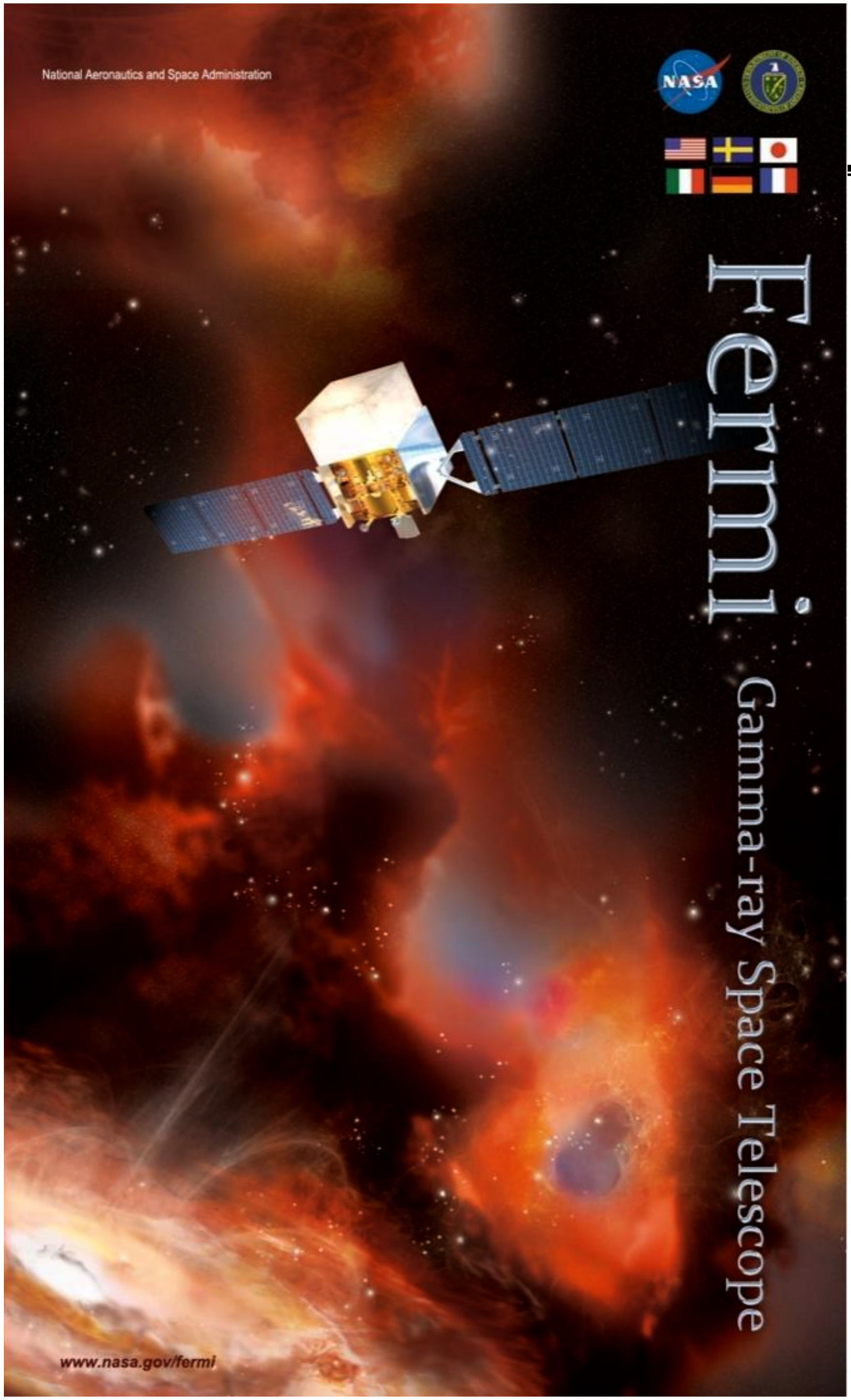


Fermi-LAT likelihood analysis

Julie McEnery

Outline

Likelihood analysis

- Introduction to likelihood
- Analysis tutorial
- Using python
- Exploiting the results
- Spectral analysis in energy bands
- Light curves

Likelihood analysis: basics (1)

The likelihood *L* is the probability of obtaining the data given an input model.

In our case, the input model is the distribution of gamma-ray sources on the sky, and includes their intensity and spectra.

One will maximize L to get the best match of the model to the data. Given a set of data, one can bin them in multidimensional (energy, sky pixels…) bins.

The observed number of counts in each bin is characterized by the Poisson distribution. *L* is the product of the probabilities of observing the detected counts in each bin, n_k, while m_k counts are predicted by the model:

$$\mathcal{L} = \prod_k \frac{m_k^{n_k} e^{-m_k}}{n_k!}$$

L can be rewritten as:

$$\mathcal{L} = \prod_k e^{-m_k} \prod_k \frac{m_k^{n_k}}{n_k!} = e^{-N_{pred}} \prod_k \frac{m_k^{n_k}}{n_k!}$$

Likelihood analysis: basics (2)

If we let the bin sizes get infinitesimally small, then nk=0 or 1, and we are left with a product running over the number of photons (unbinned likelihood).

$$\mathcal{L} = e^{-N_{pred}} \prod_i m_i$$

log L is easier to handle, this is usually the quantity that is maximized

$$\log\mathcal{L} = \sum_i \log(m_i) - N_{pred}$$

Likelihood analysis: basics (3)

The Test Statistic is defined as

$$TS = -2\log\left(\frac{\mathcal{L}_{\max,0}}{\mathcal{L}_{\max,1}}\right)$$

where $\mathcal{L}_{\max,0}$ is the maximum likelihood value for a model without an additional source (the 'null hypothesis') and $\mathcal{L}_{\max,1}$ is the maximum likelihood value for a model with the additional source at a specified location.

In the limit of a large number of counts, Wilk's Theorem states that the TS for the null hypothesis is asymptotically distributed as χ_n^2 where n is the number of parameters characterizing the additional source.

As a basic rule of thumb, the square root of the TS is approximately equal to the detection significance for a given source.

