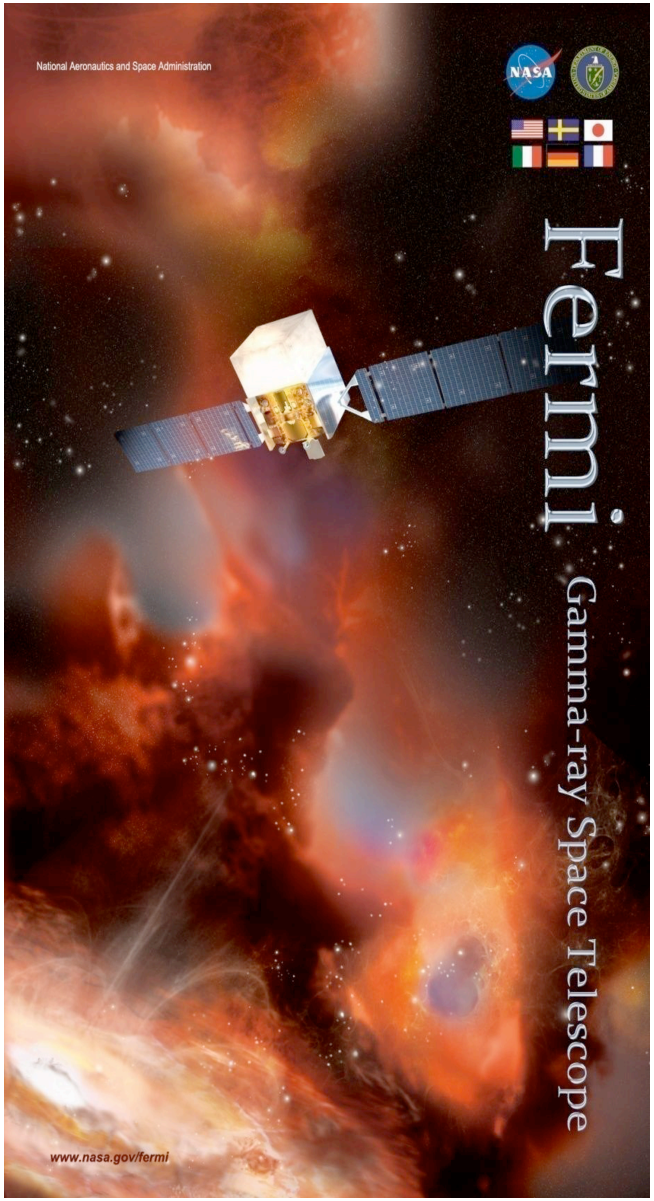

National Aeronautics and Space Administration
NASA
Fermi
Gamma-ray Space Telescope
www.nasa.gov/fermi

Spectral analysis of Fermi-LAT data

Andrea Tramacere

ISDC

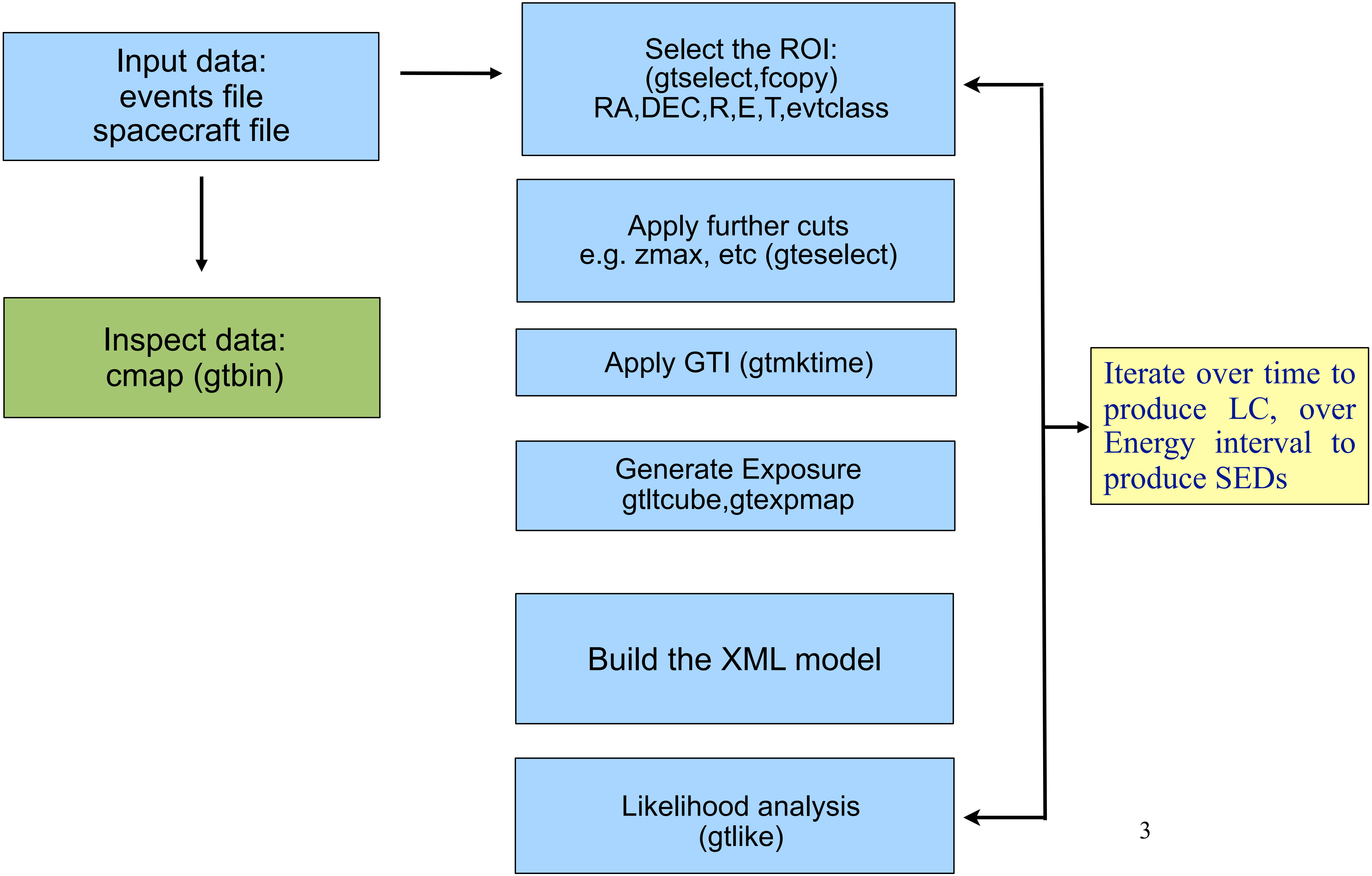
Saas Fee - Fermi data analysis session

Outline

- Generic overview
- Basics of Likelihood analysis
- Running the tools:
  1. define the region model and generate exposure
  2. run gtlike
  3. use gtlike through the python interface
- Advanced topics:
  1. Build SEDs
  2. possible biases
  3. inspect the parameter space

- http://fermi.gsfc.nasa.gov/ssc/data/analysis/documentation/Cicerone/
- http://fermi.gsfc.nasa.gov/ssc/data/analysis/scitools/references.html
- http://fermi.gsfc.nasa.gov/ssc/data/analysis/scitools/references.html

Schematic view of the process



Input data:
events file
spacecraft file

Inspect data:
cmap (gtbin)

Select the ROI:
(gtselect,fcopy)
RA,DEC,R,E,T,evtclass

Apply further cuts
e.g. zmax, etc (gteselect)

Apply GTI (gtmktime)

Generate Exposure
gtltcube,gtexpmap

Build the XML model

Likelihood analysis
(gtlike)

Iterate over time to produce LC, over Energy interval to produce SEDs

Likelihood analysis general prescription and prerequisites:

- Spacecraft file

- Events file with with cuts applied (gtselect:evt class, Energy range, Time span, zenith angle)

- GTI generated accordingly to cuts (zangle, rocking angle, etc) by gtmktime

  Indeed cuts can be more complex than just zenith angles, and GTI must always be generated accordingly in order to avoid biased fluxes:

  - **"IN_SAA!=T && LIVETIME>0 && angsep(RA_ZENITH,DEC_ZENITH,RA_SCZ,DEC_SCZ)<43.0"**
  - **"SC_DATA" filter="(IN_SAA!=T) &&(DATA_QUAL==1) && (LIVETIME>0) && ((ABS(ROCK_ANGLE)<52 && START > 273628805) || (ABS(ROCK_ANGLE)<43 && START < 273628805))"**

- A count map (gtbin:) to inspect the sourece region (SR) and the region of interest (ORI).

