [astropy.io.fits.column]
WARNING: Format %s cannot be mapped to the accepted TDISPn keyword values.  Forma
t will not be moved into TDISPn keyword. [astropy.io.fits.column]
2024-09-25 14:04:47 INFO    GTBinnedAnalysis.write_model_map(): Generating model
map for component 00.
2024-09-25 14:04:48 INFO    GTAnalysis.write_roi(): Writing /home/chen/PHYS2015A_
fermipy_working/notebooks/PKS2155_data/PKS_firstfit.npy...
```

By default fit will repeat the fit until a fit quality of 3 is obtained. After the fit returns all sources with free parameters will have their properties (flux, TS, NPred, etc.) updated in the ROIModel instance. The return value of the method is a dictionary containing the following diagnostic information about the fit.

The fit also accepts keyword arguments which can be used to configure its behavior at runtime:

In [53]:
```
print(first_fit['fit_quality'])
print(first_fit['errors'])
print(first_fit['loglike'])
print(first_fit['values'])
```

```
3
[1.28082675e-01 1.50575621e+00 3.64605732e+00 3.96070321e-01
 3.19331903e+00 1.76275664e+00 3.48762356e-04 4.70940342e-02
 2.92638574e-01 2.35259241e-01 3.67318796e-01 2.63622749e-01
 5.42138542e-01 4.52330207e-01 7.18666733e-01 1.54193044e+01
 4.23748550e-02 9.14314545e-02 6.51474095e-02 1.90515883e-01]
-8722.425022024823
[ 3.83814889e-02  1.98741979e+00  9.41626154e+00  2.99568641e+00
  1.87918182e+00  3.47892567e+00  2.50494621e-04 -4.99742991e+00
  1.83581090e+00  3.81334242e-01  1.60363440e+00  2.59678069e-01
  2.26722855e+00  4.52083661e-01  2.28626328e+00  2.32686473e+02
 -1.89390044e+00  9.63127640e-01 -1.17521789e-01  5.98131449e-01]
```

In [54]:
```
print(gta.roi.sources[0]['param_names'])
print(gta.roi.sources[0]['param_values'])
print(gta.roi.sources[0]['param_errors'])
```

```
[b'Prefactor' b'Index' b'Scale' b'' b'' b'' b'' b'' b'' b'']
[ 2.32686473e-11 -1.89390044e+00  1.00000000e+03             nan
             nan             nan             nan             nan
             nan             nan]
[1.54193044e-12 4.23748550e-02            nan            nan
            nan            nan            nan            nan
            nan            nan]
```

The optimize method performs an automatic optimization of the ROI by fitting all sources with an iterative strategy. It is generally good practice to run this method once at the start of your analysis to ensure that all parameters are close to their global likelihood maxima.

In [55]:
```
gta.load_roi('initial')
gta.print_model()
```

```
2024-09-25 14:04:54 INFO    GTAnalysis.load_roi(): Loading ROI file: /home/chen/P
HYS2015A_fermipy_working/notebooks/PKS2155_data/initial.npy
2024-09-25 14:04:54 INFO    GTBinnedAnalysis._create_binned_analysis(): Creating
BinnedAnalysis for component 00.
2024-09-25 14:04:56 INFO    GTAnalysis.load_roi(): Finished Loading ROI
2024-09-25 14:04:56 INFO    GTAnalysis.print_model():
sourcename             offset    norm  eflux     index        ts       npred  free
----------------------------------------------------------------------------------
3FGL J2158.8-3013       0.000   2.798  0.000176   1.87   3439.09       573.6    *
3FGL J2151.8-3025       1.539   0.000  9.83e-07   4.78      8.41        11.9    *
3FGL J2159.2-2841       1.542   0.384  5.58e-06   1.61     21.16         9.1    *
3FGL J2151.6-2744       2.948   1.879  4.05e-07   3.48      1.51         5.1    *
3FGL J2144.9-3356       4.743   9.384  4.04e-06   2.99     73.33        41.2    *
3FGL J2213.1-2532       5.650   0.258  1.94e-06   2.26     10.72        10.9    *
3FGL J2142.2-2546       5.760   0.038  3.34e-07   1.96      0.14         1.2    *
3FGL J2222.3-3500       6.878   0.452  1.06e-06   2.29      4.28         5.7    *
isodiff                   ---   0.600     0.018   2.12     38.40       283.1    *
galdiff                   ---   0.964     0.152  -0.12    817.13       921.3    *
```

In [65]:
```python
gta.load_roi('PKS_firstfit')
gta.print_model()
```

```
2024-09-25 14:22:25 INFO    GTAnalysis.load_roi(): Loading ROI file: /home/chen/P
HYS2015A_fermipy_working/notebooks/PKS2155_data/PKS_firstfit.npy
2024-09-25 14:22:25 INFO    GTBinnedAnalysis._create_binned_analysis(): Creating
BinnedAnalysis for component 00.
2024-09-25 14:22:27 INFO    GTAnalysis.load_roi(): Finished Loading ROI
2024-09-25 14:22:27 INFO    GTAnalysis.print_model():
sourcename             offset    norm  eflux     index        ts       npred  free
----------------------------------------------------------------------------------
PKS2155-304             0.000 232.686  0.000182   1.89   3437.83       574.2    *
3FGL J2151.8-3025       1.539   0.000  9.76e-07   4.77      8.30        11.8    *
3FGL J2159.2-2841       1.542   0.381  5.61e-06   1.60     21.12         9.0    *
3FGL J2151.6-2744       2.948   1.879  4.05e-07   3.48      1.51         5.1    *
3FGL J2144.9-3356       4.743   9.416  4.04e-06   3.00     73.37        41.2    *
3FGL J2213.1-2532       5.650   0.260  1.94e-06   2.27     10.73        10.9    *
3FGL J2142.2-2546       5.760   0.038  3.18e-07   1.99      0.14         1.2    *
3FGL J2222.3-3500       6.878   0.452  1.06e-06   2.29      4.29         5.7    *
isodiff                   ---   0.598     0.018   2.12     38.17       282.4    *
galdiff                   ---   0.963     0.153  -0.12    819.19       921.6    *
```

TS Map

tsmap() generates a test statistic (TS) map for an additional source component centered at each spatial bin in the ROI. The methodology is similar to that of the gttsmap ST application but with the following approximations:

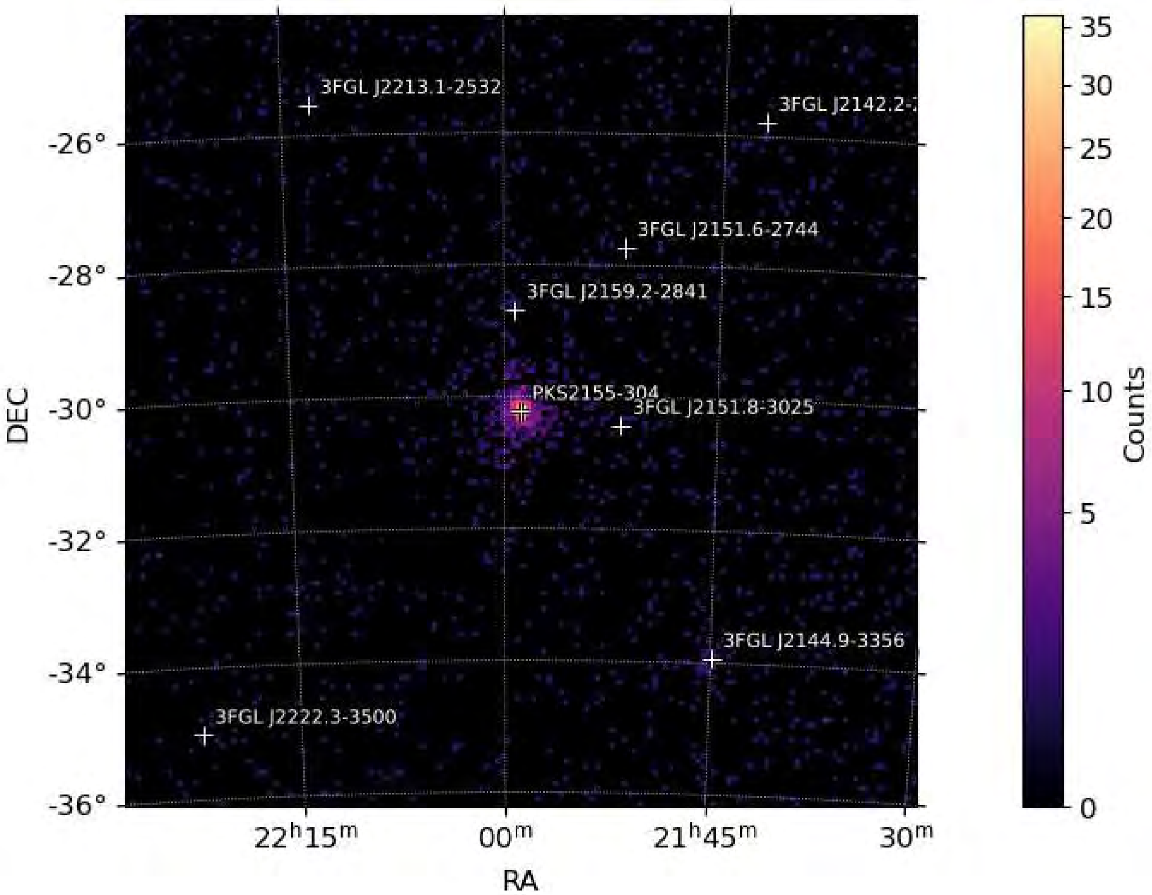
- Evaluation of the likelihood is limited to pixels in the vicinity of the test source position.
- The background model is fixed when fitting the test source amplitude.

TS Cube is a related method that can also be used to generate TS maps as well as cubes (TS vs. position and energy).

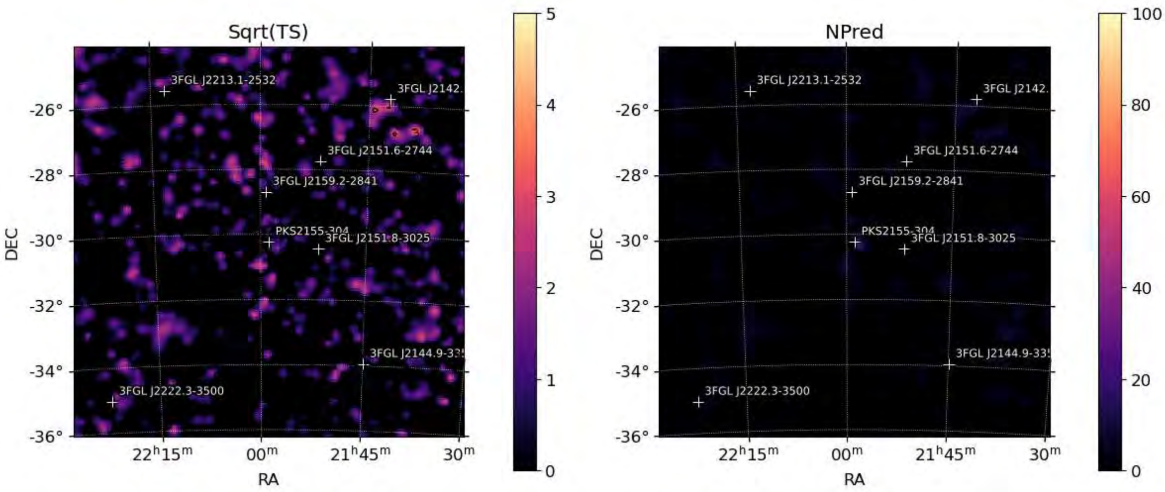
In [41]:
```
plt.clf()
tsmap_postfit = gta.tsmap(prefix='TSmap_start',make_plots=True,write_fits=True,w
```