Likelihood analysis: basics (4)

The source model is considered as:

$$S(E,\hat{p},t) = \sum_i s_i(E,t)\delta(\hat{p}-\hat{p}_i) + S_G(E,\hat{p}) + S_{\rm eg}(E,\hat{p}) + \sum_l S_l(E,\hat{p},t),$$

Point Sources | Galactic & EG Diffuse Sources | Other Sources

This model is folded with the Instrument Response Functions (IRFs) to obtain the predicted counts in the measured quantity space (*E',p',t'*):

$$M(E',\hat{p}',t) = \int_{\rm SR} dE d\hat{p} R(E',\hat{p}',t;E,\hat{p}) S(E,\hat{p},t)$$

where

$$R(E',\hat{p}';E,\hat{p},t) = A(E,\hat{p},\vec{L}(t)) D(E';E,\hat{p},\vec{L}(t)) P(\hat{p}';E,\hat{p},\vec{L}(t))$$

is the combined IRF. $\vec{L}(t)$ is the orientation vector of the spacescraft. The integral is performed over the Source Region, i.e. the sky region encompassing all sources contributing to the Region-of – Interest (ROI). In the standard analysis, only steady sources are considered

$$S(E,\hat{p},t) \rightarrow S(E,\hat{p})$$

Likelihood analysis: basics (5)

The function to maximize is:

$$\log \mathcal{L} = \sum_j \log M(E'_j, \hat{p}'_j, t_j) - N_{\rm pred}$$

where the sum is performed over photons in the ROI. The predicted number of counts is:

$$N_{\rm pred} = \int_{\rm ROI} dE' d\hat{p}'\, dt M(E', \hat{p}', t)$$

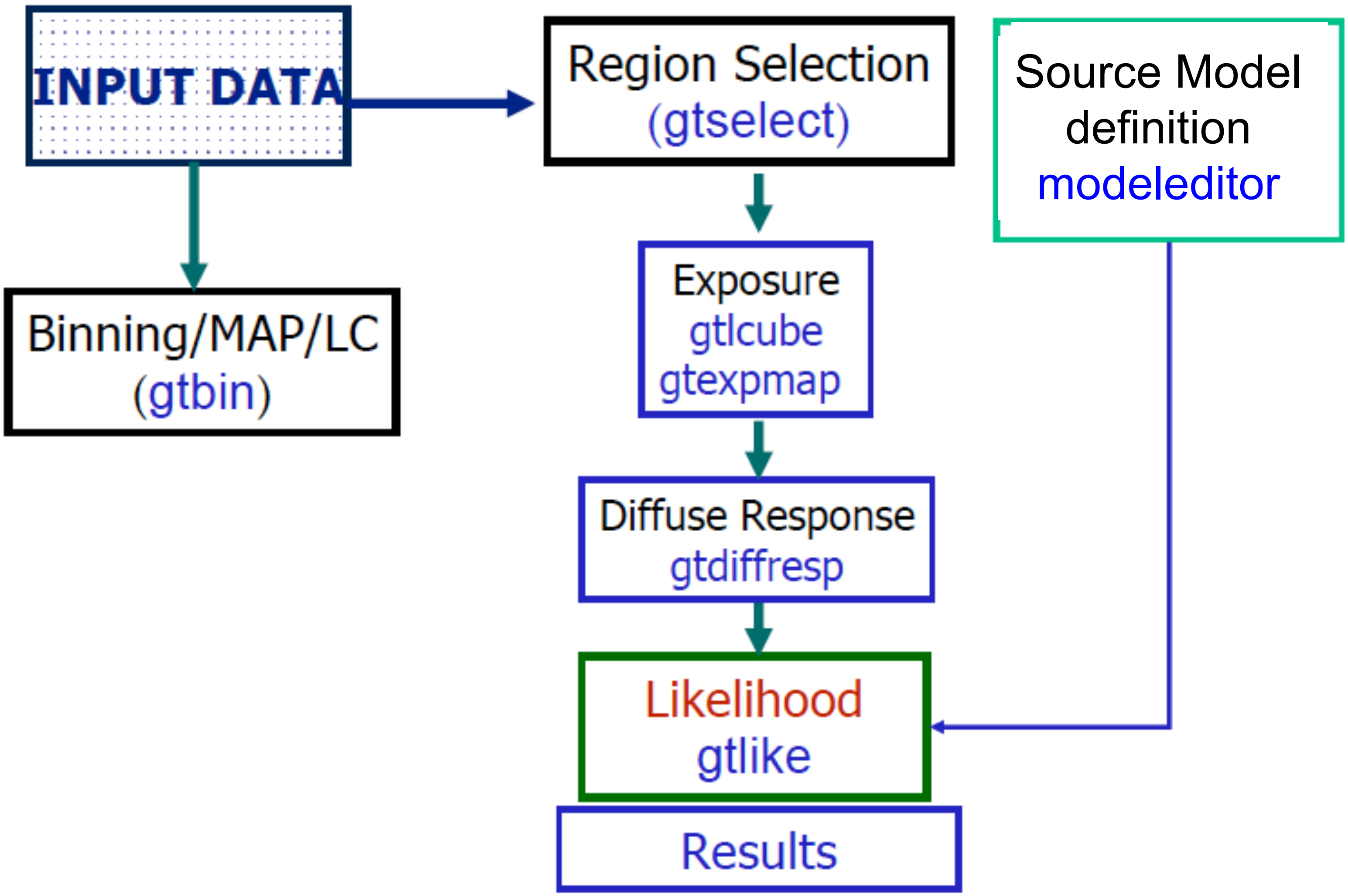
To save CPU time, a model-independent quantity, « exposure map (*cube*)» is precomputed:

$$\varepsilon(E, \hat{p}) \equiv \int_{\rm ROI} dE'\, d\hat{p}'\, dt\, R(E', \hat{p}', t; E, \hat{p})$$

Then

$$N_{\rm pred} = \int_{\rm SR} dE d\hat{p}\, S(E, \hat{p})\, \varepsilon(E, \hat{p})$$

Likelihood analysis in a nutshell



Details of the method can be found in
http://fermi.gsfc.nasa.gov/ssc/data/analysis/documentation/Cicerone

gtselect

- http://fermi.gsfc.nasa.gov/ssc/data/analysis/scitools/help/gtselect.txt
- Performs selection cuts in event data file, typically to define a Region-of-Interest (ROI) with events belonging to a certain class (standard is the « diffuse » class).
- gtselect infile outfile ra dec rad tmin tmax emin emax zmax
- Output fits file contains selected events. The header includes a record of the pruning history.
- Retrieving cuts applied to a given file can be obtained with gtvcut