Likelihood analysis general prescription and prerequisites: selecting ROI and SR

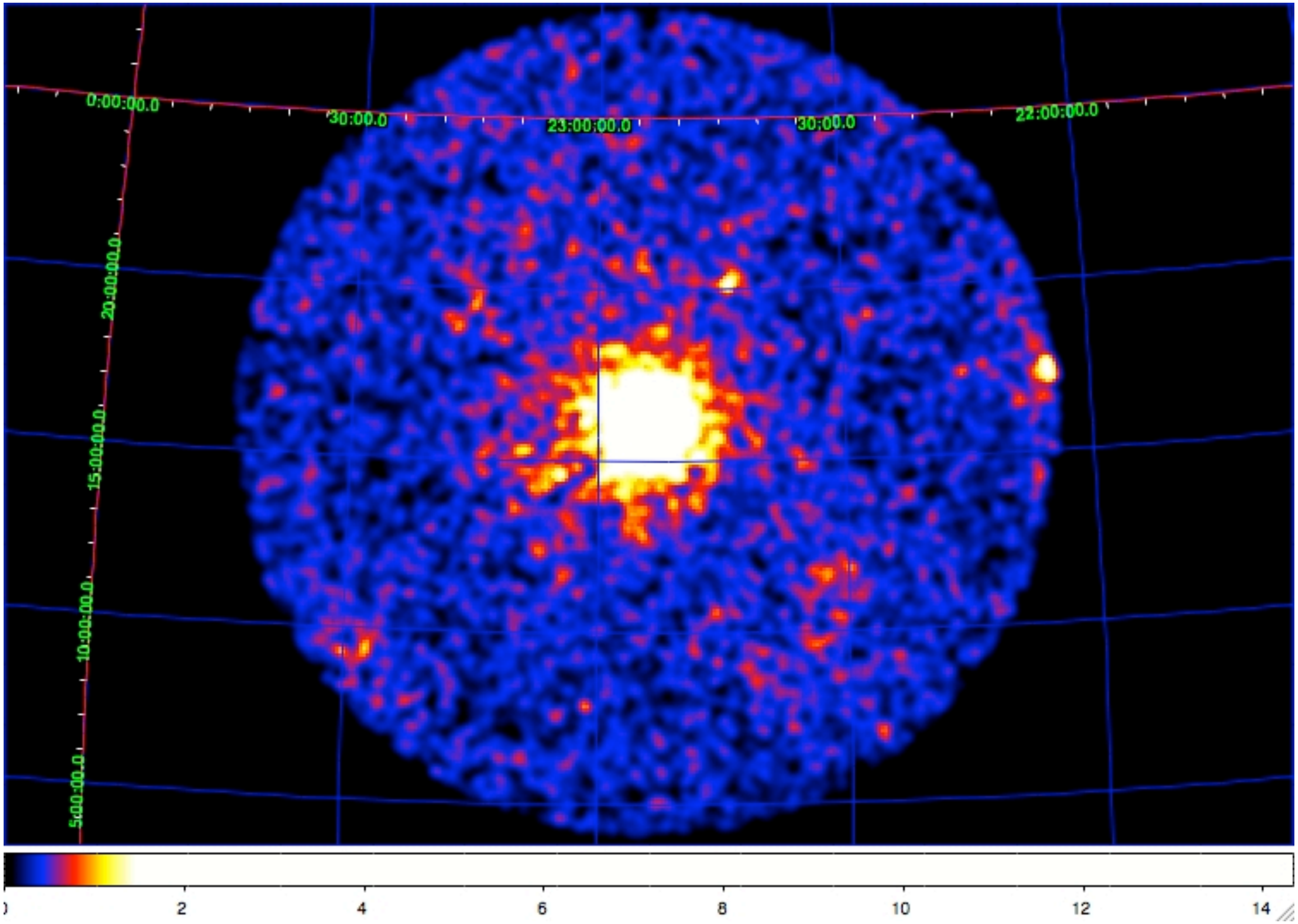
•The data from a substantial spatial region around the source(s) being analyzed must be used because of the overlapping of the point spread functions of nearby sources. The influence of sources a very great distance away from your source will be greatly attenuated (i.e., at 100 MeV, 68% of the counts will be within 3.5 degrees due to the large point spread function at low energies). **Thus, it is recommended that you include sources from a large 'Source Region' (SR), and counts from a smaller 'Region of Interest' (ROI)**. (gtselect)

•All sources in the source region will be used, and the size of the Source Region appropriate for your needs is determined by your experience and prior experimentation. In the meantime, the recommended default values are **ROI+10 and ROI+5** degrees for sources dominated by **~100 MeV and ~1 GeV** events, respectively.

•All counts in the ROI will also be included and you will also determine the appropriate ROI size based on your experience and prior experimentation. **In the meantime, the recommended default values are 15 and 10 degrees for sources dominated by ~100 MeV and ~1 GeV events, respectively**.

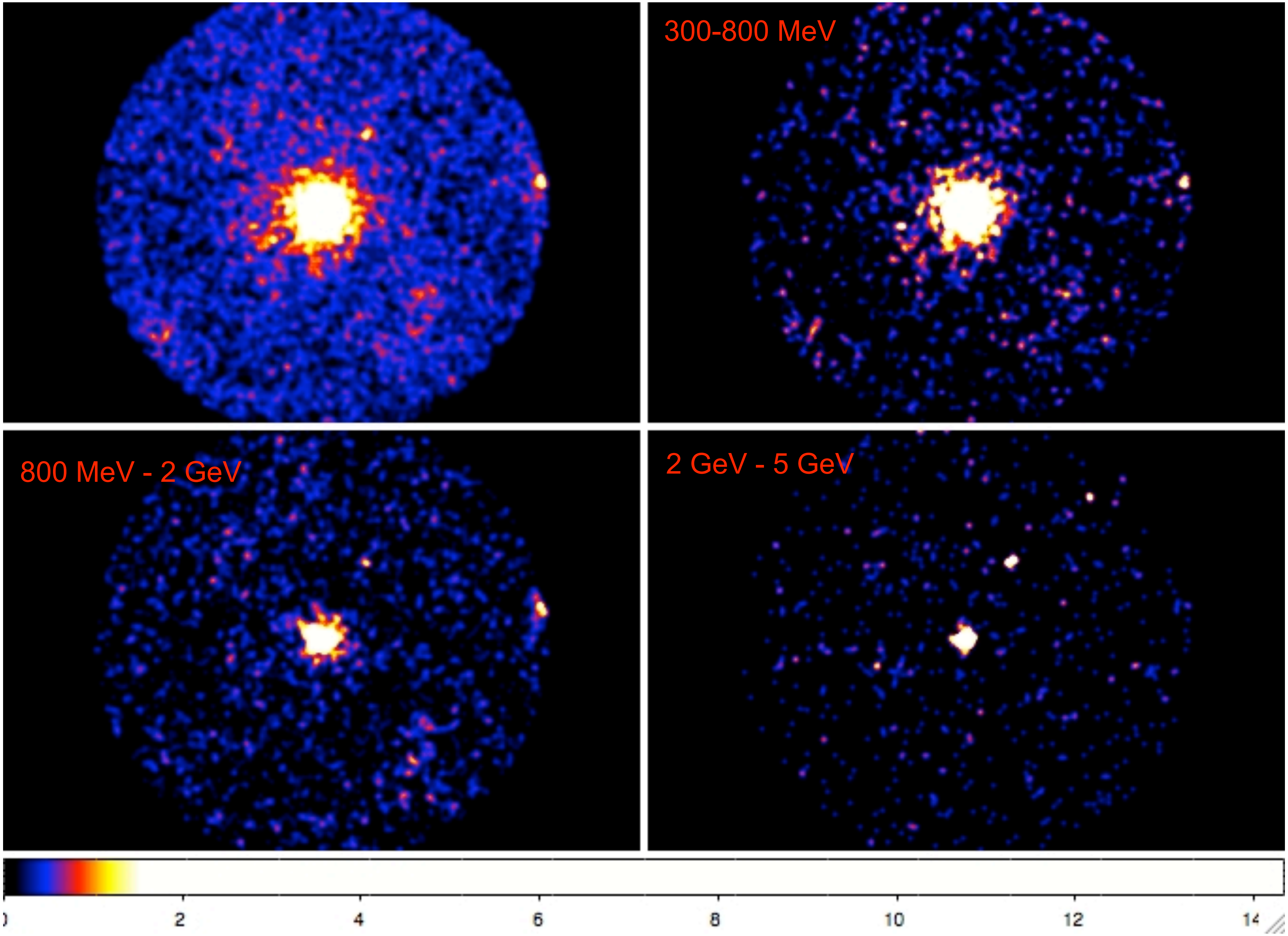
Inspect CMAPs

- Inspect the CMAP of the SR, to decide the sources to use your region model.
- How do you decide if is source is there? (see later......)



Inspect CMAPs

- Compare the full energy CMAP with others at different energy bins



Basics of likelihood analysis

Likelihood analysis

Given a data set D and a parametrized model M(x), the likelihood of the model is simply the probability of obtaining the data set if the model is true:

$$\mathcal{L}(M(\mathbf{x})) = P(D|M(\mathbf{x}))$$

Fora data binned in energy and direction the functional form of the likelihood reads:

$$\mathcal{L} = \prod_j \frac{\theta_j^{n_j} e^{-\theta_j}}{n_j!}$$

where ***n***n_j are the observed counts and θ_j are the counts predicted by the model.
The log-likelihood can be expressed in the following form:

$$\log \mathcal{L} \quad = \quad \sum_j n_j \log \theta_j - N_{\mathrm{pred}}$$

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

$$\log \mathcal{L} \quad = \quad \sum_j \log \theta_j - N_{\mathrm{pred}}$$

Likelihood analysis

In the Unbinned version of MLA the source Model considered is:

$$S(E,\hat{p},t) = \underbrace{\sum_i s_i(E,t)\delta(\hat{p}-\hat{p}_i)}_{\text{Point Sources}} + \underbrace{S_G(E,\hat{p}) + S_{\rm eg}(E,\hat{p})}_{\text{Galactic \& EG Diffuse Sources}} + \underbrace{\sum_l S_l(E,\hat{p},t)}_{\text{Other Sources}},$$

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

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

The IRFs are factored in terms of effective area (*A*), point spread function (*P*), and energy dispersion (*D*)

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

The integral is performed over the Source Region, i.e. the sky region encompassing all sources contributing to the Region-of – Interest (ROI).

Likelihood analysis

The function that is maximized is

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

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

Likelihood analysis: Exposure

Likelihood analysis considers the counts in the ROI resulting from all sources in a SR Interest.

The exposure map must extend over the entire Source Region, and is specific to the Region of Interest. **The exposure map is the total exposure – area multiplied by time – for a given position on the sky**, producing counts in the Region of Interest. Since the response function is a function of the photon energy, the exposure map is also a function of this energy.

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

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

To reduce the CPU time, a quantity, independent from any source model and similar to an exposure map, is precomputed

This exposure function can then be used to compute the expected numbers of events from a given source (N_{pred}) where $S_i(E,p)$ is the photon intensity from source i. Since the exposure calculation involves an integral over the ROI, separate exposure maps must be made for every distinct set of DSS cuts if, for example, one wants to subdivide an observation to look for secular flux variations from a particular source or sources.