```
2024-09-24 19:27:01 INFO    GTAnalysis.tsmap(): Generating TS map
2024-09-24 19:27:01 INFO    GTAnalysis._make_tsmap_fast(): Fitting test source.
2024-09-24 19:27:09 INFO    GTAnalysis.tsmap(): Finished TS map
2024-09-24 19:27:19 WARNING GTAnalysis.tsmap(): Saving TS maps in .npy files is d
isabled b/c of incompatibilities in python3, remove the maps from the /home/chen/
PHYS2015A_fermipy_working/notebooks/PKS2155_data/TSmap_start_pointsource_powerlaw
_2.00_tsmap.npy
2024-09-24 19:27:19 INFO    GTAnalysis.tsmap(): Execution time: 17.86 s
```

In [42]:
```
plt.clf()
fig = plt.figure(figsize=(14,6))
ROIPlotter(tsmap_postfit['sqrt_ts'],roi=gta.roi).plot(levels=[0,3,5,7],vmin=0,vm
plt.gca().set_title('Sqrt(TS)')
ROIPlotter(tsmap_postfit['npred'],roi=gta.roi).plot(vmin=0,vmax=100,subplot=122,
plt.gca().set_title('NPred')
plt.show()
```




```
<Figure size 800x600 with 0 Axes>
```



Looking to the TSmap it is quite clear that the model does not fit sufficiently well the data.

Residual Map

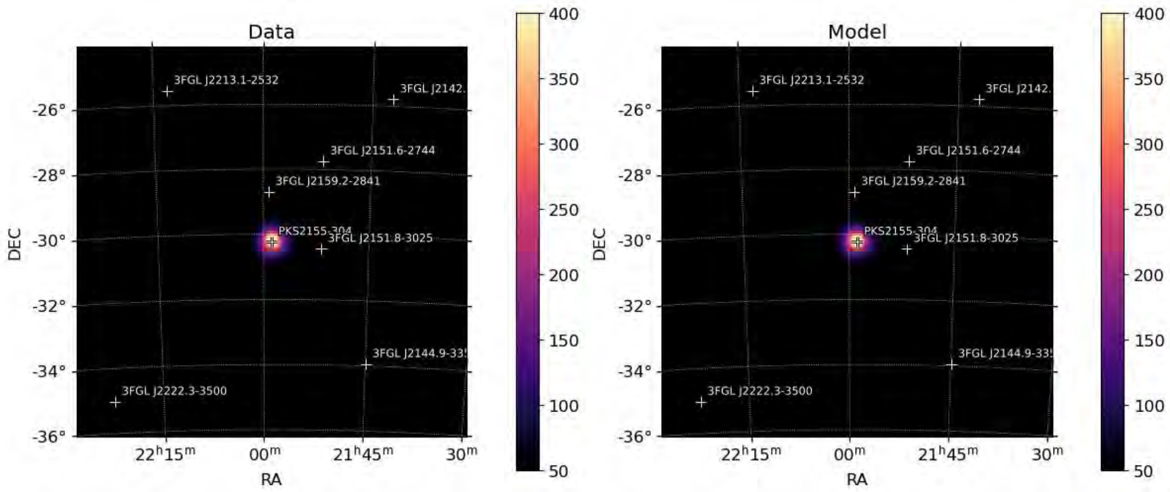
residmap() calculates the residual between smoothed data and model maps. Whereas a TS map is only sensitive to positive deviations with respect to the model, residmap() is sensitive to both positive and negative residuals and therefore can be useful for assessing the model goodness-of-fit.

In [43]:
```
resid = gta.residmap('SMC_postfit',model={'SpatialModel' : 'PointSource', 'Index
```

```
2024-09-24 19:27:34 INFO    GTAnalysis.residmap(): Generating residual maps
2024-09-24 19:27:34 INFO    GTAnalysis.add_source(): Adding source residmap_tests
ource
2024-09-24 19:27:35 INFO    GTAnalysis.delete_source(): Deleting source residmap_
testsource
2024-09-24 19:27:37 INFO    GTAnalysis.residmap(): Finished residual maps
2024-09-24 19:27:43 WARNING GTAnalysis.residmap(): Saving maps in .npy files is d
isabled b/c of incompatibilities in python3, remove the maps from the /home/chen/
PHYS2015A_fermipy_working/notebooks/PKS2155_data/SMC_postfit_pointsource_powerlaw
_2.00_residmap.npy
2024-09-24 19:27:43 INFO    GTAnalysis.residmap(): Execution time: 8.93 s
```

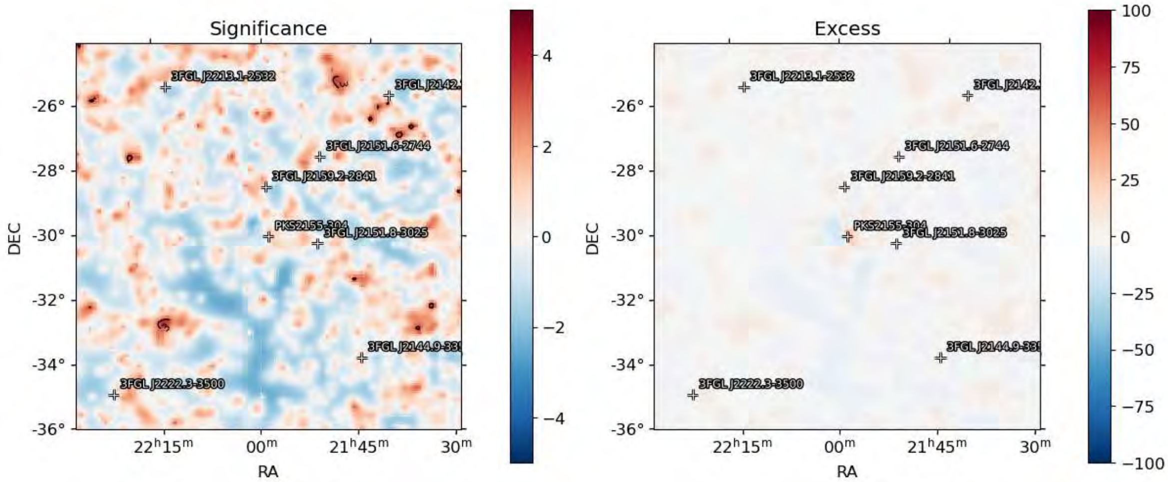
In [44]:
```
plt.clf()
fig = plt.figure(figsize=(14,6))
ROIPlotter(resid['data'],roi=gta.roi).plot(vmin=50,vmax=400,subplot=121,cmap='ma
plt.gca().set_title('Data')
ROIPlotter(resid['model'],roi=gta.roi).plot(vmin=50,vmax=400,subplot=122,cmap='m
plt.gca().set_title('Model')
plt.show()
```

```
<Figure size 640x480 with 0 Axes>
```




```
In [45]: plt.clf()
fig = plt.figure(figsize=(14,6))
ROIPlotter(resid['sigma'],roi=gta.roi).plot(vmin=-5,vmax=5,levels=[-5,-3,3,5],su
plt.gca().set_title('Significance')
ROIPlotter(resid['excess'],roi=gta.roi).plot(vmin=-100,vmax=100,subplot=122,cmap
plt.gca().set_title('Excess')
plt.show()
```

```
<Figure size 640x480 with 0 Axes>
```



Source Localization

The localize() method can be used to spatially localize a source. Localization is performed by scanning the likelihood surface in source position in a local patch around the nominal source position. The fit to the source position proceeds in two iterations:

TS Map Scan: Obtain a first estimate of the source position by generating a likelihood map of the region using the tsmap method. In this step all background parameters are fixed to their nominal values. The size of the search region used for this step is set with the dtheta_max parameter. Likelihood Scan: Refine the position of the source by performing a scan of the likelihood surface in a box centered on the best-fit position found in the first iteration. The size of the search region is set to encompass the 99% positional uncertainty contour. This method uses a full likelihood fit at each point in the likelihood scan and will re-fit all free parameters of the model. If a peak is found in the search region and the positional fit succeeds, the method will update the position of the source in the model to the new best-fit position.

In [47]:
```
gta.free_sources(free=False)
gta.print_model()
gta.free_sources(skydir=gta.roi[gta.roi.sources[0].name].skydir,distance=[3.0],f
gta.print_model()
localpks = gta.localize(gta.roi.sources[0].name, update=True, make_plots=True)
gta.print_model()
```

```
2024-09-24 19:29:36 INFO    GTAnalysis.free_source(): Fixing parameters for PKS21
55-304          : ['Prefactor', 'Index']
2024-09-24 19:29:36 INFO    GTAnalysis.free_source(): Fixing parameters for 3FGL
J2151.8-3025     : ['norm', 'alpha', 'beta']
2024-09-24 19:29:36 INFO    GTAnalysis.free_source(): Fixing parameters for 3FGL
J2159.2-2841     : ['Prefactor', 'Index']
2024-09-24 19:29:36 INFO    GTAnalysis.free_source(): Fixing parameters for 3FGL
J2151.6-2744     : ['Prefactor', 'Index']
2024-09-24 19:29:36 INFO    GTAnalysis.free_source(): Fixing parameters for isodi
ff               : ['Normalization']
2024-09-24 19:29:36 INFO    GTAnalysis.free_source(): Fixing parameters for galdi
ff               : ['Prefactor', 'Index']
2024-09-24 19:29:36 INFO    GTAnalysis.print_model():
sourcename              offset    norm  eflux     index        ts       npred free
--------------------------------------------------------------------------------
PKS2155-304              0.009 232.713  0.000182   1.89   3664.84       574.2
3FGL J2151.8-3025        1.539   0.000  9.83e-07   4.74      8.31        11.9
3FGL J2159.2-2841        1.542   0.381   5.6e-06   1.60     21.11         9.0
3FGL J2151.6-2744        2.948   1.870  4.05e-07   3.47      1.50         5.1
3FGL J2144.9-3356        4.743   9.378  4.04e-06   2.99     73.34        41.1
3FGL J2213.1-2532        5.650   0.259  1.95e-06   2.27     10.74        10.9
3FGL J2142.2-2546        5.760   0.038  3.25e-07   1.97      0.14         1.2
3FGL J2222.3-3500        6.878   0.452  1.06e-06   2.29      4.30         5.7
isodiff                    ---   0.596    0.0179   2.12     37.76       281.2
galdiff                    ---   0.964     0.153  -0.12    822.90       922.8