USAGE gtvcut file_name EVENTS

Last 3 weeks in Jan

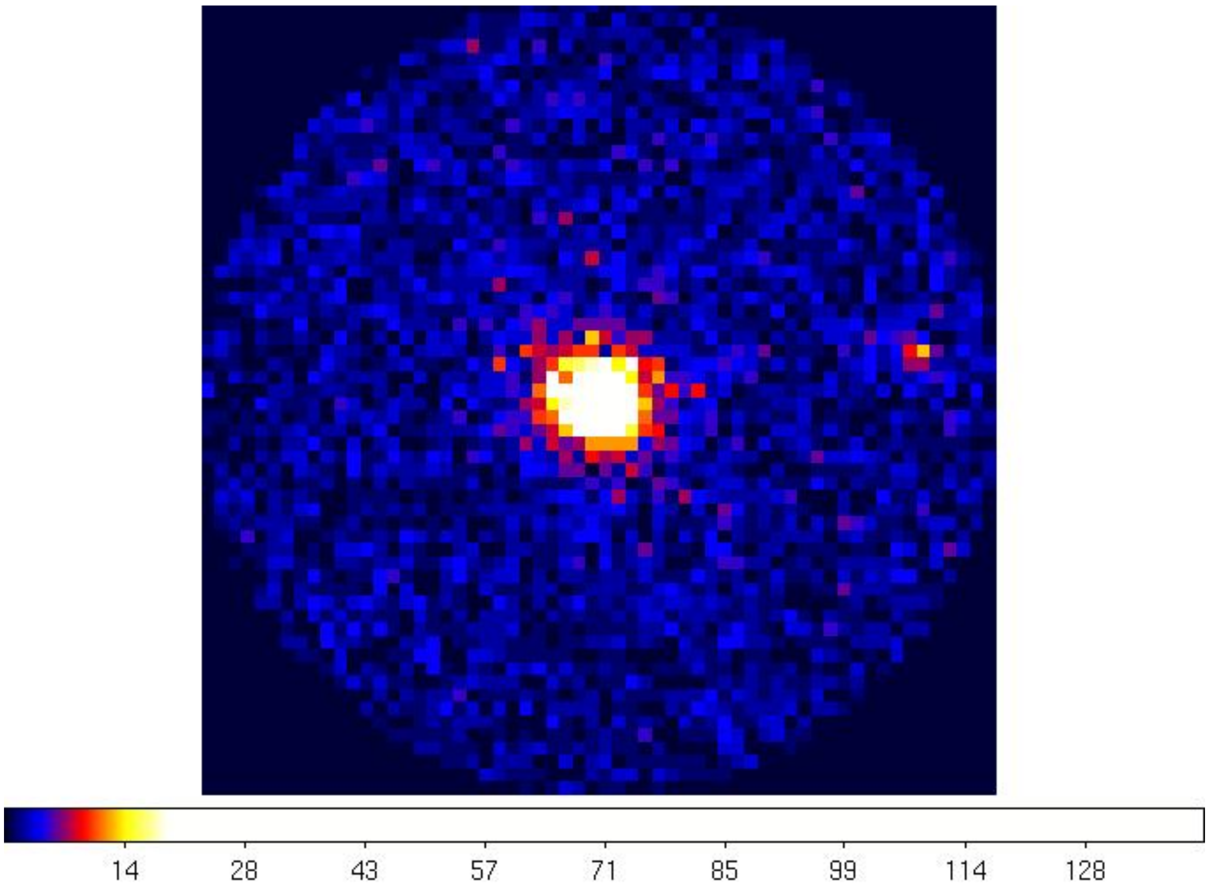
➢gtselect evclsmin=3 evclsmax=4 infile=3c454.3_long_PH00.fits outfile=3c454.3_long_15deg_sel.fits ra=343.490616 dec=16.148211 rad=15 tmin=284256002 tmax=286070400 emin=100 emax=300000 zmax=105

> gtvcut 3c454_long_sel.fits EVENTS

➢note: plist returns the parameter list for a given ftool

> plist gtselect

inspecting the ROI



Detected sources are listed in 1FGL Catalog:
http://fermi.gsfc.nasa.gov/ssc/data/access/lat/1yr_catalog/

gtmktime

- http://fermi.gsfc.nasa.gov/ssc/data/analysis/scitools/help/gtmktime.txt
- Create Good Time Intervals (GTIs) based on selections made using the spacecraft data file variables.
- gtmktime is used to update the Good Time Intervals (GTI) extension and make cuts based on spacecraft parameters contained in the Pointing and Livetime History (spacecraft) FITS file. A Good Time Interval is a time range when the data can be considered valid.
- USAGE gtmktime scfile filter roicut evfile outfile

```
> gtmktime scfile=3c454.3_long_SC00.fits filter="(DATA_QUAL==1)"
    roicut=yes evfile=3c454.3_long_sel.fits
    outfile=3c454.3_long_mktime.fits
```

gtltcube

- http://fermi.gsfc.nasa.gov/ssc/data/analysis/scitools/help/gtltcube.txt
- Calculates integrated livetime as a function of sky position and off-axis angle.
- Output file contains a array of integrated livetime as a function off-axis angle for different (healpix-based) sky positions
- USAGE gtltcube evfile scfile outfile dcostheta binsz

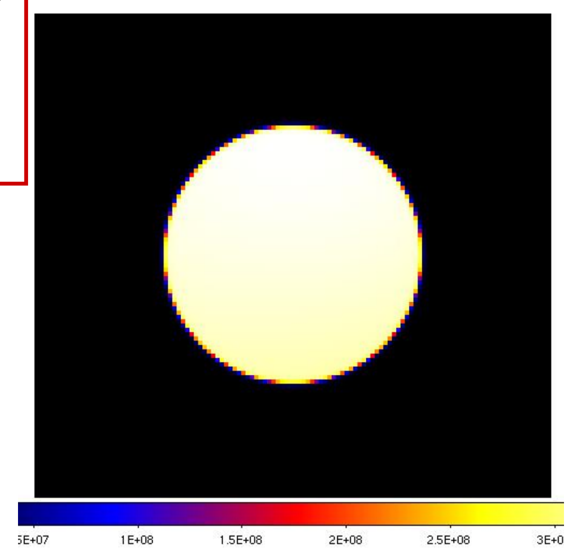
```
mcenery-2:amsterdam mcenery$ gtltcube
Event data file[3c454.3_Dec2009_mktime.fits] 3c454.3_long_15deg_mktime.fits
Spacecraft data file[3c454.3_Dec2009_SC00.fits] 3c454.3_long_SC00.fits
Output file[3c454.3_Dec2009_expCube.fits] 3c454.3_long_expCube.fits
Step size in cos(theta) (0.:1.) [0.025]
Pixel size (degrees)[1]
Working on file 3c454.3_long_SC00.fits
.....................!
mcenery-2:amsterdam mcenery$
```

gtexpmap

- http://fermi.gsfc.nasa.gov/ssc/data/analysis/scitools/help/gtexpmap.txt
- calculates ROI-specificexposure maps for unbinned likelihood analysis.
- creates exposure maps needed to compute the predicted number of photons within a given Region-of-Interest (ROI) for diffuse components in your source model.
- integral of the total response (effective area x energy dispersion x point spread function) over the entire ROI.
- output fits file contains a cube of nlong x nlat x nenergies exposure maps for the specified ROI
- USAGE gtexpmap evfile scfile expcube outfile irfs srcrad nlong nlat nenergies

«Source region » must exceeds the ROI radius by at least 10°

```
mcenery-2:amsterdam mcenery$ gtexpmap
The exposure maps generated by this tool are meant
to be used for *unbinned* likelihood analysis only.
Do not use them for binned analyses.
Event data file[3c454.3_long_15deg_mktime.fits] 3c454.3_long_15deg_mktime.fits
Spacecraft data file[3c454.3_long_SC00.fits] 3c454.3_long_SC00.fits
Exposure hypercube file[3c454.3_long_expCube.fits] 3c454.3_long_expCube.fits
output file name[3c454.3_long_expMap.fits] 3c454.3_long_expMap.fits
Response functions[P6_V3_DIFFUSE]
Radius of the source region (in degrees)[30]
Number of longitude points (2:1000) [120]
Number of latitude points (2:1000) [120]
Number of energies (2:100) [20]
Computing the ExposureMap using 3c454.3_long_expCube.fits
.....................!
mcenery-2:amsterdam mcenery$
```