The exposure needs a live time cube that is a HealPix table covering the full sky of the integrated **livetime as a function of inclination with respect to the LAT z-axis.**

The Test Statistic:

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 the source ('null hypothesis') and $\mathcal{L}_{\max,1}$ is the maximum likelihood value for a model with the additional source at a specified location.

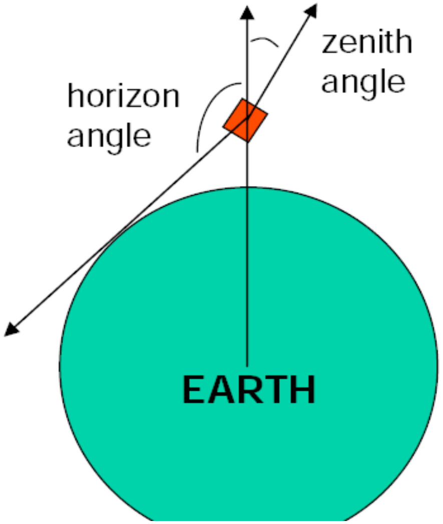
In the limit of a large number of counts, Wilk's Theorem states that the TS for the null hypothesis is asymptotically distributed as "chi square" with the degrees of freedom equal to the number of parameters characterizing the additional source.

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: running the tools

Livetime cube: gtmktime, gtltcube

Photons coming from the Earth limb are strongly anisotropic. Currently the only way of dealing with them is to get rid of them from the dataset with a zenith-angle cut. To apply this cut we used gtselect (prev. tutorial), specifying the maximum allowed angle, gtselect adds the correct DSS keywords in the output events file. The value of 105 degrees is the one most commonly used.



ZENITH_ANGLE < 105 (**After the cut above one has to correctly calculate the exposure**.To deal with the cut on zenith-angle we must select the correct GTIs using **gtmktime**. The easy way of doing this is to **exclude the time intervals where the zenith cut intersects the ROI from the list of GTIs**. Do like this if you are studying a narrow ROI (let's say with a radius of less than 20 degrees). You have to run gtmktime specifying the answer "yes" at

Apply ROI-based zenith angle cut[yes] yes

Livetime cube: gtmktime, gtltcube

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.

```
> gtmktime
Spacecraft data file[] L090923112502E0D2F37E71_SC00.fits
Filter expression[IN_SAA!=T] "(IN_SAA!=T) && (DATA_QUAL==1)"
Apply ROI-based zenith angle cut[yes] yes
Event data file[] 3c454_100_300000_evt01.fits
Output event file name[] 3c454_100_300000_evt02.fits
```

•Calculates integrated livetime as a function of sky position and offaxis angle.
• Output file contains a array of integrated livetime as a function off-axis angle for different (healpix-based) sky positions

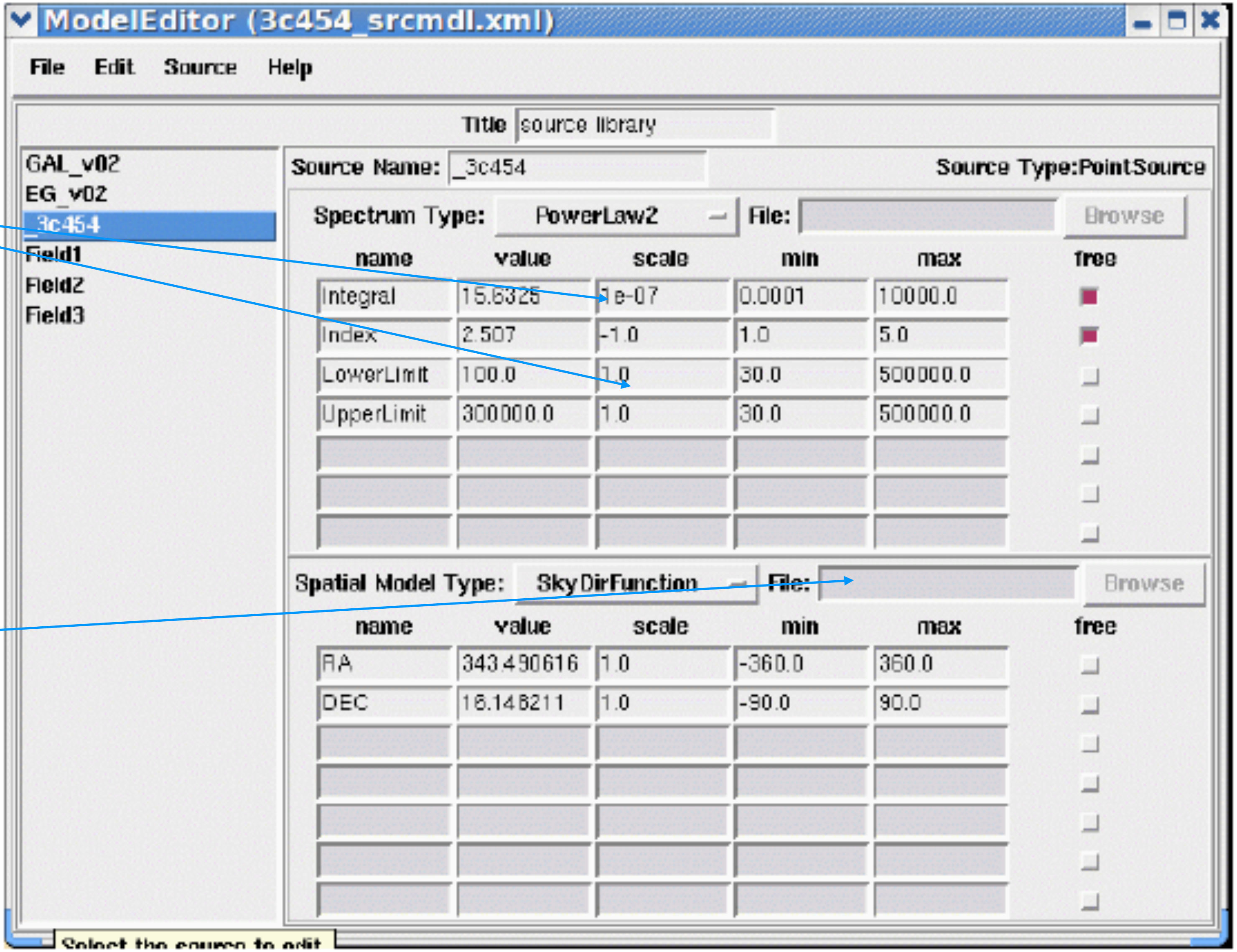
```
> gtltcube
Event data file[] 3c454_100_300000_evt02.fits
Spacecraft data file[] L090923112502E0D2F37E71_SC00.fits
Output file[] 3c454_100_300000_ExpCube.fits
Step size in cos(theta) (0.:1.) [0.025]
Pixel size (degrees)[1]
```

Build a region XMLMODEL:

1) Use the GUI model editor

Add sources, both pointlike and diffuse. Check all the parameters

If a diffuse components require a FITS image, provide the file mane



Title: source library

GAL_v02
EG_v02
_3c454
Field1
Field2
Field3

Source Name: _3c454

Source Type:PointSource

Spectrum Type: PowerLaw2

File:

name	value	scale	min	max	free
Integral	15.6325	1e-07	0.0001	10000.0	■
Index	2.507	-1.0	1.0	5.0	■
LowerLimit	100.0	1.0	30.0	500000.0	□
UpperLimit	300000.0	1.0	30.0	500000.0	□
					□
					□
					□

Spatial Model Type: SkyDirFunction

File:

name	value	scale	min	max	free
RA	343.490616	1.0	-360.0	360.0	□
DEC	16.148211	1.0	-90.0	90.0	□
					□
					□
					□
					□
					□
					□

Select the source to edit

Build a region XMLMODEL:

2) Use a python script and the Fermi catalog

```
do_regmodel.py ../DataSet/3c454_100_300000_evt02.fits /Users/orion/astro/GLAST/
Catalogs/gll_psc_v01.fit 3C454_3_script.xml 20 1FGLJ2253.9+1608 PL2 4e-9
```

This script requires:

- roi file
- lat catalog file
- source region radius (deg)
- name of the source in the lat catalog without spaces
- model for the source (PL2,PL,BPL,BLP2,LP)
- the lower limit on the flux of the source to put in the model
- you have to provide the exact path and name of diffuse models files either on the command line or editing these lines of code:

```
GD_Name='GAL_v02'
GD_File='/Users/orion/astro/GLAST/SaasFee/BkgModels/gll_iem_v02.fit'

ISO_Name='EG_v02'
ISO_File='/Users/orion/astro/GLAST/SaasFee/BkgModels/isotropic_iem_v02.txt'
```

Exposure: gtexpmap

calculates ROI-specificexposure maps for unbinned likelihood analysis.

- **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**

```
$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_100_300000_evt02.fits
Spacecraft data file[]  L090923112502E0D2F37E71_SC00.fits
Exposure hypercube file[] 3c454_100_300000_ExpCube.fits
output file name[] 3c454_100_300000_ExpMap.fits
Response functions[P6_V3_DIFFUSE]
Radius of the source region (in degrees)[30] 25
Number of longitude points (2:1000) [120]
Number of latitude points (2:1000) [120]
Number of energies (2:100) [20]
```

Irfs must be consistent with the evt. class selection !!!

Diffuse response: gtdiffrsp

The "diffuse response" is the proportional to the probability density that an event *j* arises from a source *i* with incident spectrum and intrinsic angular distribution $S_i(e,p)$

$$\begin{aligned} r_{ij} &\equiv \int d\hat{p}\, S_i(\varepsilon,\hat{p}) A(\varepsilon,\hat{p}) P(\hat{p}'_j;\varepsilon,\hat{p}) D(\varepsilon';\varepsilon,\hat{p}) \\ &\simeq \int d\hat{p}\, S_i({\varepsilon'}_j,\hat{p}) A(\varepsilon,\hat{p}) P(\hat{p}'_j;{\varepsilon'}_j,\hat{p}) \end{aligned}$$

- Here ε and $\hat{p}$ are the true photon energy and direction, respectively; and primes indicate the corresponding measured quantities. *A, P* and *D* are the effective area, point spread function and energy dispersion.