2024-09-24 19:29:36 INFO    GTAnalysis.free_source(): Freeing parameters for PKS2
155-304          : ['Prefactor', 'Index']
2024-09-24 19:29:36 INFO    GTAnalysis.free_source(): Freeing parameters for 3FGL
J2151.8-3025     : ['norm', 'alpha', 'beta']
2024-09-24 19:29:36 INFO    GTAnalysis.free_source(): Freeing parameters for 3FGL
J2159.2-2841     : ['Prefactor', 'Index']
2024-09-24 19:29:36 INFO    GTAnalysis.free_source(): Freeing parameters for 3FGL
J2151.6-2744     : ['Prefactor', 'Index']
2024-09-24 19:29:36 INFO    GTAnalysis.free_source(): Freeing parameters for isod
iff              : ['Normalization']
2024-09-24 19:29:36 INFO    GTAnalysis.free_source(): Freeing parameters for gald
iff              : ['Prefactor', 'Index']
2024-09-24 19:29:36 INFO    GTAnalysis.print_model():
sourcename              offset    norm  eflux     index        ts       npred free
--------------------------------------------------------------------------------
PKS2155-304              0.009 232.713  0.000182   1.89   3664.84       574.2    *
3FGL J2151.8-3025        1.539   0.000  9.83e-07   4.74      8.31        11.9    *
3FGL J2159.2-2841        1.542   0.381   5.6e-06   1.60     21.11         9.0    *
3FGL J2151.6-2744        2.948   1.870  4.05e-07   3.47      1.50         5.1    *
3FGL J2144.9-3356        4.743   9.378  4.04e-06   2.99     73.34        41.1
3FGL J2213.1-2532        5.650   0.259  1.95e-06   2.27     10.74        10.9
3FGL J2142.2-2546        5.760   0.038  3.25e-07   1.97      0.14         1.2
3FGL J2222.3-3500        6.878   0.452  1.06e-06   2.29      4.30         5.7
isodiff                    ---   0.596    0.0179   2.12     37.76       281.2    *
galdiff                    ---   0.964     0.153  -0.12    822.90       922.8    *

2024-09-24 19:29:36 INFO    GTAnalysis.localize(): Running localization for PKS21
55-304
2024-09-24 19:29:40 INFO    GTAnalysis._localize(): Localization succeeded.
2024-09-24 19:29:40 INFO    GTAnalysis._localize(): Updating source PKS2155-304 t
o localized position.
2024-09-24 19:29:40 INFO    GTAnalysis.delete_source(): Deleting source PKS2155-3
04
2024-09-24 19:29:40 INFO    GTAnalysis.add_source(): Adding source PKS2155-304
```

```
2024-09-24 19:29:40 INFO    GTAnalysis._localize(): Localization completed with n
ew position:
(  ra, dec) = (  329.7097 +/-   0.0059,  -30.2252 +/-   0.0057)
(glon,glat) = (   17.7300 +/-   0.0057,  -52.2396 +/-   0.0059)
offset =   0.0000 r68 =   0.0088 r95 =   0.0142 r99 =   0.0176
2024-09-24 19:29:40 INFO    GTAnalysis._localize(): LogLike:    -8721.103 DeltaLo
gLike:       -0.000
2024-09-24 19:29:40 INFO    GTAnalysis.localize(): Finished localization.
2024-09-24 19:29:46 WARNING GTAnalysis.localize(): Saving TS maps in .npy files i
s disabled b/c of incompatibilities in python3, remove the maps from the /home/ch
en/PHYS2015A_fermipy_working/notebooks/PKS2155_data/pks2155-304_loc.npy
2024-09-24 19:29:46 INFO    GTAnalysis.localize(): Execution time: 10.19 s
2024-09-24 19:29:46 INFO    GTAnalysis.print_model():
sourcename            offset    norm  eflux     index        ts        npred free
--------------------------------------------------------------------------------
PKS2155-304            0.009 232.710 0.000182   1.89   3664.84        574.2    *
3FGL J2151.8-3025      1.539   0.000 9.83e-07   4.74      8.31         11.9    *
3FGL J2159.2-2841      1.542   0.381  5.6e-06   1.60     21.11          9.0    *
3FGL J2151.6-2744      2.948   1.870 4.05e-07   3.47      1.50          5.1    *
3FGL J2144.9-3356      4.743   9.378 4.04e-06   2.99     73.34         41.1
3FGL J2213.1-2532      5.650   0.259 1.95e-06   2.27     10.74         10.9
3FGL J2142.2-2546      5.760   0.038 3.25e-07   1.97      0.14          1.2
3FGL J2222.3-3500      6.878   0.452 1.06e-06   2.29      4.30          5.7
isodiff                  ---   0.596   0.0179   2.12     37.76        281.2    *
galdiff                  ---   0.964    0.153  -0.12    822.90        922.8    *
```

PKS2155-304 is relocalized at 0.009deg.

In [49]:

```python
print(localpks['glon'])
print(localpks['glat'])
print(localpks['pos_r68'])
print(localpks['pos_r95'])
print(localpks['pos_r99'])
print(localpks['pos_err_semimajor'])
print(localpks['pos_err_semiminor'])
print(localpks['dloglike_loc'])
```

```
17.730028344812883
-52.2396077094473
0.008791961331231373
0.014188460076057336
0.017593476546711506
0.005902349857603159
0.00569401693096676
-2.592358214315027e-06
```

Extension Fitting

The extension() method executes a source extension analysis for a given source by computing a likelihood ratio test with respect to the no-extension (point-source) hypothesis and a best-fit model for extension. The best-fit extension is found by performing a likelihood profile scan over the source width (68% containment) and fitting for the extension that maximizes the model likelihood. Currently this method supports two models for extension: a 2D Gaussian (RadialGaussian) or a 2D disk (RadialDisk).

By default the method will fix all background parameters before performing the extension fit. One can leave background parameters free by setting free_background=True.

In [51]:
```
gta.free_sources(free=False)
gta.print_model()
gta.free_sources(skydir=gta.roi[gta.roi.sources[0].name].skydir,distance=[3.0],f
gta.print_model()
extensionpks = gta.extension(gta.roi.sources[0].name,update=True,make_plots=True
gta.print_model()
```

```
2024-09-24 19:30:48 INFO    GTAnalysis.free_source(): Fixing parameters for PKS21
55-304           : ['Prefactor', 'Index']
2024-09-24 19:30:48 INFO    GTAnalysis.free_source(): Fixing parameters for 3FGL
J2151.8-3025     : ['norm', 'alpha', 'beta']
2024-09-24 19:30:48 INFO    GTAnalysis.free_source(): Fixing parameters for 3FGL
J2159.2-2841     : ['Prefactor', 'Index']
2024-09-24 19:30:48 INFO    GTAnalysis.free_source(): Fixing parameters for 3FGL
J2151.6-2744     : ['Prefactor', 'Index']
2024-09-24 19:30:48 INFO    GTAnalysis.free_source(): Fixing parameters for isodi
ff               : ['Normalization']
2024-09-24 19:30:48 INFO    GTAnalysis.free_source(): Fixing parameters for galdi
ff               : ['Prefactor', 'Index']
2024-09-24 19:30:48 INFO    GTAnalysis.print_model():
sourcename              offset    norm  eflux     index        ts       npred free
--------------------------------------------------------------------------------
PKS2155-304              0.009 232.710  0.000182   1.89   3664.84       574.2
3FGL J2151.8-3025        1.539   0.000  9.83e-07   4.74      8.31        11.9
3FGL J2159.2-2841        1.542   0.381   5.6e-06   1.60     21.11         9.0
3FGL J2151.6-2744        2.948   1.870  4.05e-07   3.47      1.50         5.1
3FGL J2144.9-3356        4.743   9.378  4.04e-06   2.99     73.34        41.1
3FGL J2213.1-2532        5.650   0.259  1.95e-06   2.27     10.74        10.9
3FGL J2142.2-2546        5.760   0.038  3.25e-07   1.97      0.14         1.2
3FGL J2222.3-3500        6.878   0.452  1.06e-06   2.29      4.30         5.7
isodiff                    ---   0.596    0.0179   2.12     37.76       281.2
galdiff                    ---   0.964     0.153  -0.12    822.90       922.8

2024-09-24 19:30:48 INFO    GTAnalysis.free_source(): Freeing parameters for PKS2
155-304           : ['Prefactor', 'Index']
2024-09-24 19:30:48 INFO    GTAnalysis.free_source(): Freeing parameters for 3FGL
J2151.8-3025     : ['norm', 'alpha', 'beta']
2024-09-24 19:30:48 INFO    GTAnalysis.free_source(): Freeing parameters for 3FGL
J2159.2-2841     : ['Prefactor', 'Index']
2024-09-24 19:30:48 INFO    GTAnalysis.free_source(): Freeing parameters for 3FGL
J2151.6-2744     : ['Prefactor', 'Index']
2024-09-24 19:30:48 INFO    GTAnalysis.free_source(): Freeing parameters for isod
iff               : ['Normalization']
2024-09-24 19:30:48 INFO    GTAnalysis.free_source(): Freeing parameters for gald
iff               : ['Prefactor', 'Index']
2024-09-24 19:30:48 INFO    GTAnalysis.print_model():
sourcename              offset    norm  eflux     index        ts       npred free
--------------------------------------------------------------------------------
PKS2155-304              0.009 232.710  0.000182   1.89   3664.84       574.2    *
3FGL J2151.8-3025        1.539   0.000  9.83e-07   4.74      8.31        11.9    *
3FGL J2159.2-2841        1.542   0.381   5.6e-06   1.60     21.11         9.0    *
3FGL J2151.6-2744        2.948   1.870  4.05e-07   3.47      1.50         5.1    *
3FGL J2144.9-3356        4.743   9.378  4.04e-06   2.99     73.34        41.1
3FGL J2213.1-2532        5.650   0.259  1.95e-06   2.27     10.74        10.9
3FGL J2142.2-2546        5.760   0.038  3.25e-07   1.97      0.14         1.2
3FGL J2222.3-3500        6.878   0.452  1.06e-06   2.29      4.30         5.7
isodiff                    ---   0.596    0.0179   2.12     37.76       281.2    *
galdiff                    ---   0.964     0.153  -0.12    822.90       922.8    *

2024-09-24 19:30:48 INFO    GTAnalysis.extension(): Running extension fit for PKS
2155-304
2024-09-24 19:30:56 INFO    GTAnalysis._extension(): Fitting extended-source mode
l.
2024-09-24 19:30:59 INFO    GTAnalysis._extension(): Generating TS map.
2024-09-24 19:31:00 INFO    GTAnalysis._extension(): Testing point-source model.
2024-09-24 19:31:01 INFO    GTAnalysis._extension(): Best-fit extension: 0.0265 +
0.0183 -     nan
```