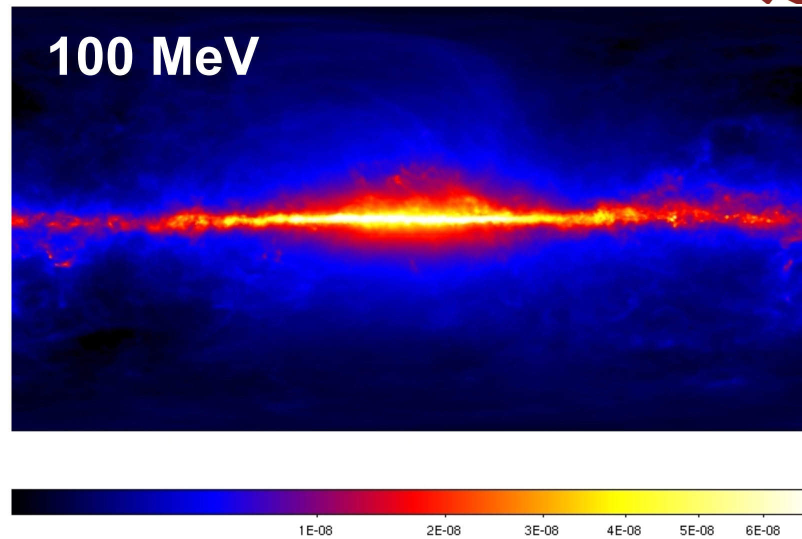


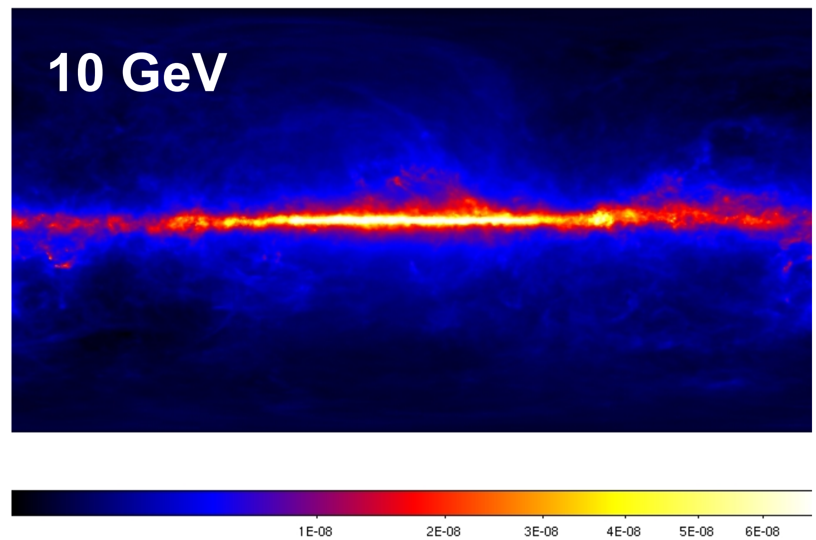
Diffuse models

- **Galactic diffuse model:**
  gll_iem_v02.fit
  model adjusted to data

- **Extragalactic diffuse model actually sum of true gamma-ray extragalactic diffuse+ instrumental background**
  isotropic_iem_v02.txt

39.3884 6.57144e-07 4.6946e-08
64.0414 4.09665e-07 5.72124e-09
104.125 1.72000e-07 8.35794e-10
169.296 6.60007e-08 2.15325e-10
275.257 2.24126e-08 7.58059e-11
447.539 7.21114e-09 2.95711e-11
727.651 2.20758e-09 1.16796e-11
1183.08 7.20365e-10 4.68072e-12
1923.57 2.35566e-10 1.93256e-12
3127.52 7.36933e-11 8.02165e-13
.....





Source models

Many different models are available:

http://fermi.gsfc.nasa.gov/ssc/data/analysis/documentation/Cicerone/Cicerone_Likelihood/Model_Selection.html

http://fermi.gsfc.nasa.gov/ssc/data/analysis/scitools/xml_model_defs.html

The units for the spectral models are cm^{-2} s^{-1} MeV^{-1} for point sources and cm^{-2} s^{-1} MeV^{-1} sr^{-1} for diffuse sources. All energies are in MeV.

PowerLaw simple power law

$N(E) = N_0 (E/E_0)^{\Gamma}$

N_0 :Prefactor, Γ:spectral index

E_0:energy scale

BrokenPowerLaw two component power law

$N(E) = N_0 (E/E_b)^{\Gamma 1+1}$ E<Eb

$= N_0 (E/Eb)^{\Gamma 2+1}$ E>Eb

N_0 :Prefactor Γ1:low energy spectral index Γ2:high energy spectral index

Eb:break energy

PowerLaw2 simple power law with the integral number of counts between two energies as the normalization

$N(E) = (\Gamma+1)N E^{\Gamma}/(Emax^{\Gamma+1} - Emin^{\Gamma+1})$

N: Integral number of counts between Emax and Emin Γ:spectral index

Emin:low end of energy range (always a fixed quantity)