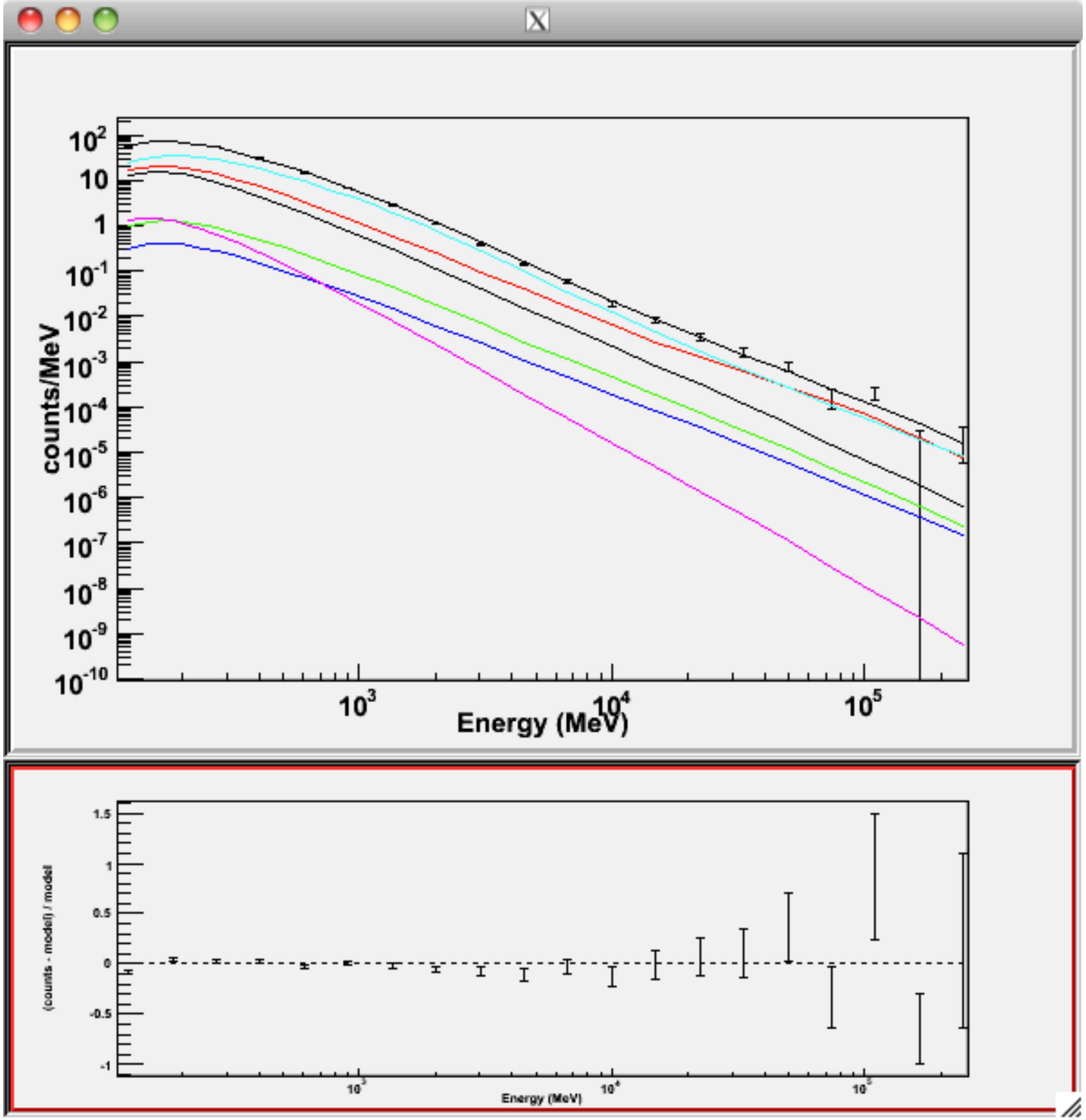
- The second line is the calculation performed by **gtdiffrsp**, to calculates the integral over solid angle of a diffuse source model convolved with the instrumental response function we use

- This step can be skipped if you use standard data files, which include diffuse responses. The diffuse component names in your model and the events file have to match.

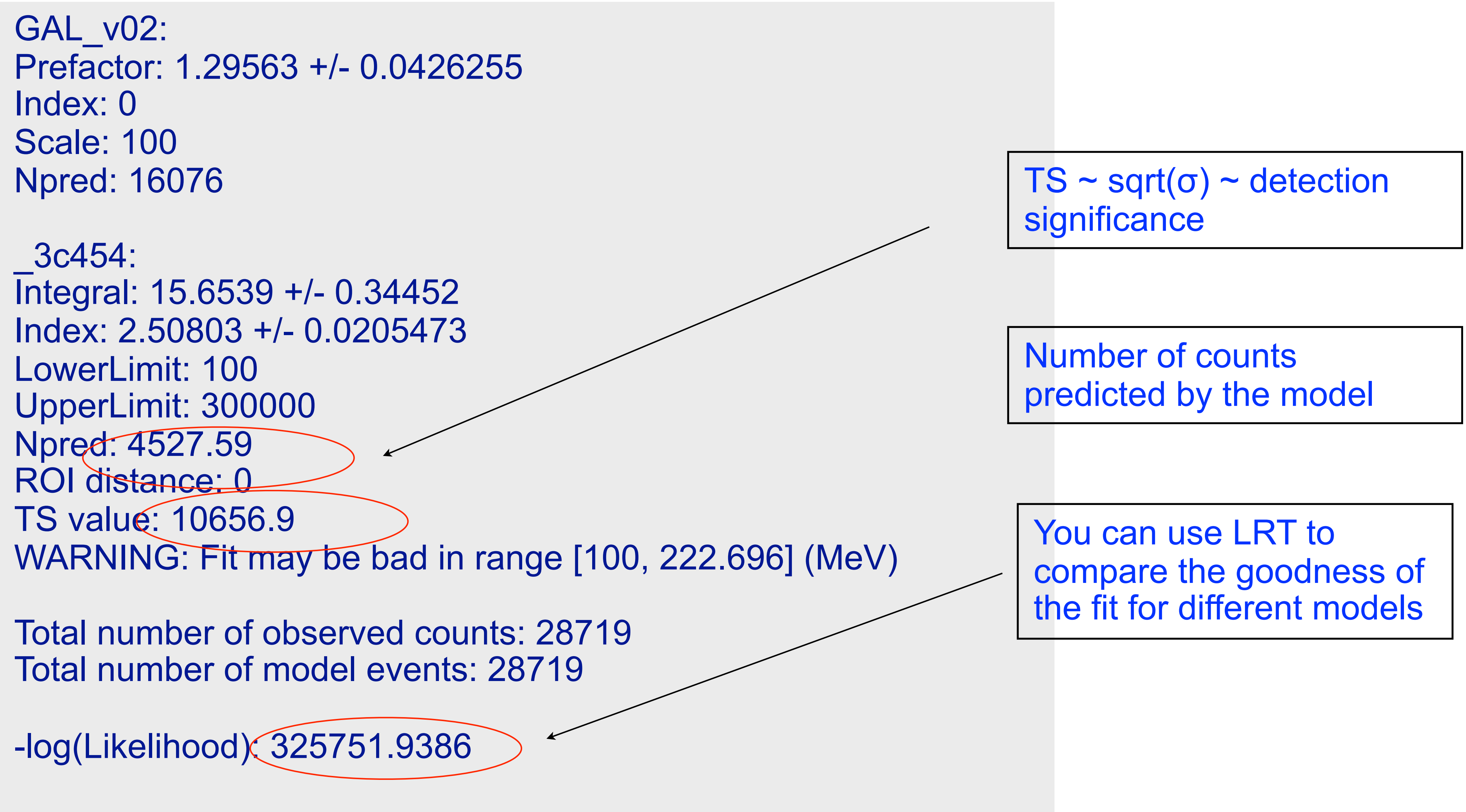
Running Likelihood: gtlike

```
$gtlike plot=yes
Statistic to use (BINNED|UNBINNED) [UNBINNED]
Spacecraft file[L090923112502E0D2F37E71_SC00.fits]
Event file[3c454_100_300000_evt02.fits]
Unbinned exposure map[3c454_100_300000_ExpMap.fits]
Exposure hypercube file[3c454_100_300000_ExpCube.fits]
Source model file[3c454_srcmdl.xml]
Response functions to use[                              ]
Optimizer (DRMNFB|NEWMINUIT|MINUIT|DRMNGB|LBFGS) [MINUIT]
```

Running Likelihood: gtlike



Running Likelihood: gtlike



```
GAL_v02:
Prefactor: 1.29563 +/- 0.0426255
Index: 0
Scale: 100
Npred: 16076

_3c454:
Integral: 15.6539 +/- 0.34452
Index: 2.50803 +/- 0.0205473
LowerLimit: 100
UpperLimit: 300000
Npred: 4527.59
ROI distance: 0
TS value: 10656.9
WARNING: Fit may be bad in range [100, 222.696] (MeV)

Total number of observed counts: 28719
Total number of model events: 28719

-log(Likelihood): 325751.9386
```