```
2024-09-24 19:31:01 INFO    GTAnalysis._extension(): TS_ext:        0.529
2024-09-24 19:31:01 INFO    GTAnalysis._extension(): Extension UL: 0.0537
2024-09-24 19:31:01 INFO    GTAnalysis._extension(): LogLike:    -8720.930 DeltaL
ogLike:        0.173
2024-09-24 19:31:01 INFO    GTAnalysis.extension(): Finished extension fit.
{'spatial_model': 'RadialGaussian', 'width': [], 'fit_position': False, 'width_mi
n': 0.01, 'width_max': 1.0, 'width_nstep': 21, 'free_background': False, 'fix_sha
pe': False, 'free_radius': None, 'fit_ebin': False, 'update': True, 'save_model_m
ap': False, 'sqrt_ts_threshold': 3.0, 'psf_scale_fn': None, 'make_tsmap': True,
'tsmap_fitter': 'tsmap', 'make_plots': True, 'write_fits': True, 'write_npy': Tru
e, 'reoptimize': False, 'optimizer': {'optimizer': 'MINUIT', 'tol': 0.001, 'max_i
ter': 100, 'init_lambda': 0.0001, 'retries': 3, 'min_fit_quality': 2, 'verbosit
y': 0}, 'prefix': '', 'outfile': None, 'loge_bins': []}
{'spatial_model': 'RadialGaussian', 'width': [], 'fit_position': False, 'width_mi
n': 0.01, 'width_max': 1.0, 'width_nstep': 21, 'free_background': False, 'fix_sha
pe': False, 'free_radius': None, 'fit_ebin': False, 'update': True, 'save_model_m
ap': False, 'sqrt_ts_threshold': 3.0, 'psf_scale_fn': None, 'make_tsmap': True,
'tsmap_fitter': 'tsmap', 'make_plots': True, 'write_fits': True, 'write_npy': Tru
e, 'reoptimize': False, 'optimizer': {'optimizer': 'MINUIT', 'tol': 0.001, 'max_i
ter': 100, 'init_lambda': 0.0001, 'retries': 3, 'min_fit_quality': 2, 'verbosit
y': 0}, 'prefix': '', 'outfile': None, 'loge_bins': []}
2024-09-24 19:31:04 WARNING GTAnalysis.extension(): Saving maps in .npy files is
disabled b/c of incompatibilities in python3, remove the maps from the /home/che
n/PHYS2015A_fermipy_working/notebooks/PKS2155_data/pks2155-304_ext.npy
2024-09-24 19:31:04 INFO    GTAnalysis.extension(): Execution time: 15.96 s
2024-09-24 19:31:04 INFO    GTAnalysis.print_model():
sourcename              offset    norm  eflux     index        ts       npred free
--------------------------------------------------------------------------------
PKS2155-304              0.009 232.710 0.000182   1.89   3664.84       574.2    *
3FGL J2151.8-3025        1.539   0.000 9.83e-07   4.74      8.31        11.9    *
3FGL J2159.2-2841        1.542   0.381  5.6e-06   1.60     21.11         9.0    *
3FGL J2151.6-2744        2.948   1.870 4.05e-07   3.47      1.50         5.1    *
3FGL J2144.9-3356        4.743   9.378 4.04e-06   2.99     73.34        41.1
3FGL J2213.1-2532        5.650   0.259 1.95e-06   2.27     10.74        10.9
3FGL J2142.2-2546        5.760   0.038 3.25e-07   1.97      0.14         1.2
3FGL J2222.3-3500        6.878   0.452 1.06e-06   2.29      4.30         5.7
isodiff                    ---   0.596   0.0179   2.12     37.76       281.2    *
galdiff                    ---   0.964    0.153  -0.12    822.90       922.8    *
```

{'name': 'PKS2155-304', 'file': None, 'config': {'spatial_model': 'RadialGaussia
n', 'width': [], 'fit_position': False, 'width_min': 0.01, 'width_max': 1.0, 'wid
th_nstep': 21, 'free_background': False, 'fix_shape': False, 'free_radius': None,
'fit_ebin': False, 'update': True, 'save_model_map': False, 'sqrt_ts_threshold':
3.0, 'psf_scale_fn': None, 'make_tsmap': True, 'tsmap_fitter': 'tsmap', 'make_plo
ts': True, 'write_fits': True, 'write_npy': True, 'reoptimize': False, 'optimize
r': {'optimizer': 'MINUIT', 'tol': 0.001, 'max_iter': 100, 'init_lambda': 0.0001,
'retries': 3, 'min_fit_quality': 2, 'verbosity': 0}, 'prefix': '', 'outfile': Non
e, 'loge_bins': []}, 'width': array([0.   , 0.01   , 0.01258925, 0.015848
93, 0.01995262,
       0.02511886, 0.03162278, 0.03981072, 0.05011872, 0.06309573,
       0.07943282, 0.1   , 0.12589254, 0.15848932, 0.19952623,
       0.25118864, 0.31622777, 0.39810717, 0.50118723, 0.63095734,
       0.79432823, 1.   ]), 'dloglike': array([8.82405567e-03, 7.77112250e
-02, 1.14383682e-01, 1.62099470e-01,
        2.16428007e-01, 2.60050236e-01, 2.43559785e-01, 3.53590399e-02,
       -6.72668025e-01, -2.54269958e+00, -6.67280763e+00, -1.43274844e+01,
       -2.68273278e+01, -4.61723140e+01, -7.37227735e+01, -1.10553042e+02,
       -1.61968390e+02, -2.27506139e+02, -3.08209775e+02, -4.05694144e+02,
       -5.19544634e+02, -6.44670081e+02]), 'loglike': array([-8721.18603949, -872
1.11715232, -8721.08047987, -8721.03276408,
       -8720.97843554, -8720.93481331, -8720.95130376, -8721.15950451,
       -8721.86753157, -8723.73756313, -8727.86767118, -8735.52234794,
       -8748.02219139, -8767.36717757, -8794.91763704, -8831.74790548,
       -8883.1632531 , -8948.70100257, -9029.40463819, -9126.88900739,
       -9240.73949755, -9365.86494445]), 'loglike_ptsrc': -8721.194863548704, 'lo
glike_ext': -8720.930295480644, 'loglike_init': -8721.102957346766, 'loglike_bas
e': -8721.102957346704, 'ext': 0.026527076368933644, 'ext_err_hi': 0.018311827105
248167, 'ext_err_lo': nan, 'ext_err': 0.018311827105248167, 'ext_ul95': 0.0537120
37289574695, 'ts_ext': 0.5291361361196323, 'ebin_e_min': array([], dtype=float6
4), 'ebin_e_ctr': array([], dtype=float64), 'ebin_e_max': array([], dtype=float6
4), 'ebin_ext': array([], dtype=float64), 'ebin_ext_err': array([], dtype=float6
4), 'ebin_ext_err_hi': array([], dtype=float64), 'ebin_ext_err_lo': array([], dty
pe=float64), 'ebin_ext_ul95': array([], dtype=float64), 'ebin_ts_ext': array([],
dtype=float64), 'ebin_dloglike': array([], shape=(0, 22), dtype=float64), 'ebin_l
oglike': array([], shape=(0, 22), dtype=float64), 'ebin_loglike_ptsrc': array([],
dtype=float64), 'ebin_loglike_ext': array([], dtype=float64), 'ra': 329.709717458
93144, 'dec': -30.225189700941957, 'glon': 17.730028344812883, 'glat': -52.239607
7094473, 'ra_err': nan, 'dec_err': nan, 'glon_err': nan, 'glat_err': nan, 'pos_of
fset': 0.0, 'pos_err': nan, 'pos_r68': nan, 'pos_r95': nan, 'pos_r99': nan, 'pos_
err_semimajor': nan, 'pos_err_semiminor': nan, 'pos_angle': nan, 'tsmap': <gammap
y.maps.wcs.ndmap.WcsNDMap object at 0x7f9bc645edf0>, 'ptsrc_tot_map': None, 'ptsr
c_src_map': None, 'ptsrc_bkg_map': None, 'ext_tot_map': None, 'ext_src_map': Non
e, 'ext_bkg_map': None, 'source_fit': {'param_names': array([b'Prefactor', b'Inde
x', b'Scale', b'', b'', b'', b'', b'', b'',
       b''], dtype='|S32'), 'param_values': array([2.32024684e-11, -1.89421590e+
00, 1.00000000e+03,   nan,
                   nan,   nan,   nan,   nan,
                   nan,   nan]), 'param_errors': array([1.52410662e-12,
4.22431706e-02,   nan,   nan,
                  nan,   nan,   nan,   nan,
                  nan,   nan]), 'ts': 3665.185165882667, 'loglike': -872
0.930295480644, 'loglike_scan': array([-10553.52287842, -8723.64081848, -8722.9
8746977, -8722.42853383,
        -8721.96162888, -8721.58446368, -8721.29483293, -8721.09061299,
        -8720.96975788, -8720.93029548, -8720.95934547, -8721.04473646,
        -8721.18537116, -8721.38018453, -8721.62814257, -8721.9282411 ,
        -8722.27950462, -8722.68098521, -8723.13176148, -8723.63093761]), 'dlo
glike_scan': array([-1.83259258e+03, -2.71052300e+00, -2.05717429e+00, -1.4982383
5e+00,

```
       -1.03133340e+00, -6.54168201e-01, -3.64537450e-01, -1.60317509e-01,
       -3.94623954e-02,  0.00000000e+00, -2.90499868e-02, -1.14440983e-01,
       -2.55075677e-01, -4.49889045e-01, -6.97847087e-01, -9.97945621e-01,
       -1.34920914e+00, -1.75068973e+00, -2.20146600e+00, -2.70064213e+00]), 'efl
ux_scan': array([0.        , 0.0001635 , 0.00016577, 0.00016804, 0.00017031,
       0.00017258, 0.00017486, 0.00017713, 0.0001794 , 0.00018167,
       0.0001836 , 0.00018553, 0.00018747, 0.0001894 , 0.00019133,
       0.00019326, 0.00019519, 0.00019713, 0.00019906, 0.00020099]), 'flux_scan':
array([0.00000000e+00, 2.32142376e-08, 2.35368145e-08, 2.38593914e-08,
       2.41819683e-08, 2.45045452e-08, 2.48271220e-08, 2.51496989e-08,
       2.54722758e-08, 2.57948527e-08, 2.60691405e-08, 2.63434284e-08,
       2.66177162e-08, 2.68920041e-08, 2.71662919e-08, 2.74405798e-08,
       2.77148677e-08, 2.79891555e-08, 2.82634434e-08, 2.85377312e-08]), 'norm_sc
an': array([  0.        , 208.8120526 , 211.71363147, 214.61521034,
       217.51678922, 220.41836809, 223.31994697, 226.22152584,
       229.12310471, 232.02468359, 234.49190264, 236.95912168,
       239.42634073, 241.89355978, 244.36077883, 246.82799787,
       249.29521692, 251.76243597, 254.22965502, 256.69687406]), 'npred': 573.050
8537830627, 'npred_wt': 573.0508537830627, 'pivot_energy': 3148.6820532287256, 'f
lux': 2.5794852691648842e-08, 'flux100': 2.0331862595554595e-07, 'flux1000': 2.58
02236910381953e-08, 'flux10000': 3.1597690630353774e-09, 'flux_err': 1.1459883795
68179e-09, 'flux100_err': 2.1996996038847975e-08, 'flux1000_err': 1.1462262972690
278e-09, 'flux10000_err': 3.1444544970508274e-10, 'flux_ul95': 2.7716379398887197
e-08, 'flux100_ul95': 2.1846436741500665e-07, 'flux1000_ul95': 2.7724313687577186
e-08, 'flux10000_ul95': 3.39514860622981e-09, 'eflux': 0.00018167122134412315, 'e
flux100': 0.0002313298182472882, 'eflux1000': 0.00018391231462068695, 'eflux1000
0': 0.00012341749841835394, 'eflux_err': 1.7267603815980034e-05, 'eflux100_err':
1.5186656894825425e-05, 'eflux1000_err': 1.7698113976490774e-05, 'eflux10000_er
r': 1.650953707128165e-05, 'eflux_ul95': 0.00019520439045822165, 'eflux100_ul95':
0.0002485621874046693, 'eflux1000_ul95': 0.00019761242869220804, 'eflux10000_ul9
5': 0.00013261119385000848, 'dnde': 2.6422303665521376e-12, 'dnde100': 1.81865402
7109073e-09, 'dnde1000': 2.3202468358820944e-11, 'dnde10000': 2.9601811554991236e
-13, 'dnde_err': 1.1719723722799424e-13, 'dnde100_err': 2.770169790154615e-10, 'd
nde1000_err': 1.5241066205768404e-12, 'dnde10000_err': 1.952516727176953e-14, 'dn
de_index': 1.8942158995212823, 'dnde100_index': 1.8942158995101803, 'dnde1000_ind
ex': 1.8942158995212823, 'dnde10000_index': 1.8942158995101803, 'name': 'PKS2155-
304', 'spectral_pars': {'Prefactor': {'name': 'Prefactor', 'value': 232.024683588
20937, 'error': 15.241066205768401, 'min': 0.0001, 'max': 10000.0, 'free': True,
'scale': 1.0000000000000003e-13}, 'Index': {'name': 'Index', 'value': -1.89421589
95161362, 'error': 0.04224317056806755, 'min': -5.0, 'max': -1.0, 'free': True,
'scale': 1.0}, 'Scale': {'name': 'Scale', 'value': 1000.0, 'error': nan, 'min':
1.0, 'max': 100000.0, 'free': False, 'scale': 1.0}}, 'model_counts': array([123.2
5618433, 100.44802129,  79.90195231,  62.29486275,
        47.86824402,  36.54448302,  28.16091053,  21.94499345,
        16.99768446,  13.12455476,  10.16858074,   7.97525634,
         6.29239505,   4.91475723,   3.80383378,   2.93788233,
         2.26704315,   1.74968619,   1.35298708,   1.04654097]), 'model_counts_w
t': array([123.25618433, 100.44802129,  79.90195231,  62.29486275,
        47.86824402,  36.54448302,  28.16091053,  21.94499345,
        16.99768446,  13.12455476,  10.16858074,   7.97525634,
         6.29239505,   4.91475723,   3.80383378,   2.93788233,
         2.26704315,   1.74968619,   1.35298708,   1.04654097]), 'covar': array
([[ 2.32290099e+02, -4.74881730e-01],
       [-4.74881730e-01,  1.78448546e-03]]), 'model_flux': {'energies': array([
1000.        ,  1123.44927778,  1262.13827975,  1417.94833884,
         1592.9930372 ,  1789.64687715,  2010.57749162,  2258.78183088,
         2537.62681657,  2850.89501435,  3202.8359449 ,  3598.22372914,
         4042.4218498 ,  4541.45590764,  5102.09535951,  5731.94534681,
         6439.54986016,  7234.50763962,  8127.60238283,  9130.94902708,
        10258.15808993, 11524.52029749, 12947.21400498, 14545.53822317,
```