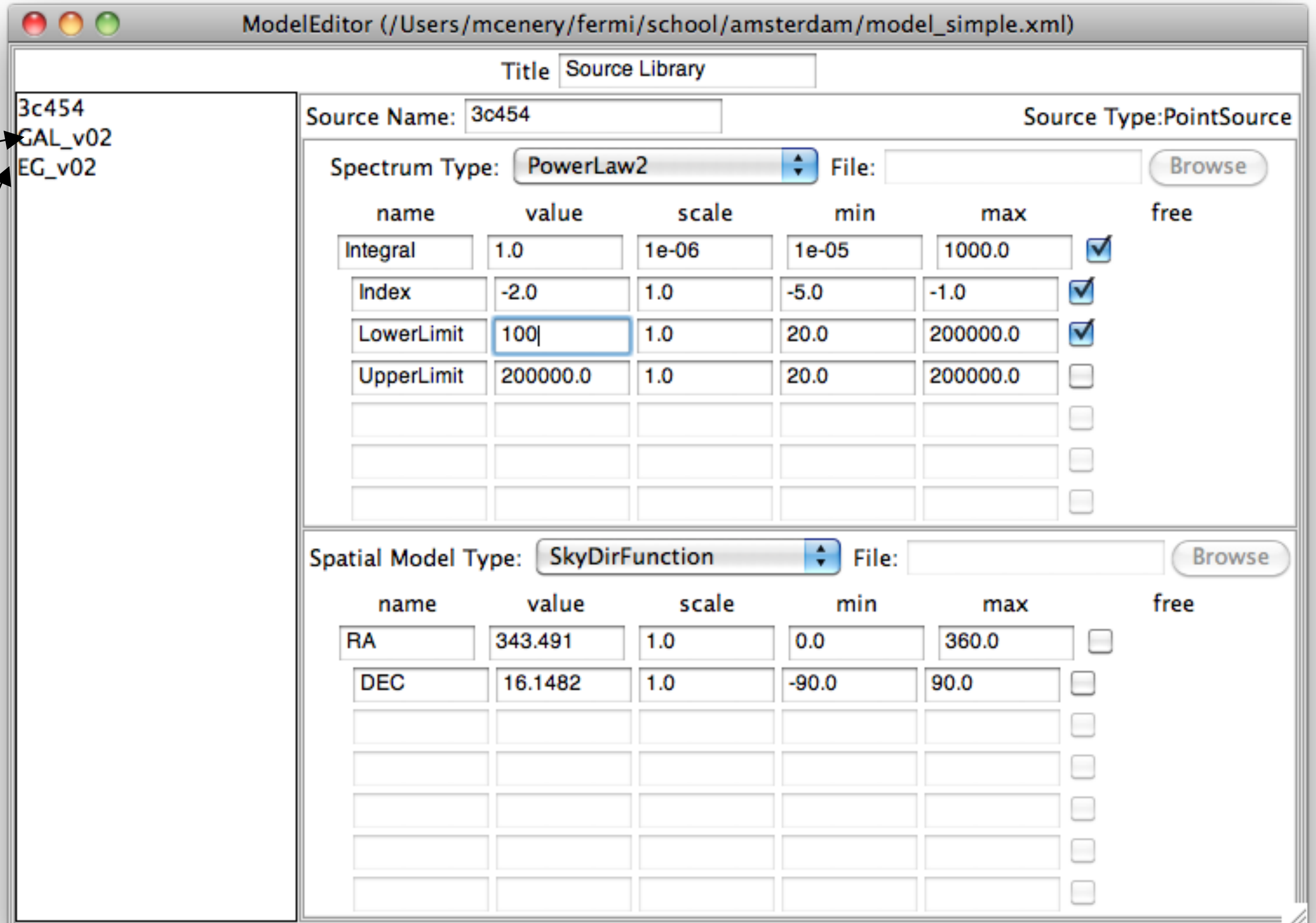
Emax:high end of energy range (always a fixed quantity)

……………..

Building a source model (1) - ModelEditor

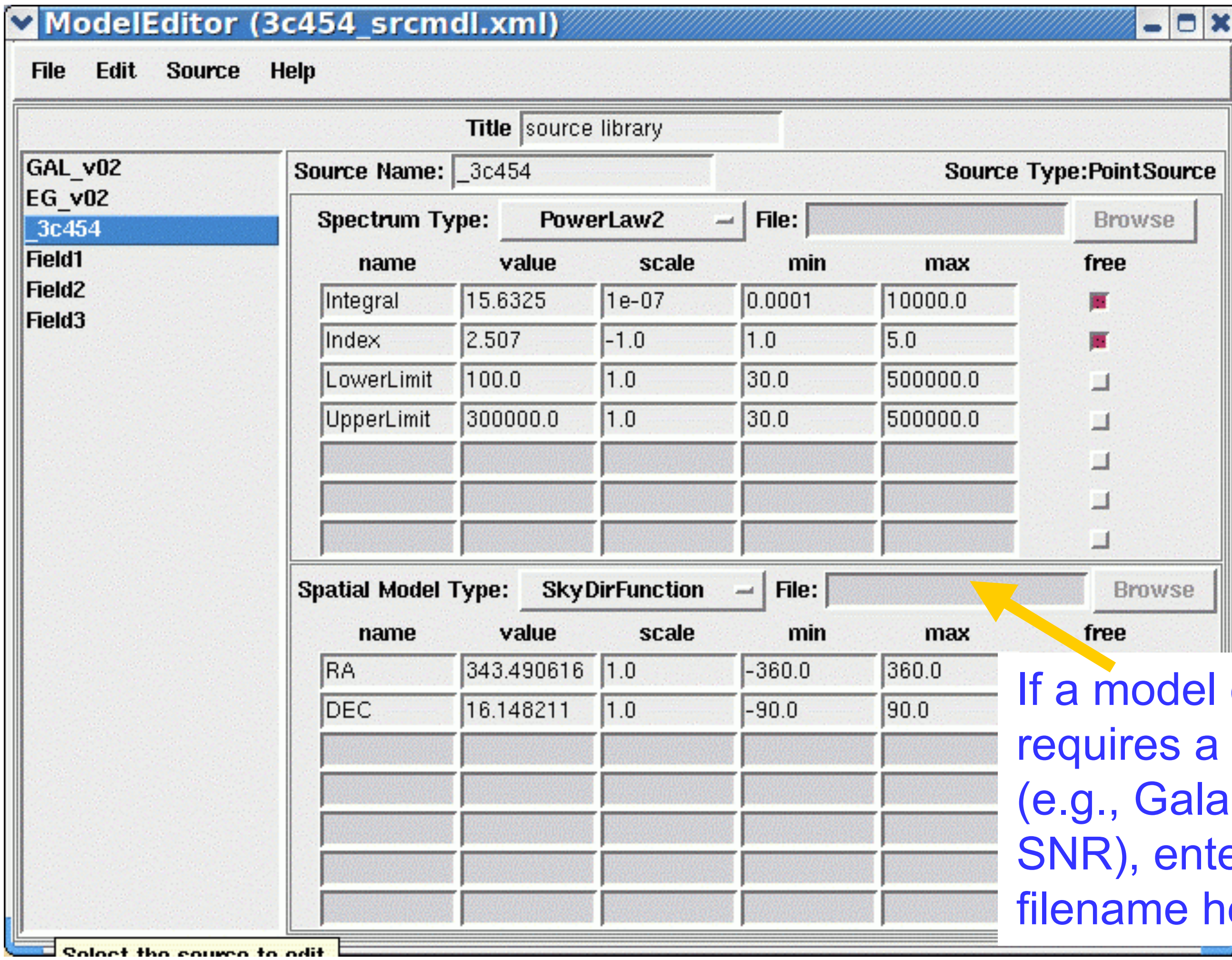
Add GALPROP source

Add extragalactic source (use filefunction spectral type with isotropic_iem_v02.txt)



preparing the source model (2)

Edit source name,default fit parameters, bounds, scaling, etc.