TS ~ sqrt(σ) ~ detection significance

Number of counts predicted by the model

You can use LRT to compare the goodness of the fit for different models

Likelihood form python

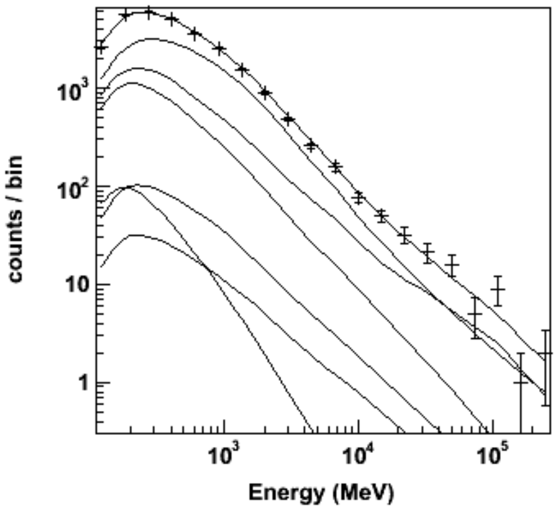
```
from UnbinnedAnalysis import *
my_obs=UnbinnedObs('3c454_100_300000_evt02.fits',
scFile='L090923112502E0D2F37E71_SC00.fits',
expMap='3c454_100_300000_ExpMap.fits',
expCube='3c454_100_300000_ExpCube.fits',
irfs='P6_V3_DIFFUSE')
analysis= UnbinnedAnalysis (my_obs,'3c454_srcmdl.xml',optimizer='MINUIT')
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')
print "TS=",ts
print "Npred=",npred
```

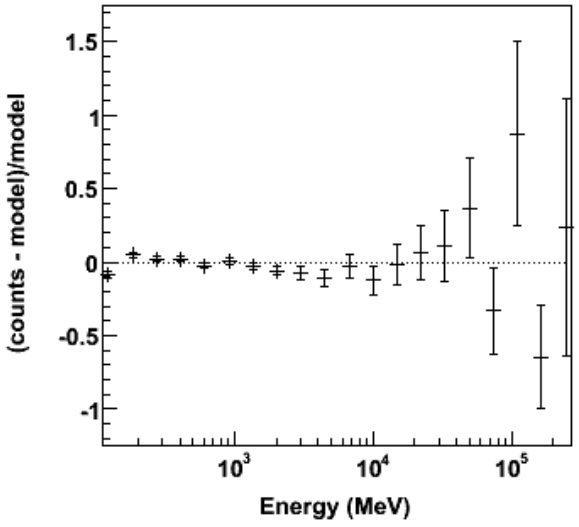
Likelihood form python

print analysis['_3c454']

```
_3c454
   Spectrum: PowerLaw2
16       Integral:  1.565e+01  3.445e-01  1.000e-04  1.000e+04 ( 1.000e-07)
17          Index:  2.508e+00  2.055e-02  1.000e+00  5.000e+00 (-1.000e+00)
18     LowerLimit:  1.000e+02  0.000e+00  3.000e+01  5.000e+05 ( 1.000e+00) fixed
19     UpperLimit:  3.000e+05  0.000e+00  3.000e+01  5.000e+05 ( 1.000e+00) fixed
```

analysis.plot()





building SEDs

when the statistics is decent it is possible to build SEDs from the unbinned likelihood done in different energy bins:

1. Perform the likelihood analysis by means of a PL2, partitioning the whole energy window in *N* bins uniformly distributed in the logarithm:
2.The PowerLaw2 model uses the integrated flux as a free parameter, use this model
3. you can covert the energy bin integrated flux using the following algorithm

$$E_{ref} = \sqrt{E_1 \cdot E_2}$$

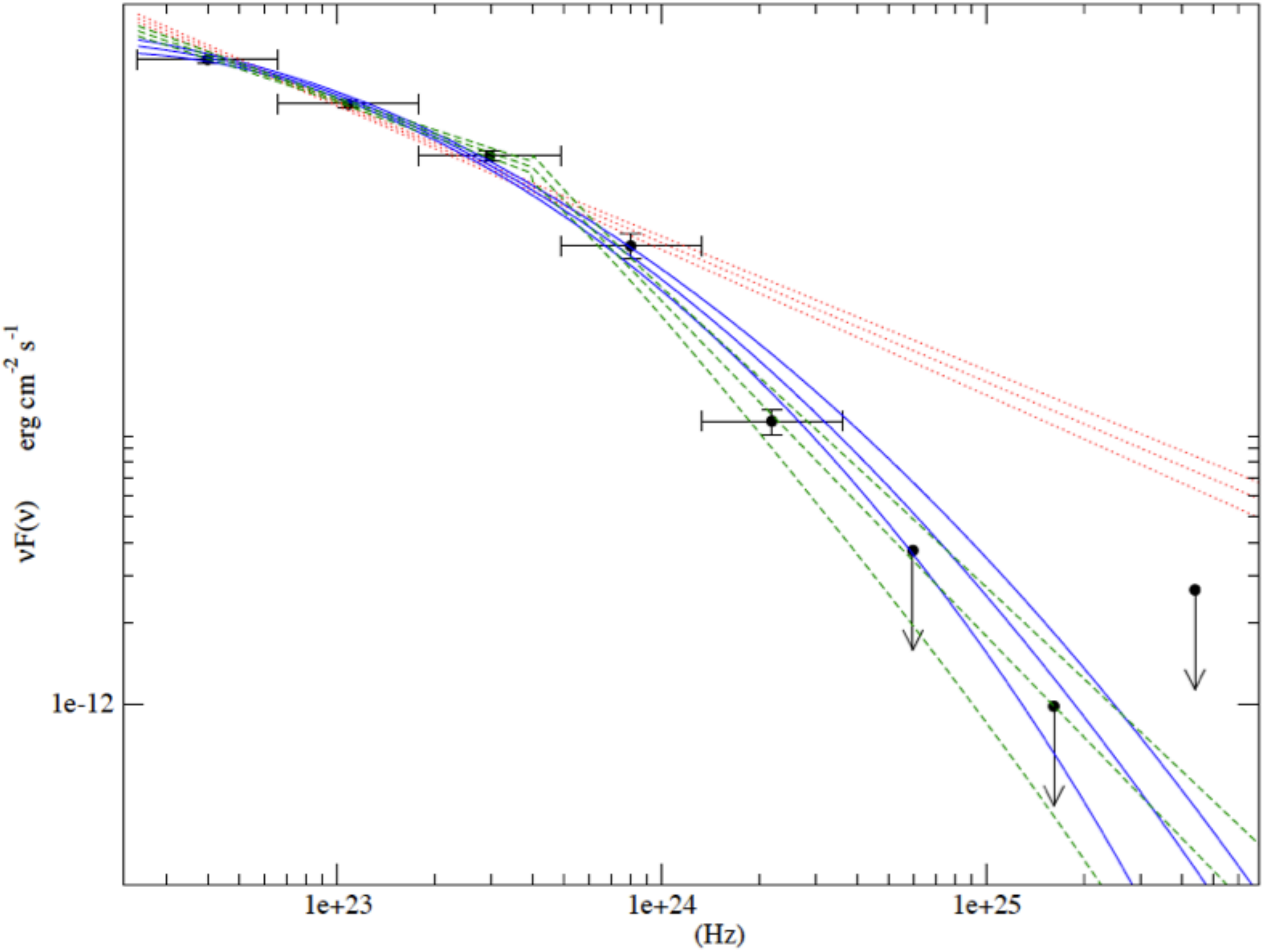
$$S = \frac{F_{bin} \cdot (index - 1) \cdot E_{ref}^{2-index}}{E_1^{-index+1} - E_2^{-index+1}}$$

where, index is the PL2 index, and F_{bin}, is the PL2 integral, in each energy bin.

- You can do the same to generate a “butterfly” of the best fit model.