```
         16341.17441175,  18358.48059096,  20624.82176107,  23170.94111182,
         26031.37705758,  29244.93175497,  32855.19745886,  36911.1478565 ,
         41467.80240143,  46586.97265904,  52338.10078778,  58799.20153044,
         66057.92049345,  74212.72307006,  83374.23013519,  93666.71863089,
        105229.80739795, 118220.35112222, 132814.56808722, 149210.43059633,
        167630.35049078, 188324.19619298, 211572.68220161, 237691.17701751,
        267033.98115514, 299999.13327161]), 'log_energies': array([3.        , 3.0
5055347, 3.10110694, 3.15166041, 3.20221388,
        3.25276735, 3.30332082, 3.35387429, 3.40442776, 3.45498122,
        3.50553469, 3.55608816, 3.60664163, 3.6571951 , 3.70774857,
        3.75830204, 3.80885551, 3.85940898, 3.90996245, 3.96051592,
        4.01106939, 4.06162286, 4.11217633, 4.1627298 , 4.21328327,
        4.26383673, 4.3143902 , 4.36494367, 4.41549714, 4.46605061,
        4.51660408, 4.56715755, 4.61771102, 4.66826449, 4.71881796,
        4.76937143, 4.8199249 , 4.87047837, 4.92103184, 4.97158531,
        5.02213878, 5.07269224, 5.12324571, 5.17379918, 5.22435265,
        5.27490612, 5.32545959, 5.37601306, 5.42656653, 5.47712   ]), 'dnde': arra
y([2.32024684e-11, 1.86112269e-11, 1.49284878e-11, 1.19744792e-11,
        9.60500172e-12, 7.70439005e-12, 6.17986625e-12, 4.95701109e-12,
        3.97613119e-12, 3.18934514e-12, 2.55824618e-12, 2.05202737e-12,
        1.64597776e-12, 1.32027615e-12, 1.05902349e-12, 8.49466794e-13,
        6.81376610e-13, 5.46547656e-13, 4.38398289e-13, 3.51649225e-13,
        2.82065831e-13, 2.26251410e-13, 1.81481395e-13, 1.45570349e-13,
        1.16765283e-13, 9.36600859e-14, 7.51268823e-14, 6.02609787e-14,
        4.83366998e-14, 3.87719648e-14, 3.10998736e-14, 2.49459150e-14,
        2.00096851e-14, 1.60502229e-14, 1.28742483e-14, 1.03267270e-14,
        8.28330230e-15, 6.64422492e-15, 5.32948373e-15, 4.27489995e-15,
        3.42899435e-15, 2.75047425e-15, 2.20621787e-15, 1.76965747e-15,
        1.41948245e-15, 1.13859912e-15, 9.13296212e-16, 7.32575636e-16,
        5.87615558e-16, 4.71339786e-16]), 'dnde_lo': array([2.17723052e-11, 1.7522
2355e-11, 1.40993131e-11, 1.13426439e-11,
        9.12264304e-12, 7.33494712e-12, 5.89551096e-12, 4.73665606e-12,
        3.80388482e-12, 3.05331001e-12, 2.44957963e-12, 1.96420289e-12,
        1.57420285e-12, 1.26103621e-12, 1.00973026e-12, 8.08193965e-13,
        6.46666705e-13, 5.17274942e-13, 4.13673705e-13, 3.30755506e-13,
        2.64413533e-13, 2.11348988e-13, 1.68914529e-13, 1.34987293e-13,
        1.07866119e-13, 8.61885386e-14, 6.88638493e-14, 5.50192375e-14,
        4.39564728e-14, 3.51171323e-14, 2.80547027e-14, 2.24122169e-14,
        1.79043379e-14, 1.43030131e-14, 1.14259925e-14, 9.12764146e-15,
        7.29159290e-15, 5.82487069e-15, 4.65319139e-15, 3.71720850e-15,
        2.96951037e-15, 2.37222111e-15, 1.89508325e-15, 1.51392579e-15,
        1.20944011e-15, 9.66201956e-16, 7.71890083e-16, 6.16662060e-16,
        4.92655577e-16, 3.93590083e-16]), 'dnde_hi': array([2.47265750e-11, 1.9767
8981e-11, 1.58064260e-11, 1.26415104e-11,
        1.01128651e-11, 8.09244089e-12, 6.47793672e-12, 5.18761731e-12,
        4.15617717e-12, 3.33144109e-12, 2.67173331e-12, 2.14377870e-12,
        1.72102521e-12, 1.38229901e-12, 1.11072312e-12, 8.92847343e-13,
        7.17949573e-13, 5.77476919e-13, 4.64600619e-13, 3.73862794e-13,
        3.00896599e-13, 2.42204616e-13, 1.94983207e-13, 1.56983121e-13,
        1.26398646e-13, 1.01779330e-13, 8.19595260e-14, 6.60021062e-14,
        5.31534128e-14, 4.28071758e-14, 3.44755797e-14, 2.77660474e-14,
        2.23625972e-14, 1.80108661e-14, 1.45060721e-14, 1.16833348e-14,
        9.40989136e-15, 7.57883346e-15, 6.10406802e-15, 4.91626165e-15,
        3.95957608e-15, 3.18904026e-15, 2.56843455e-15, 2.06858722e-15,
        1.66600264e-15, 1.34175670e-15, 1.08060719e-15, 8.70277414e-16,
        7.00879193e-16, 5.64448148e-16]), 'dnde_err': array([1.52410662e-12, 1.156
67125e-12, 8.77938144e-13, 6.67031235e-13,
        5.07863344e-13, 3.88050839e-13, 2.98070462e-13, 2.30606222e-13,
        1.80045980e-13, 1.42095952e-13, 1.13487135e-13, 9.17513282e-14,
        7.50474476e-14, 6.20228617e-14, 5.16996318e-14, 4.33805487e-14,
```

```
        3.65729634e-14, 3.09292635e-14, 2.62023303e-14, 2.22135684e-14,
        1.88307686e-14, 1.59532061e-14, 1.35018124e-14, 1.14127717e-14,
        9.63336308e-15, 8.11924385e-15, 6.83264370e-15, 5.74112749e-15,
        4.81671304e-15, 4.03521095e-15, 3.37570609e-15, 2.82013237e-15,
        2.35291213e-15, 1.96064323e-15, 1.63182372e-15, 1.35660777e-15,
        1.12658906e-15, 9.34608533e-16, 7.74584297e-16, 6.41361700e-16,
        5.30581731e-16, 4.38566004e-16, 3.62216681e-16, 2.98929754e-16,
        2.46520192e-16, 2.03157581e-16, 1.67310974e-16, 1.37701777e-16,
        1.13263636e-16, 9.31083622e-17]), 'dnde_ferr': array([0.06366283, 0.060330
86, 0.05717635, 0.05423478, 0.05154721,
        0.04915988, 0.04712284, 0.04548721, 0.0443009 , 0.04360317,
        0.04341914, 0.0437557 , 0.04460035, 0.04592327, 0.04768207,
        0.04982736, 0.05230798, 0.05507477, 0.05808293, 0.06129302,
        0.06467119, 0.06818881, 0.0718219 , 0.07555051, 0.07935804,
        0.08323071, 0.08715706, 0.09112753, 0.09513413, 0.09917015,
        0.10322995, 0.10730876, 0.11140254, 0.11550783, 0.11962172,
        0.12374169, 0.12786558, 0.13199153, 0.13611793, 0.14024342,
        0.14436677, 0.14848696, 0.15260308, 0.15671435, 0.16082007,
        0.16491965, 0.16901258, 0.17309841, 0.17717674, 0.18124723]), 'pivot_energ
y': 3148.6820532287256}}}
```

In this specific case PKS2155-304 is found not to be extended, with TSext=0.529 and with an angular extension upper limit of 0.0537deg.

In [52]:
```python
print(extensionpks['ext'])
print(extensionpks['ext_err_hi'])
print(extensionpks['ext_err_lo'])
print(extensionpks['ext_err'])
print(extensionpks['ext'])
print(extensionpks['ext_ul95'])
print(extensionpks['ts_ext'])
```

```
0.026527076368933644
0.018311827105248167
nan
0.018311827105248167
0.026527076368933644
0.053712037289574695
0.5291361361196323
```

Source Finding

find_sources() is an iterative source-finding algorithm that uses peak detection on a TS map to find new source candidates. The procedure for adding new sources at each iteration is as follows:

- Generate a TS map for the test source model defined with the model argument.
- Identify peaks with sqrt(TS) > sqrt_ts_threshold and an angular distance of at least min_separation from a higher amplitude peak in the map.
- Order the peaks by TS and add a source at each peak starting from the highest TS peak. Set the source position by fitting a 2D parabola to the log-likelihood surface around the peak maximum. After adding each source, re-fit its spectral parameters.
- Add sources at the N highest peaks up to N = sources_per_iter.

Source finding is repeated up to max_iter iterations or until no peaks are found in a given iteration. Sources found by the method are added to the model and given designations PS JXXXX.X+XXXX according to their position in celestial coordinates.

This may take a few minutes.

In [53]:
```
gta.free_sources()
model = {'Index' : 2.0, 'SpatialModel' : 'PointSource'}
findsource26 = gta.find_sources(model=model,sqrt_ts_threshold=5,min_separation=0
```