If a model component requires a FITS image (e.g., Galactic diffuse, SNR), enter the filename here

```
mcenery-2:amsterdam mcenery$ cat model.xml
<?xml version="1.0" ?>
<source_library title="source library">
<!-- Sources between (0.0,4.0] degrees of ROI center -->
<source name="_3c454.3" type="PointSource">
        <spectrum type="PowerLaw2">
        <!-- Source is 0.00423151829987 degrees away from ROI center -->
                <parameter free="1" max="1e4" min="1e-4" name="Integral" scale="1e-07" value="0.462116105382"/>
                <parameter free="1" max="5.0" min="0.0" name="Index" scale="-1.0" value="2.46619"/>
        <parameter free="0" max="5e5" min="30" name="LowerLimit" scale="1.0" value="1e3"/>
                <parameter free="0" max="5e5" min="30" name="UpperLimit" scale="1.0" value="1e5"/>
        </spectrum>
        <spatialModel type="SkyDirFunction">
                <parameter free="0" max="360.0" min="-360.0" name="RA" scale="1.0" value="343.495"/>
                <parameter free="0" max="90" min="-90" name="DEC" scale="1.0" value="16.1495"/>
        </spatialModel>
</source>
<!-- Diffuse Sources -->
<source name="gal_v02" type="DiffuseSource">
        <spectrum type="PowerLaw">
                <parameter free="1" max="10" min="0" name="Prefactor" scale="1" value="1"/>
                <parameter free="0" max="1" min="-1" name="Index" scale="1.0" value="0"/>
                <parameter free="0" max="2e2" min="5e1" name="Scale" scale="1.0" value="1e2"/>
        </spectrum>
        <spatialModel file="./gll_iem_v02.fit" type="MapCubeFunction">
                <parameter free="0" max="1e3" min="1e-3" name="Normalization" scale="1.0" value="1.0"/>
        </spatialModel>
</source>
<source name="eg_v02" type="DiffuseSource">
        <spectrum file="./isotropic_iem_v02.txt" type="FileFunction">
                <parameter free="1" max="10" min="1e-2" name="Normalization" scale="1" value="1"/>
        </spectrum>
        <spatialModel type="ConstantValue">
                <parameter free="0" max="10.0" min="0.0" name="Value" scale="1.0" value="1.0"/>
        </spatialModel>
</source>
</source_library>
```

More on Source Model

- For this example, start with a simple model including just 3c454.3 and the Galactic and isotropic backgrounds (this is reasonable because the source is so bright).
- More generally: data from a substantial spatial region around the source being studied must be used because of overlapping point spread functions of nearby sources. Thus it is common to include sources from a region larger than your ROI.
  - Try using a model which includes all 1FGL sources in the ROI+5 deg, and leave free parameters only for the brightest sources within 10 deg of the center of the FoV.
  - Easiest to do this using the python script *make1FGLxml.py*

```
mcenery-2:amsterdam mcenery$ python
Python 2.5.1 (r251:54863, Aug  8 2009, 12:04:48)
[GCC 4.0.1 (Apple Inc. build 5465)] on darwin
Type "help", "copyright", "credits" or "license" for more information.
>>> from make1FGLxml import *
This is make1FGLxml version 02.
>>> mymodel=srcList("gll_psc_v02.fit", "3c454.3_long_15deg_mktime.fits","model_ful
l.xml")
>>> mymodel.makeModel('gll_iem_v02.fit','gal_v02','isotropic_iem_v02.txt','eg_v02'
)
Creating file and adding sources
>>> quit()
mcenery-2:amsterdam mcenery$ emacs -nw model_full.xml
```

gtdiffrsp

- http://fermi.gsfc.nasa.gov/ssc/data/analysis/scitools/help/gtdiffrsp.txt
- **Calculates the integral over solid angle of a diffuse source model convolved with the instrumental response function**
- **USAGE gtdiffrsp evfile scfile srcmdl irfs**
- **The instrument response can be precomputed for each diffuse model component. The gtdiffrsp tool will perform these integrations and add the results as an additional colum for each diffuse source into the input event file.**
- **This step can be skipped if you use standard data files, which include diffuse responses. The diffuse component names in your model and the FT1 file have to match.**

gtlike

- http://fermi.gsfc.nasa.gov/ssc/data/analysis/scitools/help/gtlike.txt
- **Performs unbinned (or binned) likelihood analysis of LAT data.**
- **USAGE gtlike irfs expcube srcmdl statistic optimizer evfile scfile expmap cmap bexpmap**

```
plist gtlike
Parameters for /home/local1/pfiles/gtlike.par
        irfs = P6_V3_DIFFUSE    Response functions to use
      (edisp = no)             Use energy dispersion?
     expcube = 3c454_expcube.fits Exposure hypercube file
      srcmdl = 3c454_srcmdl_1.xml Source model file
      (sfile = none)           Source model output file
  (check_fit = yes)            Issue warnings regarding fit?
    (results = results.dat)    Output file for fit results
   (specfile = counts_spectra.fits) Output file for counts spectra
   statistic = UNBINNED        Statistic to use
   optimizer = MINUIT          Optimizer
       (ftol = 1e-2)           Fit tolerance
    (toltype = ABS)            Fit tolerance convergence type (absolute vs relative)
      (tsmin = no)             Re-optimize for TS fits?
       (save = yes)            Write output files?
      (refit = no)             Allow for refitting?
      evfile = 3c454_100_300000_evt02.fits Event file
    (evtable = EVENTS)         Event table extension
      scfile = L090923112502E0D2F37E71_SC00.fits Spacecraft file
    (sctable = SC_DATA)        Spacecraft table extension
      expmap = 3c454_expmap.fits Unbinned exposure map
       (plot = no)             Plot unbinned counts spectra?
        cmap = none            Counts map file
     bexpmap = none            Binned exposure map
    (psfcorr = yes)            apply psf integral corrections
    (chatter = 2)              Output verbosity
    (clobber = yes)            Overwrite existing output files
      (debug = no)             Activate debugging mode
        (gui = no)             GUI mode activated
       (mode = ql)             Mode of automatic parameters
```