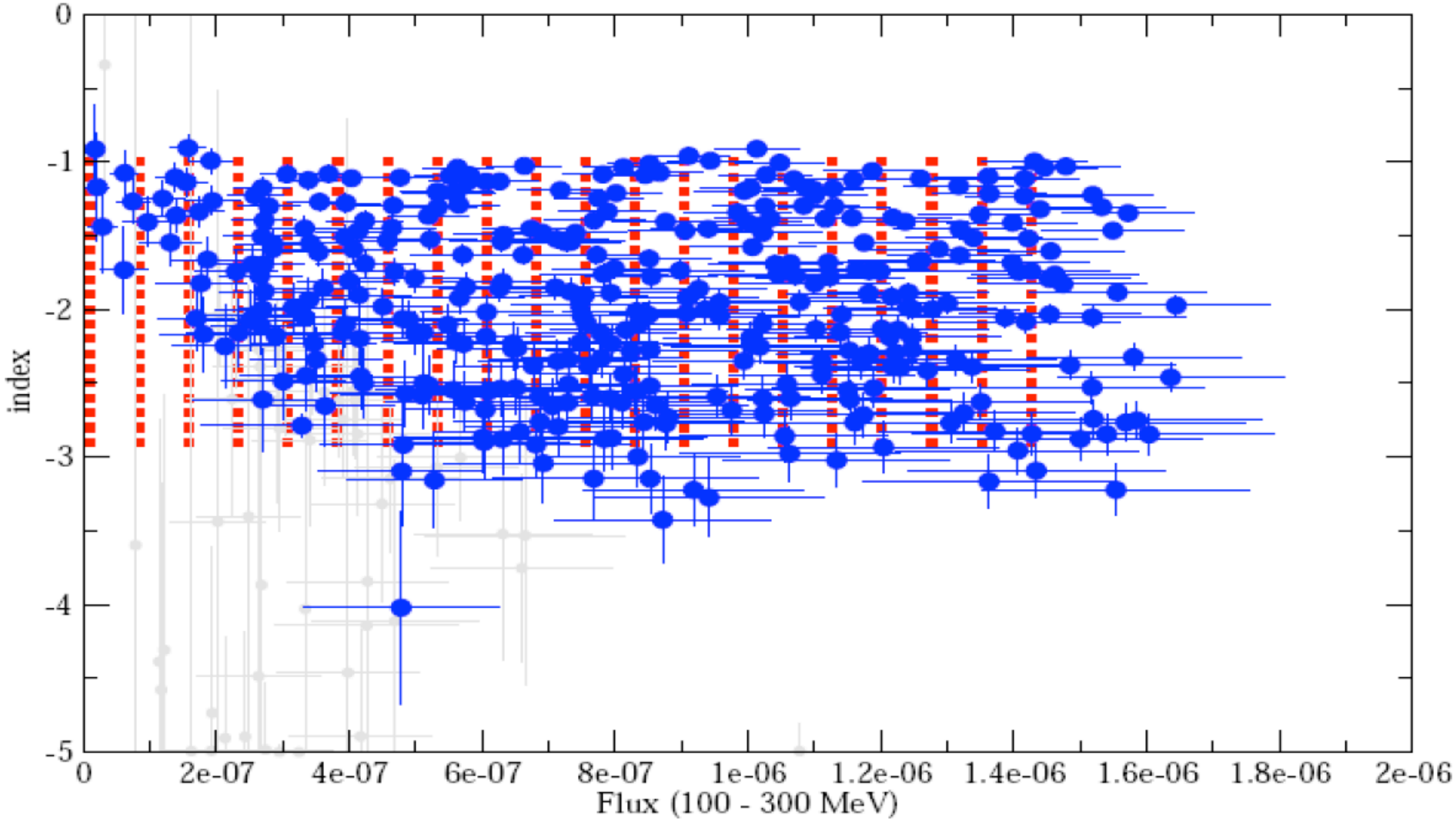
building SEDs: 3C454.3

compare binned fluxes and PL, LogParabola, and Broken PL

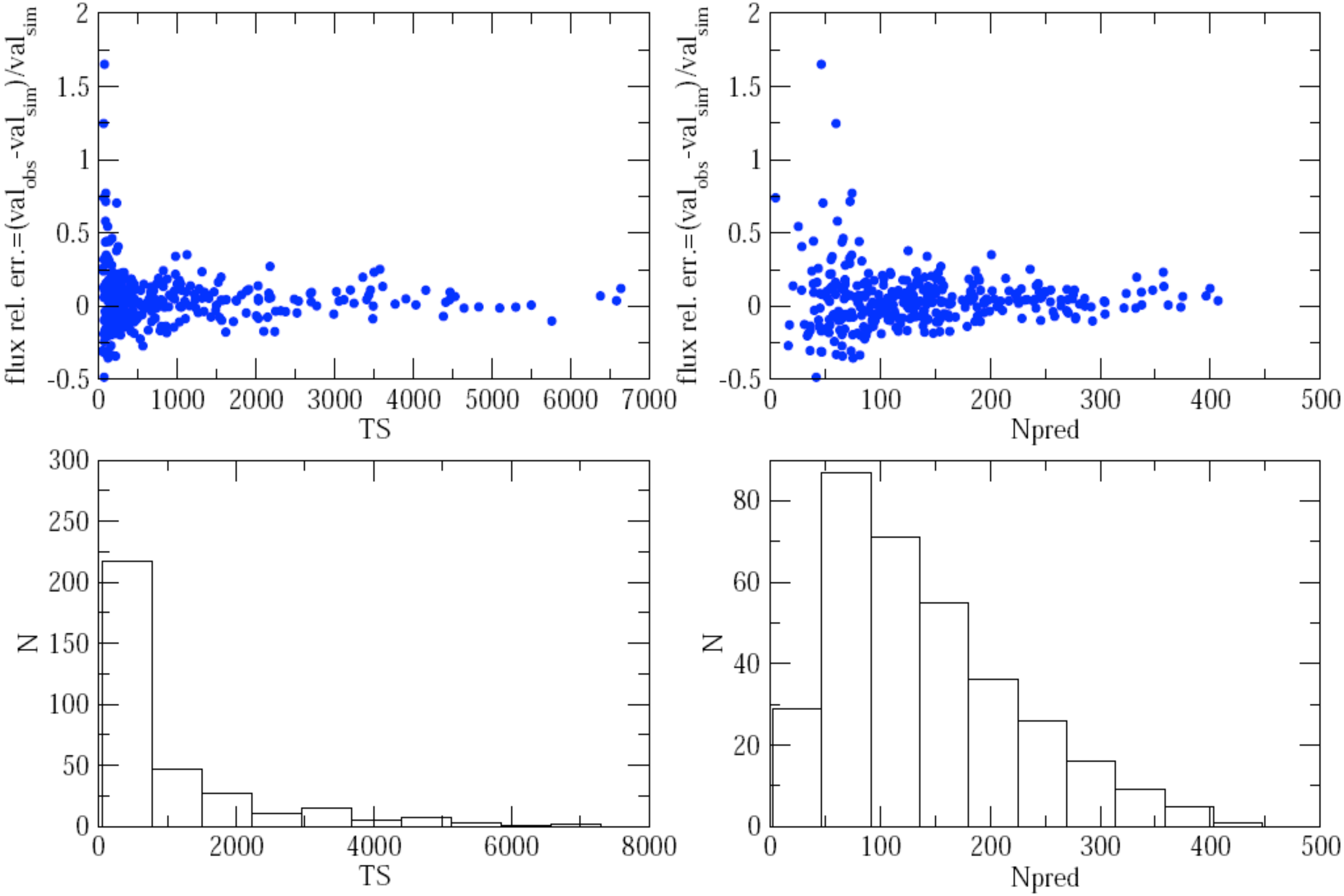


More on the spectral analysis

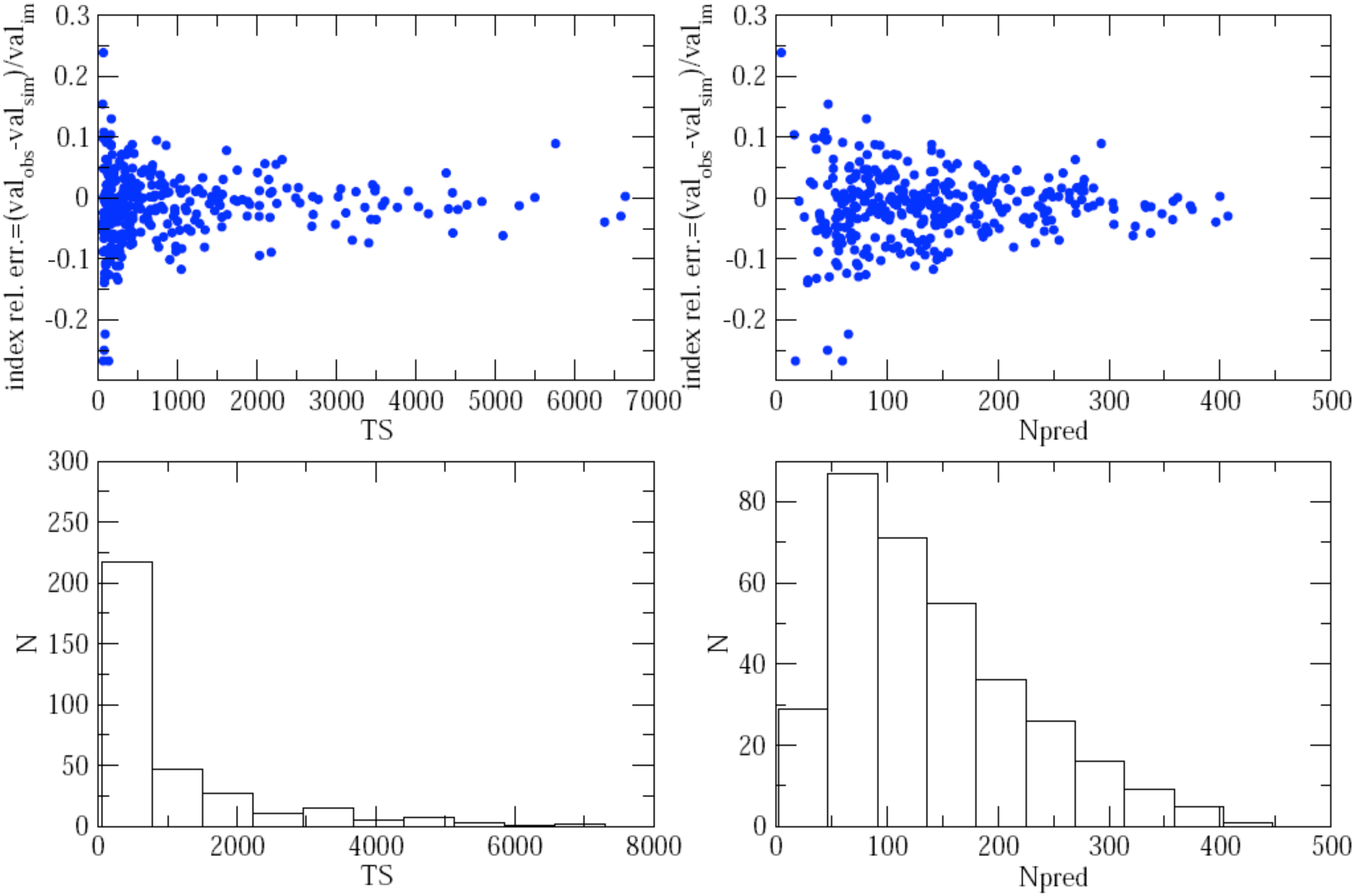
Bias on weekly time scale



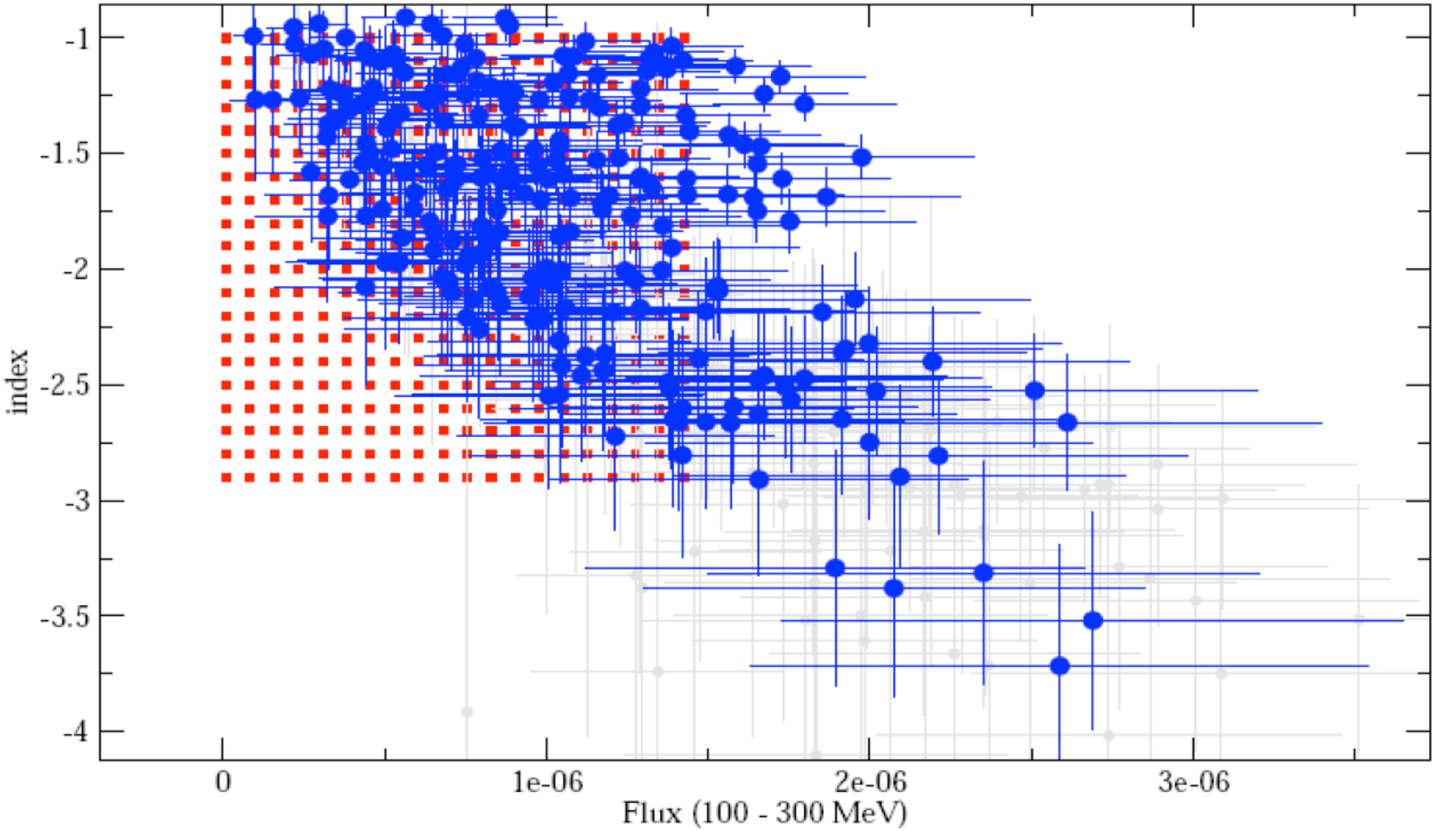
Flux dispersion: weekly



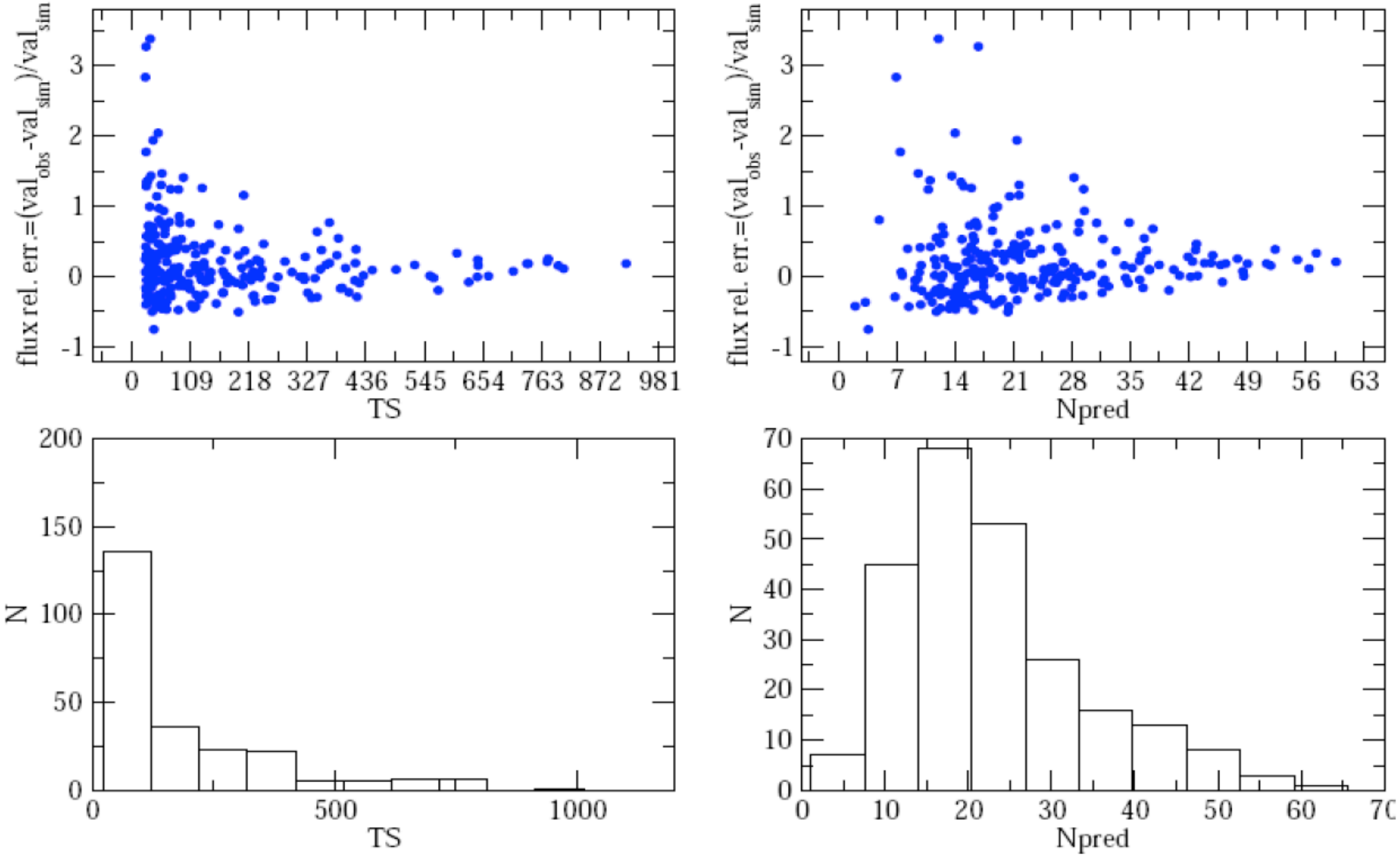
Index dispersion: weekly



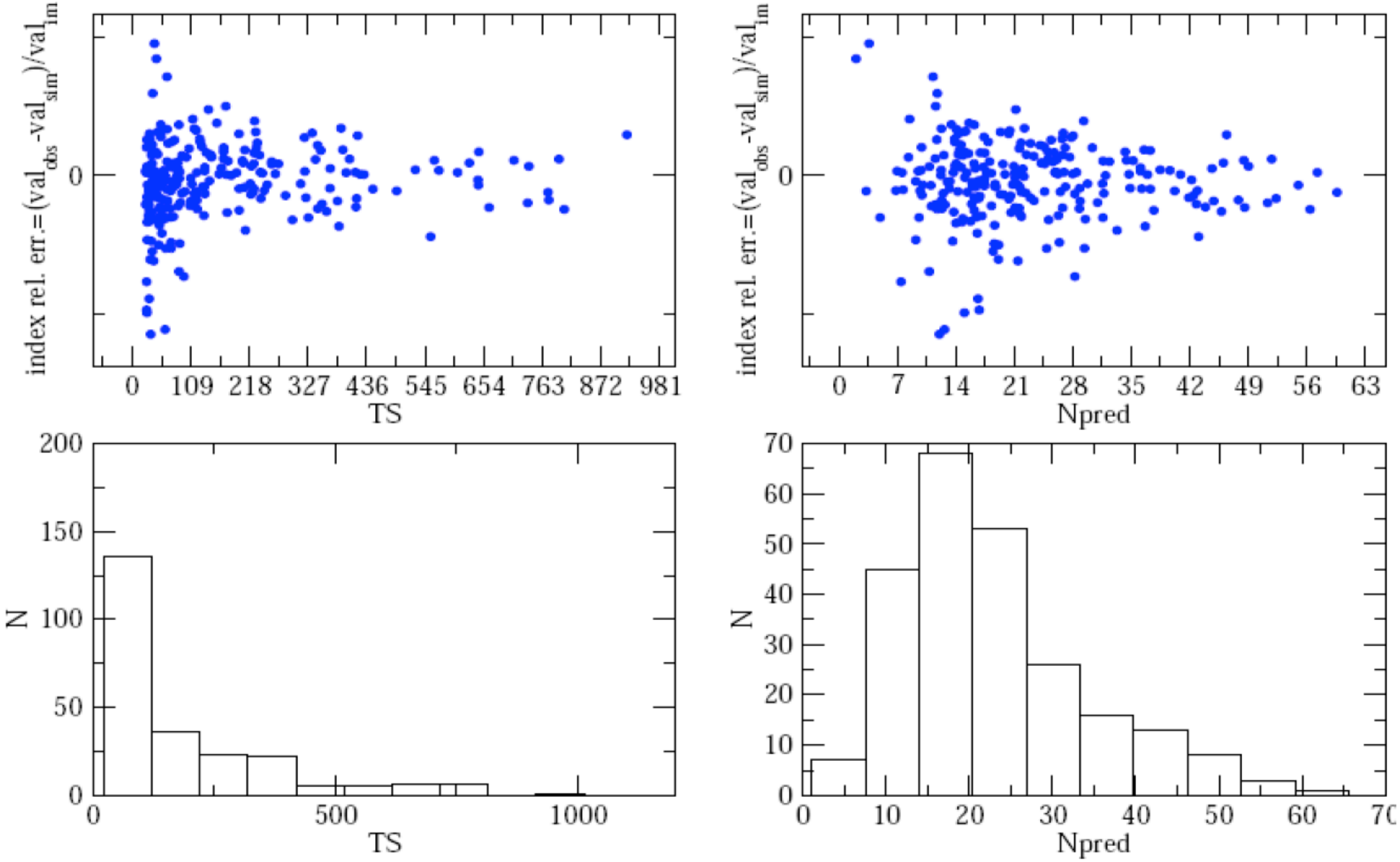
Bias on daily time scale



Flux dispersion: daily



Index dispersion: daily



Understanding better the parameter space and best model choosing

The problem of model decision in spectral analysis

•We want to understand the capability of the LAT to measure non-power law spectra

•This means to understand in terms of statistics whether a source spectra is described better by a power-law (PL) or a log-parabola (LP), broken-power-law(BKN) or exponential cut-off etc...