```
2024-09-24 19:34:15 INFO    GTAnalysis.free_source(): Freeing parameters for 3FGL
J2144.9-3356     : ['Prefactor', 'Index']
2024-09-24 19:34:15 INFO    GTAnalysis.free_source(): Freeing parameters for 3FGL
J2213.1-2532     : ['Prefactor', 'Index']
2024-09-24 19:34:15 INFO    GTAnalysis.free_source(): Freeing parameters for 3FGL
J2142.2-2546     : ['Prefactor', 'Index']
2024-09-24 19:34:15 INFO    GTAnalysis.free_source(): Freeing parameters for 3FGL
J2222.3-3500     : ['Prefactor', 'Index']
2024-09-24 19:34:15 INFO    GTAnalysis.find_sources(): Starting.
2024-09-24 19:34:15 INFO    GTAnalysis.tsmap(): Generating TS map
2024-09-24 19:34:16 INFO    GTAnalysis._make_tsmap_fast(): Fitting test source.
2024-09-24 19:34:22 INFO    GTAnalysis.tsmap(): Finished TS map
2024-09-24 19:34:30 WARNING GTAnalysis.tsmap(): Saving TS maps in .npy files is d
isabled b/c of incompatibilities in python3, remove the maps from the /home/chen/
PHYS2015A_fermipy_working/notebooks/PKS2155_data/sourcefind_00_pointsource_powerl
aw_2.00_tsmap.npy
2024-09-24 19:34:30 INFO    GTAnalysis.tsmap(): Execution time: 14.70 s
2024-09-24 19:34:30 INFO    GTAnalysis.find_sources(): Found 0 sources in iterati
on 0.
2024-09-24 19:34:30 INFO    GTAnalysis.find_sources(): Done.
2024-09-24 19:34:30 INFO    GTAnalysis.find_sources(): Execution time: 14.70 s
```

In [54]:
```
gta.print_model()
gta.write_roi('PKS_relext_TS25',make_plots=True,save_model_map=True)
```

```
2024-09-24 19:35:42 INFO    GTAnalysis.print_model():
sourcename              offset    norm  eflux      index       ts       npred free
--------------------------------------------------------------------------------
PKS2155-304              0.009 232.710  0.000182    1.89    3664.84      574.2    *
3FGL J2151.8-3025        1.539   0.000  9.83e-07    4.74       8.31       11.9    *
3FGL J2159.2-2841        1.542   0.381   5.6e-06    1.60      21.11        9.0    *
3FGL J2151.6-2744        2.948   1.870  4.05e-07    3.47       1.50        5.1    *
3FGL J2144.9-3356        4.743   9.378  4.04e-06    2.99      73.34       41.1    *
3FGL J2213.1-2532        5.650   0.259  1.95e-06    2.27      10.74       10.9    *
3FGL J2142.2-2546        5.760   0.038  3.25e-07    1.97       0.14        1.2    *
3FGL J2222.3-3500        6.878   0.452  1.06e-06    2.29       4.30        5.7    *
isodiff                    ---   0.596    0.0179    2.12      37.76      281.2    *
galdiff                    ---   0.964     0.153   -0.12     822.90      922.8    *

2024-09-24 19:35:42 INFO    GTBinnedAnalysis.write_xml(): Writing /home/chen/PHYS
2015A_fermipy_working/notebooks/PKS2155_data/PKS_relext_TS25_00.xml...
2024-09-24 19:35:42 INFO    GTAnalysis.write_fits(): Writing /home/chen/PHYS2015A
_fermipy_working/notebooks/PKS2155_data/PKS_relext_TS25.fits...
WARNING: Format %s cannot be mapped to the accepted TDISPn keyword values.  Forma
t will not be moved into TDISPn keyword. [astropy.io.fits.column]
WARNING: Format %f cannot be mapped to the accepted TDISPn keyword values.  Forma
t will not be moved into TDISPn keyword. [astropy.io.fits.column]
WARNING: Format %s cannot be mapped to the accepted TDISPn keyword values.  Forma
t will not be moved into TDISPn keyword. [astropy.io.fits.column]
2024-09-24 19:35:51 INFO    GTBinnedAnalysis.write_model_map(): Generating model
map for component 00.
2024-09-24 19:35:53 INFO    GTAnalysis.write_roi(): Writing /home/chen/PHYS2015A_
fermipy_working/notebooks/PKS2155_data/PKS_relext_TS25.npy...
```

In [55]:

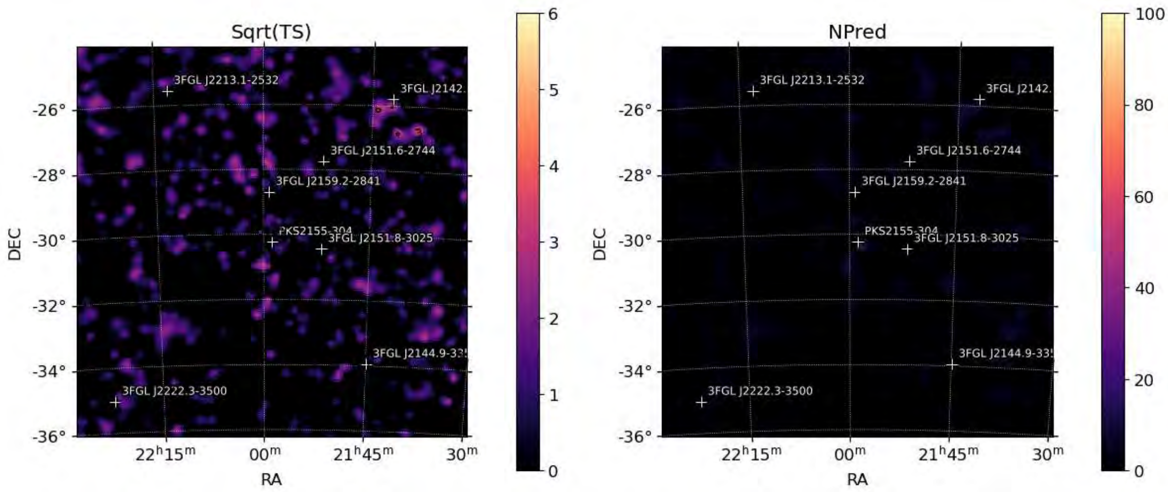
```
tsmap_relext = gta.tsmap(prefix='TSmap_relext_TS25',make_plots=True,write_fits=T
```

```
2024-09-24 19:36:13 INFO    GTAnalysis.tsmap(): Generating TS map
2024-09-24 19:36:14 INFO    GTAnalysis._make_tsmap_fast(): Fitting test source.
2024-09-24 19:36:22 INFO    GTAnalysis.tsmap(): Finished TS map
2024-09-24 19:36:30 WARNING GTAnalysis.tsmap(): Saving TS maps in .npy files is d
isabled b/c of incompatibilities in python3, remove the maps from the /home/chen/
PHYS2015A_fermipy_working/notebooks/PKS2155_data/TSmap_relext_TS25_pointsource_po
werlaw_2.00_tsmap.npy
2024-09-24 19:36:30 INFO    GTAnalysis.tsmap(): Execution time: 16.15 s
```

In [56]:

```
plt.clf()
fig = plt.figure(figsize=(14,6))
ROIPlotter(tsmap_relext['sqrt_ts'],roi=gta.roi).plot(levels=[0,3,5,7],vmin=0,vma
plt.gca().set_title('Sqrt(TS)')
ROIPlotter(tsmap_relext['npred'],roi=gta.roi).plot(vmin=0,vmax=100,subplot=122,c
plt.gca().set_title('NPred')
plt.show()
```

```
<Figure size 800x600 with 0 Axes>
```

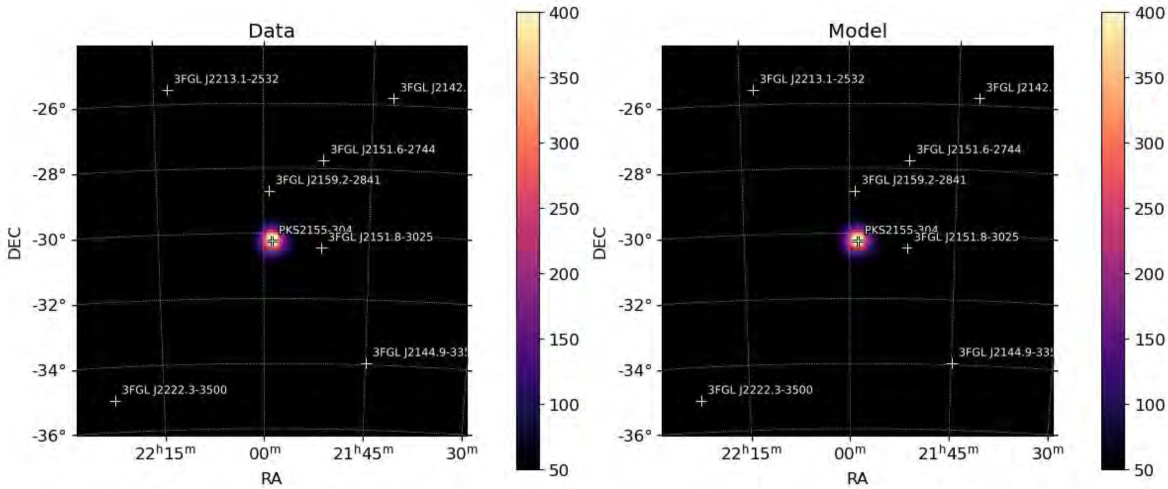


In [57]:
```
resid_relext = gta.residmap('TSmap_relext_TS26',model={'SpatialModel' : 'PointSc
```