gtlike

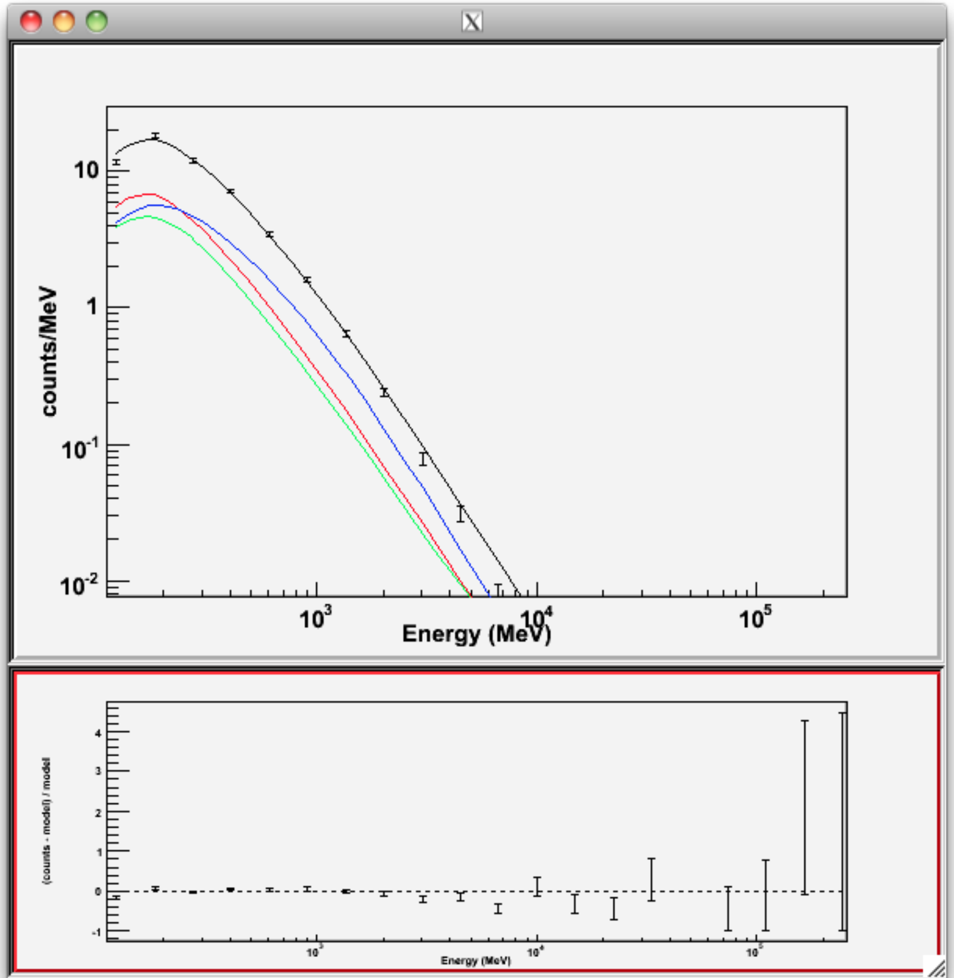
```
mcenery-2:amsterdam mcenery$ gtlike
Statistic to use (BINNED|UNBINNED) [UNBINNED]
Spacecraft file[3c454.3_long_SC00.fits]
Event file[3c454.3_long_15deg_mktime.fits]
Unbinned exposure map[3c454.3_long_expMap.fits]
Exposure hypercube file[3c454.3_long_expCube.fits]
Source model file[model.xml]
Response functions to use[P6_V3_DIFFUSE]
Optimizer (DRMNFB|NEWMINUIT|MINUIT|DRMNGB|LBFGS) [MINUIT]

 **********
 **    1 **SET PRINT     .000
 **********
 **********
 **    2 **SET NOWARN
 **********

 PARAMETER DEFINITIONS:
    NO.    NAME         VALUE      STEP SIZE      LIMITS
     1 'Integral  '     .46212       1.0000        .10000E-03    10000.
     2 'Index     '     2.4662       1.0000         .0000        5.0000
     3 'Normalizat'     1.0000       1.0000        .10000E-01    10.000
     4 'Prefactor '     1.0000       1.0000         .0000        10.000
 **********
 **    3 **SET ERR     .5000
 **********
 **********
 **    4 **SET GRAD     1.000
 **********
 **********
 **    5 **MINIMIZE    1200.        20.00
 **********
```

```
_3c454.3:
Integral: 1.60391 +/- 0.0793391
Index: 2.40337 +/- 0.0261108
LowerLimit: 1000
UpperLimit: 100000
Npred: 2240.84
ROI distance: 0.00440388
TS value: 7985.12

eg_v02:
Normalization: 0.776667 +/- 0.108149
Npred: 1658.13

gal_v02:
Prefactor: 1.20277 +/- 0.10194
Index: 0
Scale: 100
Npred: 2712.04
WARNING: Fit may be bad in range [100, 222.696] (MeV)
WARNING: Fit may be bad in range [2459.51, 3670.33] (MeV)
WARNING: Fit may be bad in range [5477.23, 8173.66] (MeV)
WARNING: Fit may be bad in range [40536, 60491.9] (MeV)

Total number of observed counts: 6611
Total number of model events: 6611

-log(Likelihood): 70764.06969
```

Checking the fit quality

> gtlike plot=yes

Counts vs model predictions and residuals *for the whole ROI*



Exploiting the results: output files

Results are stored:

- **in an ascii file (default name: results.dat)**
- **in a fits.file (default name: counts_spectra.fits**
- **in an xml file (optional)**


```
{'_3c454.3': {'Integral': '1.60391 +/- 0.0793391',
'Index': '2.40337 +/- 0.0261108',
'LowerLimit': '1000',
'UpperLimit': '100000',
'Npred': '2240.84',
'ROI distance': '0.00440388',
'TS value': '7985.12',
},
'eg_v02': {'Normalization': '0.776667 +/- 0.108149',
'Npred': '1658.13',
},
'gal_v02': {'Prefactor': '1.20277 +/- 0.10194',
'Index': '0',
'Scale': '100',
'Npred': '2712.04',
},
}
```

using python

```
from UnbinnedAnalysis import *
my_obs=UnbinnedObs('3c454_long_mktime.fits',
                scFile='3c454_long_SC00.fits',
                expMap='3c454_expmap.fits',
                expCube='3c454_expcube.fits',
                irfs='P6_V3_DIFFUSE')
analysis= UnbinnedAnalysis (my_obs,'3c454_srcmdl.xml',optimizer='MINUIT')
print analysis
dir(analysis)
loglike=analysis.fit(covar=True)
print loglike
cov=analysis.covariance
print cov
analysis.plot()
analysis.model
print analysis['_3c454']
analysis.writeCountsSpectra("spectra.fits")
analysis.writeXml("results.xml")
analysis.sourceNames()
ts=analysis.Ts('_3c454')
npred=analysis.logLike.NpredValue('_3c454')
```

Comparing models

Likelihood Ratio Test (LRT)

Enable comparison of two nested models (the more complex one derives from the other by adding Δn more parameters)

ex Power law (2 parameters) vs Broken Power Law (4 parameters): Δn=2

Under regularity assumptions, the probability distribution of the test statistic TS is asymptotically a chi-square with Δn degrees of freedom.