```
2024-09-24 19:36:45 INFO    GTAnalysis.residmap(): Generating residual maps
2024-09-24 19:36:45 INFO    GTAnalysis.add_source(): Adding source residmap_tests
ource
2024-09-24 19:36:46 INFO    GTAnalysis.delete_source(): Deleting source residmap_
testsource
2024-09-24 19:36:47 INFO    GTAnalysis.residmap(): Finished residual maps
2024-09-24 19:36:54 WARNING GTAnalysis.residmap(): Saving maps in .npy files is d
isabled b/c of incompatibilities in python3, remove the maps from the /home/chen/
PHYS2015A_fermipy_working/notebooks/PKS2155_data/TSmap_relext_TS26_pointsource_po
werlaw_2.00_residmap.npy
2024-09-24 19:36:54 INFO    GTAnalysis.residmap(): Execution time: 9.17 s
```

In [58]:
```
plt.clf()
fig = plt.figure(figsize=(14,6))
ROIPlotter(resid_relext['data'],roi=gta.roi).plot(vmin=50,vmax=400,subplot=121,c
plt.gca().set_title('Data')
ROIPlotter(resid_relext['model'],roi=gta.roi).plot(vmin=50,vmax=400,subplot=122,
plt.gca().set_title('Model')
plt.show()
```

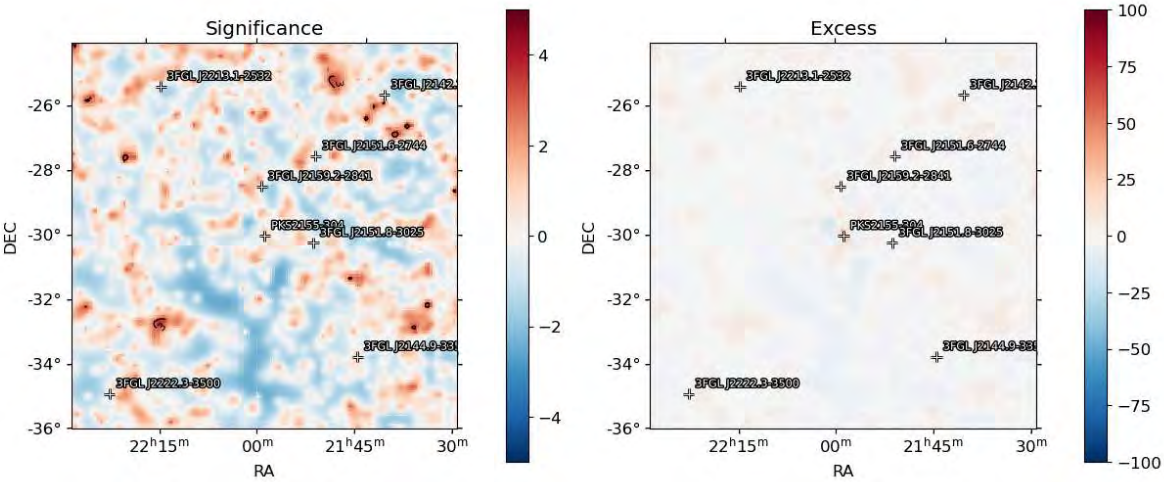
```
<Figure size 640x480 with 0 Axes>
```



In [59]:
```
plt.clf()
fig = plt.figure(figsize=(14,6))
ROIPlotter(resid_relext['sigma'],roi=gta.roi).plot(vmin=-5,vmax=5,levels=[-5,-3,
plt.gca().set_title('Significance')
ROIPlotter(resid_relext['excess'],roi=gta.roi).plot(vmin=-100,vmax=100,subplot=1
```

```
plt.gca().set_title('Excess')
plt.show()
```

<Figure size 640x480 with 0 Axes>



Sed Analysis

The sed() method computes a spectral energy distribution (SED) by performing independent fits for the flux normalization of a source in bins of energy. The normalization in each bin is fit using a power-law spectral parameterization with a fixed index. The value of this index can be set with the bin_index parameter or allowed to vary over the energy range according to the local slope of the global spectral model (with the use_local_index parameter).

The free_background, free_radius, and cov_scale parameters control how nuisance parameters are dealt with in the fit. By default the method will fix the parameters of background components ROI when fitting the source normalization in each energy bin (free_background=False). Setting free_background=True will profile the normalizations of all background components that were free when the method was executed. In order to minimize overfitting, background normalization parameters are constrained with priors taken from the global fit. The strength of the priors is controlled with the cov_scale parameter. A larger (smaller) value of cov_scale applies a weaker (stronger) constraint on the background amplitude. Setting cov_scale=None performs an unconstrained fit without priors.

In [79]:
```
gta.free_sources(free=False)
gta.print_model()
gta.free_sources(skydir=gta.roi[gta.roi.sources[0].name].skydir,distance=[3.0],f
gta.print_model()
sedpks = gta.sed(gta.roi.sources[0].name, bin_index=2.2, outfile='sedPKS.fits',
```

```
2024-09-24 19:46:20 INFO    GTAnalysis.free_source(): Fixing parameters for PKS21
55-304           : ['Prefactor', 'Index']
2024-09-24 19:46:20 INFO    GTAnalysis.free_source(): Fixing parameters for 3FGL
J2151.8-3025     : ['norm', 'alpha', 'beta']
2024-09-24 19:46:20 INFO    GTAnalysis.free_source(): Fixing parameters for 3FGL
J2159.2-2841     : ['Prefactor', 'Index']
2024-09-24 19:46:20 INFO    GTAnalysis.free_source(): Fixing parameters for 3FGL
J2151.6-2744     : ['Prefactor', 'Index']
2024-09-24 19:46:20 INFO    GTAnalysis.free_source(): Fixing parameters for isodi
ff               : ['Normalization']
2024-09-24 19:46:20 INFO    GTAnalysis.free_source(): Fixing parameters for galdi
ff               : ['Prefactor', 'Index']
2024-09-24 19:46:20 INFO    GTAnalysis.print_model():
sourcename              offset    norm  eflux     index         ts       npred free
--------------------------------------------------------------------------------
PKS2155-304              0.009 232.710  0.000182   1.89   3664.84       574.2
3FGL J2151.8-3025        1.539   0.000  9.83e-07   4.74      8.31        11.9
3FGL J2159.2-2841        1.542   0.381   5.6e-06   1.60     21.11         9.0
3FGL J2151.6-2744        2.948   1.870  4.05e-07   3.47      1.50         5.1
3FGL J2144.9-3356        4.743   9.378  4.04e-06   2.99     73.34        41.1
3FGL J2213.1-2532        5.650   0.259  1.95e-06   2.27     10.74        10.9
3FGL J2142.2-2546        5.760   0.038  3.25e-07   1.97      0.14         1.2
3FGL J2222.3-3500        6.878   0.452  1.06e-06   2.29      4.30         5.7
isodiff                    ---   0.596    0.0179   2.12     37.76       281.2
galdiff                    ---   0.964     0.153  -0.12    822.90       922.8

2024-09-24 19:46:20 INFO    GTAnalysis.free_source(): Freeing parameters for PKS2
155-304           : ['Prefactor', 'Index']
2024-09-24 19:46:20 INFO    GTAnalysis.free_source(): Freeing parameters for 3FGL
J2151.8-3025     : ['norm', 'alpha', 'beta']
2024-09-24 19:46:20 INFO    GTAnalysis.free_source(): Freeing parameters for 3FGL
J2159.2-2841     : ['Prefactor', 'Index']
2024-09-24 19:46:20 INFO    GTAnalysis.free_source(): Freeing parameters for 3FGL
J2151.6-2744     : ['Prefactor', 'Index']
2024-09-24 19:46:20 INFO    GTAnalysis.free_source(): Freeing parameters for isod
iff               : ['Normalization']
2024-09-24 19:46:20 INFO    GTAnalysis.free_source(): Freeing parameters for gald
iff               : ['Prefactor', 'Index']
2024-09-24 19:46:20 INFO    GTAnalysis.print_model():
sourcename              offset    norm  eflux     index         ts       npred free
--------------------------------------------------------------------------------
PKS2155-304              0.009 232.710  0.000182   1.89   3664.84       574.2    *
3FGL J2151.8-3025        1.539   0.000  9.83e-07   4.74      8.31        11.9    *
3FGL J2159.2-2841        1.542   0.381   5.6e-06   1.60     21.11         9.0    *
3FGL J2151.6-2744        2.948   1.870  4.05e-07   3.47      1.50         5.1    *
3FGL J2144.9-3356        4.743   9.378  4.04e-06   2.99     73.34        41.1
3FGL J2213.1-2532        5.650   0.259  1.95e-06   2.27     10.74        10.9
3FGL J2142.2-2546        5.760   0.038  3.25e-07   1.97      0.14         1.2
3FGL J2222.3-3500        6.878   0.452  1.06e-06   2.29      4.30         5.7
isodiff                    ---   0.596    0.0179   2.12     37.76       281.2    *
galdiff                    ---   0.964     0.153  -0.12    822.90       922.8    *

2024-09-24 19:46:20 INFO    GTAnalysis.sed(): Computing SED for PKS2155-304
2024-09-24 19:46:20 INFO    GTAnalysis._make_sed(): Fitting SED
2024-09-24 19:46:20 INFO    GTAnalysis.free_source(): Fixing parameters for PKS21
55-304           : ['Index']
2024-09-24 19:46:20 INFO    GTAnalysis.free_source(): Fixing parameters for 3FGL
J2151.8-3025     : ['alpha', 'beta']
2024-09-24 19:46:20 INFO    GTAnalysis.free_source(): Fixing parameters for 3FGL
J2159.2-2841     : ['Index']
```