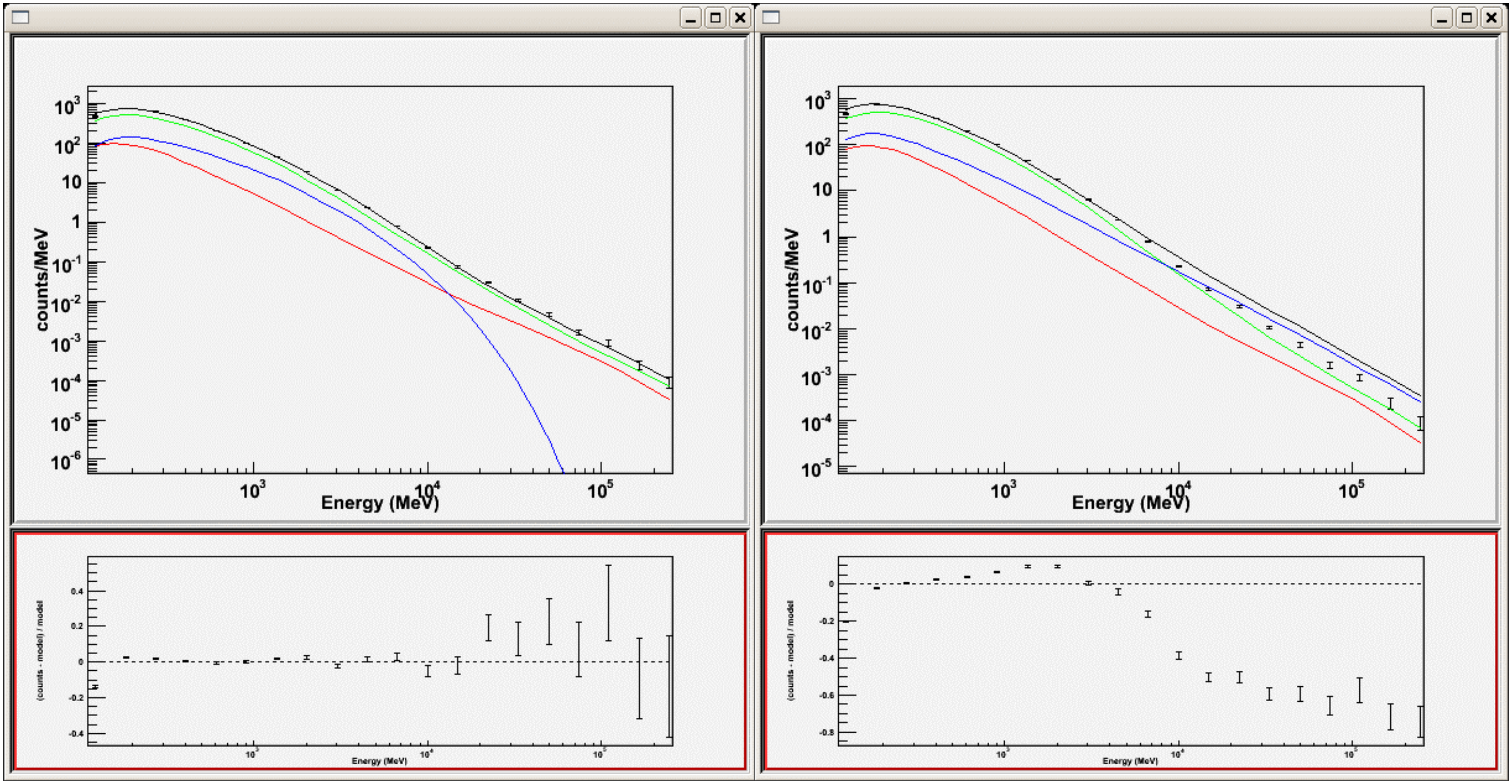
Ex: Vela
PLSuperExpcutoff: -log(Likelihood): 3321842.126
PowerLaw : -log(Likelihood): 3324727.121

Ex: Vela

PLSuperExpcutoff: -log(Likelihood): 3321842.126

PowerLaw : -log(Likelihood): 3324727.121

LRT= 2 Δlog(Likelihood)=5770 « p-value » from a chi-square with 2 dof



- **PLSuperExpCutoff** (see example XML Model Definition) For modeling pulsars.

$$\frac{dN}{dE} = N_0 \left(\frac{E}{E_0}\right)^{\gamma_1} \exp\left(-\left(\frac{E}{E_c}\right)^{\gamma_2}\right)$$

  where

  - Prefactor = N_0
  - Index1 = γ_1
  - Scale = E_0
  - Cutoff = E_c
  - Index2 = γ_2

- **PowerLaw** (see example XML Model Definition) This function has the form

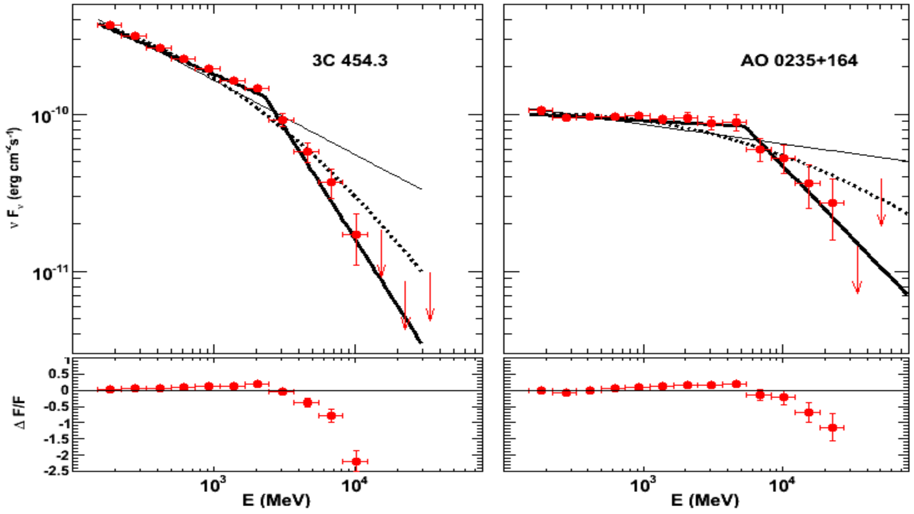
$$\frac{dN}{dE} = N_0 \left(\frac{E}{E_0}\right)^{\gamma}$$

  where the parameters in the XML definition have the following mappings:

  - Prefactor = N_0
  - Index = γ
  - Scale = E_0

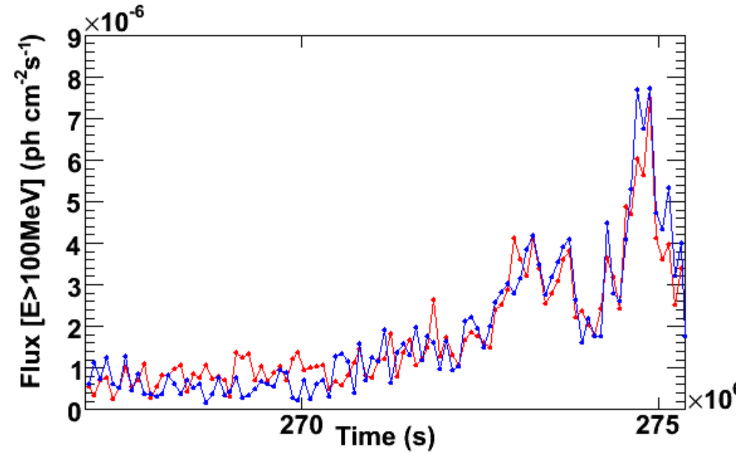
Spectral analysis in bands

- Standard analysis requires a model
- If enough statistics, the analysis can be performed independently in different energy bands. This assumes that the energy redistribution due to a finite resolution is negligible. That's essentially justified for Fermi.
- Use gtselect to define the bands.
- Beware of the TS and number of photons per bin.



Producing light curves

- Scripting is required:
  - determine required time binning
  - bin data in time with gtselect
  - run the analysis
  - parse the output files



- Beware of the number of degrees of freedom left free! You may consider holding some parameters fixed
  - e.g, spectal index,
  - parameters of steady/or faint sources

- Monitor the source TS, compute upper limit if necessary (TS<TSthresh)

- Using F(E>E0)

Optimal results are obtained with E0 being the « pivot energy ».
Typically the pivot energy is 200-300 MeV.