```
2024-09-24 19:46:20 INFO    GTAnalysis.free_source(): Fixing parameters for 3FGL J2151.6-2744      : ['Index']
2024-09-24 19:46:20 INFO    GTAnalysis.free_source(): Fixing parameters for galdiff             : ['Index']
2024-09-24 19:46:21 INFO    GTAnalysis.sed(): Finished SED
2024-09-24 19:46:28 INFO    GTAnalysis.sed(): Execution time: 8.27 s
```

In [66]:
```
print(sedpks['e_min'])
print(sedpks['e_max'])
print(sedpks['e_ref'])
print(sedpks['flux'])
print(sedpks['eflux'])
print(sedpks['e2dnde'])
print(sedpks['dnde_ul95'])
print(sedpks['ts'])
```

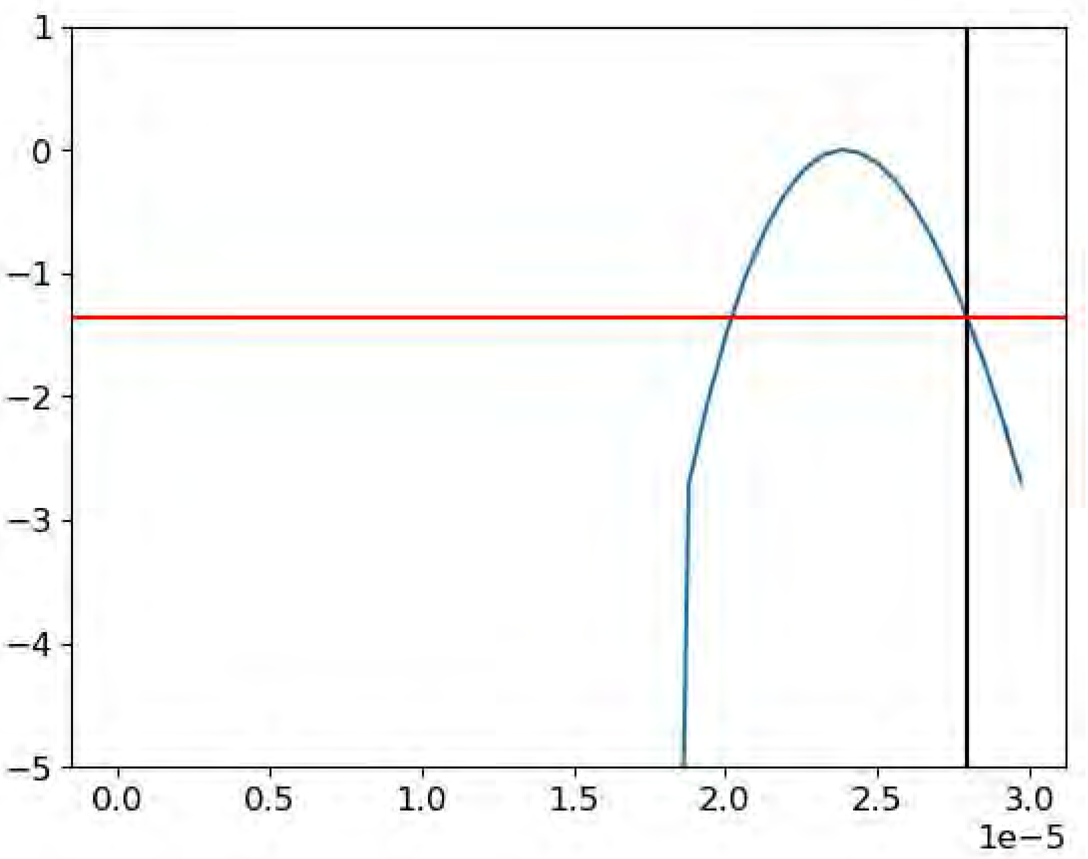
```
[   1000.           1330.02559927   1768.92736318   2352.71867628
   3129.10401621   4161.78844433   5535.2851697    7361.90145895
   9791.51739969  13022.66893578  17320.48305538  23036.68585534
  30638.67642017  40750.22396651  54197.59309129  72084.18623012
  95873.81298845 127511.6894751  169593.81120778 225564.11038375]
[   1330.02559927   1768.92736318   2352.71867628   3129.10401621
   4161.78844433   5535.2851697    7361.90145895   9791.51739969
  13022.66893578  17320.48305538  23036.68585534  30638.67642017
  40750.22396651  54197.59309129  72084.18623012  95873.81298845
 127511.6894751  169593.81120778 225564.11038375 299999.13327161]
[   1153.2673581    1533.85744979   2040.04618681   2713.28241415
   3608.69352201   4799.65476418   6383.59020979   8490.24064618
  11292.10739743  15018.61899903  19975.14773436  26567.15196845
  35334.58541089  46995.36207878  62504.31499998  83132.33901468
 110566.86606112 147055.06924831 195587.00660236 260132.73075933]
[5.92591562e-09 4.55589233e-09 3.07537599e-09 2.66977862e-09
 2.17683969e-09 1.71784746e-09 1.16651740e-09 1.28570557e-09
 1.07773961e-09 5.77599417e-10 1.97816409e-10 4.45257370e-10
 2.59881368e-10 2.13051116e-10 8.69502872e-11 8.26101680e-11
 4.33653714e-11 1.30206863e-10 4.37617270e-11 4.36833675e-11]
[6.80184141e-06 6.95504305e-06 6.24423526e-06 7.20960749e-06
 7.81839281e-06 8.20607789e-06 7.41135456e-06 1.08643203e-05
 1.21124003e-05 8.63371651e-06 3.93272294e-06 1.17732804e-05
 9.13936838e-06 9.96506612e-06 5.40906324e-06 6.83509482e-06
 4.77209896e-06 1.90570167e-05 8.51874250e-06 1.13097363e-05]
[2.38462917e-05 2.43853643e-05 2.18914036e-05 2.52779032e-05
 2.74101768e-05 2.87693457e-05 2.59852569e-05 3.80887669e-05
 4.24677879e-05 3.02685861e-05 1.37875691e-05 4.12787856e-05
 3.20413299e-05 3.49389307e-05 1.89634089e-05 2.39628735e-05
 1.67316537e-05 6.68111991e-05 2.98655035e-05 3.96535346e-05]
[2.10005979e-11 1.23323874e-11 6.44608171e-12 4.25537054e-12
 2.66103222e-12 1.62216987e-12 8.71288053e-13 7.14236686e-13
 4.58658392e-13 2.06794242e-13 7.01713385e-14 9.42067882e-14
 4.69006788e-14 3.05836848e-14 1.28282470e-14 9.45418661e-15
 4.98148281e-15 7.02162847e-15 2.84106485e-15 2.13382232e-15]
[519.37747602 480.8617535  349.27301068 369.86642839 341.45273383
 333.10468227 222.9049853  258.25107394 240.75139334 107.8268568
  36.81685166 148.35219256  75.36804485  62.16465513  23.77353533
  13.79702992  13.99918895  47.32787907  15.72433978  17.20898524]
```

In [67]:
```
# E^2 x Differential flux ULs in each bin in units of MeV cm^{-2} s^{-1}
print(sedpks['e2dnde_ul95'])

e2dnde_scan = sedpks['norm_scan']*sedpks['ref_e2dnde'][:,None]
```

```
plt.clf()
plt.figure()
plt.plot(e2dnde_scan[0],sedpks['dloglike_scan'][0]-np.max(sedpks['dloglike_scan'
plt.gca().set_ylim(-5,1)
plt.gca().axvline(sedpks['e2dnde_ul95'][0],color='k')
plt.gca().axhline(-2.71/2.,color='r')
plt.show()
```

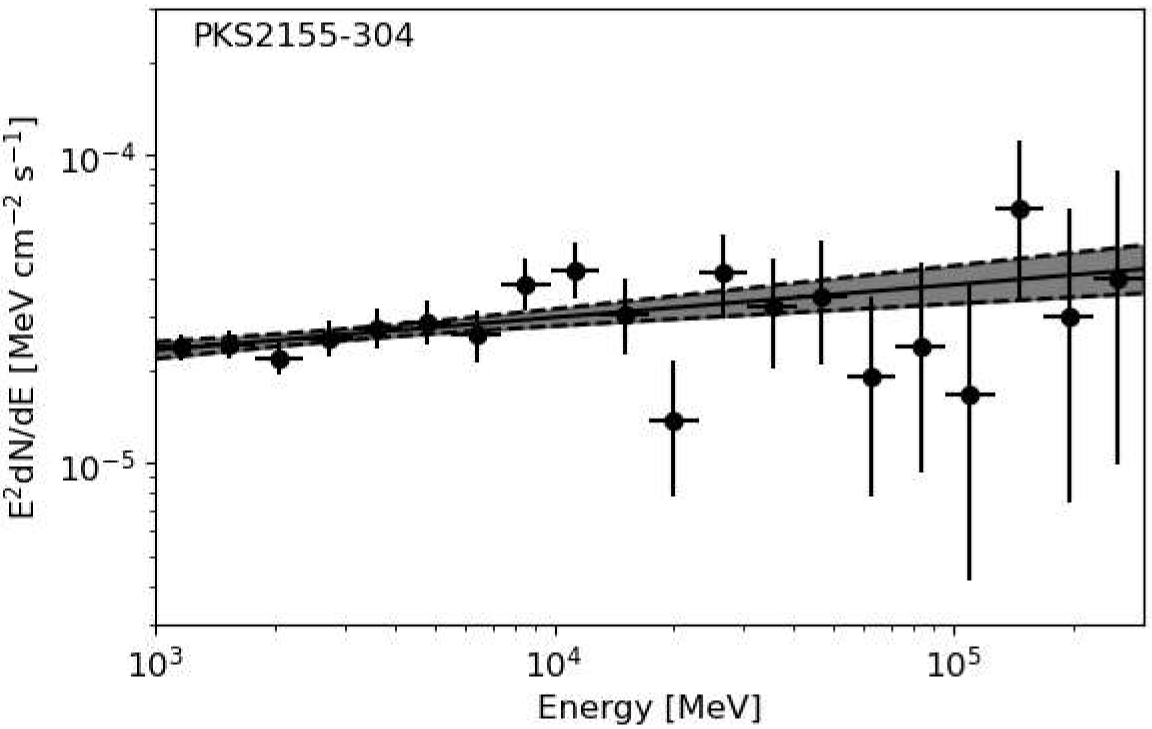
```
[2.79313328e-05 2.90146383e-05 2.68272284e-05 3.13276186e-05
 3.46537416e-05 3.73694178e-05 3.55051833e-05 5.14851703e-05
 5.84843064e-05 4.66442853e-05 2.79988221e-05 6.64924289e-05
 5.85570417e-05 6.75460269e-05 5.01172592e-05 6.53377493e-05
 6.08987861e-05 1.51844074e-04 1.08682882e-04 1.44393703e-04]
<Figure size 640x480 with 0 Axes>
```



In [75]:

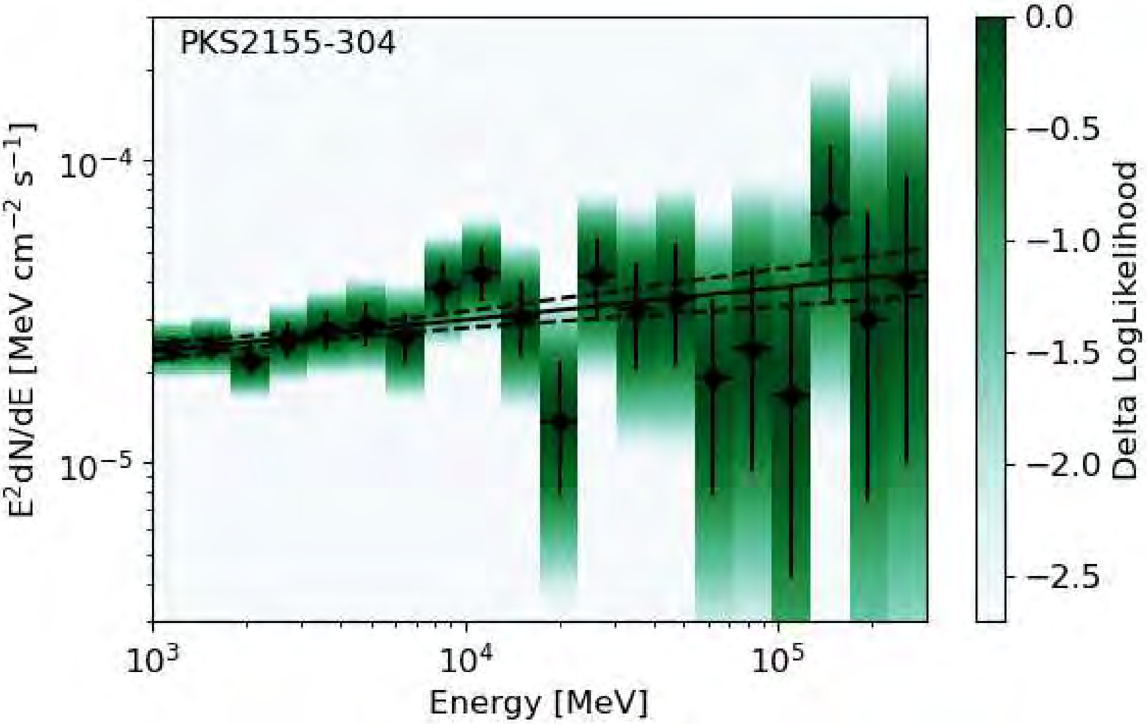
```
plt.clf()
fig = plt.figure(figsize=(14,4))
ylim=[3E-6,3E-4]
fig.add_subplot(121)
SEDPlotter(sedpks).plot()
plt.gca().set_ylim(ylim)
plt.show()
```

```
<Figure size 640x480 with 0 Axes>
```



In [76]:
```
plt.clf()
fig = plt.figure(figsize=(14,4))
fig.add_subplot(121)
SEDPlotter(sedpks).plot(showlnl=True, ylim=ylim)
plt.gca().set_ylim(ylim)
plt.show()
```

```
<Figure size 640x480 with 0 Axes>
```



In [85]:
```
from astropy.table import Table
sed_tab = Table.read('PKS2155_data/sedPKS.fits')
sed_tab.write('PKS2155_data/PKS_2155-304_sed.ecsv',exclude_names=['norm_scan','d
```

In [ ]:
```
```