•The problem has two level of complexity, one is observational/instrumental and relies mainly on the detector performance (sensibility and calibration) on the flux of the source the bkg....

•The second level of complexity relies on the statistic side. In fact (Protassov etal. 2002 ApJ 571, James Chiang private communication) the method of the LRT in its classical (or frequentist) derivation can't be applied to many of our cases.

•In particular LRT(H0/H1) ~ c^2(K0-K1) is true only if: the models are nested,the null hypothesis parameters do not lie on their boundary space, some mathematical regularities conditions are satisfied.

Understanding better the parameter space and best model choosing

Possible solution to the problem

- I will not go through the details of the statistical methods nor in the dispute Bayesian vs. frequentist (there is a vast literature....), but one of the possible solutions to the problem consists in re-sampling the LRT statistic for each specific case

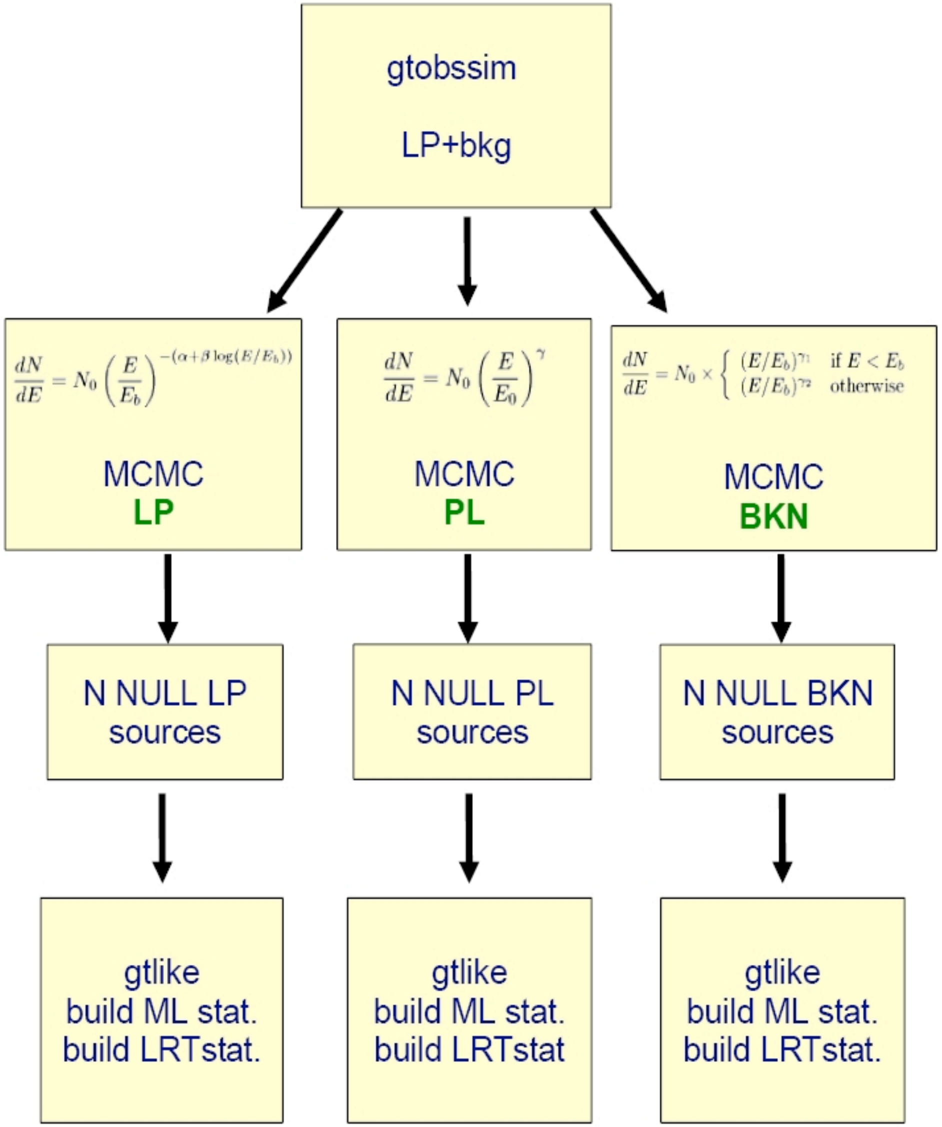
- The Posterior predictive p-value (Protassov et al. 2002 ApJ 571) should be one of the most robust technique, but also Bayes factor or BIC (Bayesian information criterion) are often used in cosmology econometrics biology etc...

- We use the technique of the Posterior predictive p-value via a MCMC (Monte Carlo Markov Chain) using a Metropolis algorithm to re-calibrate the LRT statistics getting rid of limitations given in the previous slide

$$T(x) = -2Log\left(\frac{L(\theta_0|x)}{L(\theta_1|x)}\right)$$

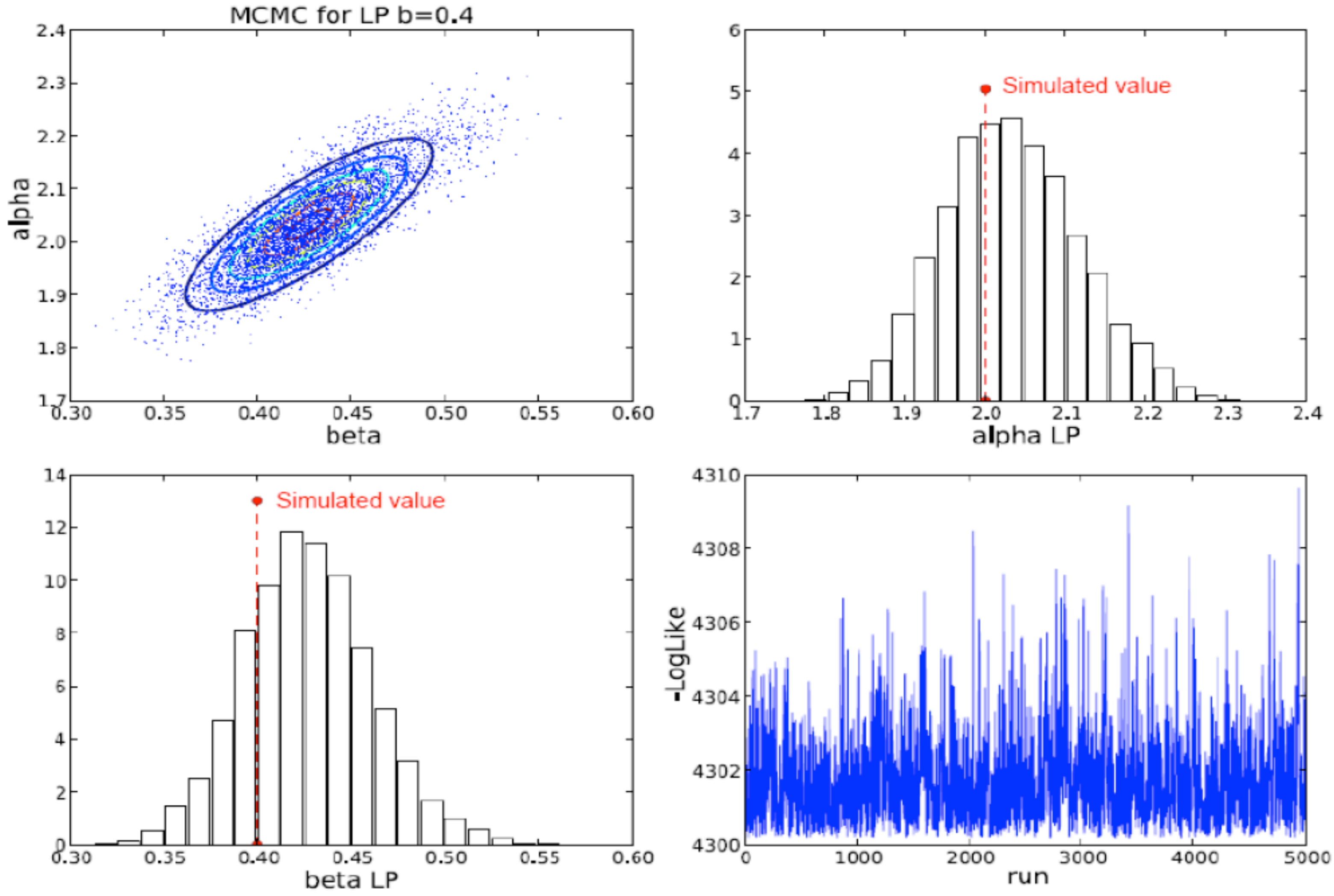
$$p = \frac{1}{N}\sum\left[T(x_{MCMC}) > T(x_{obs})\right]$$

Schematic view of MCMC approach

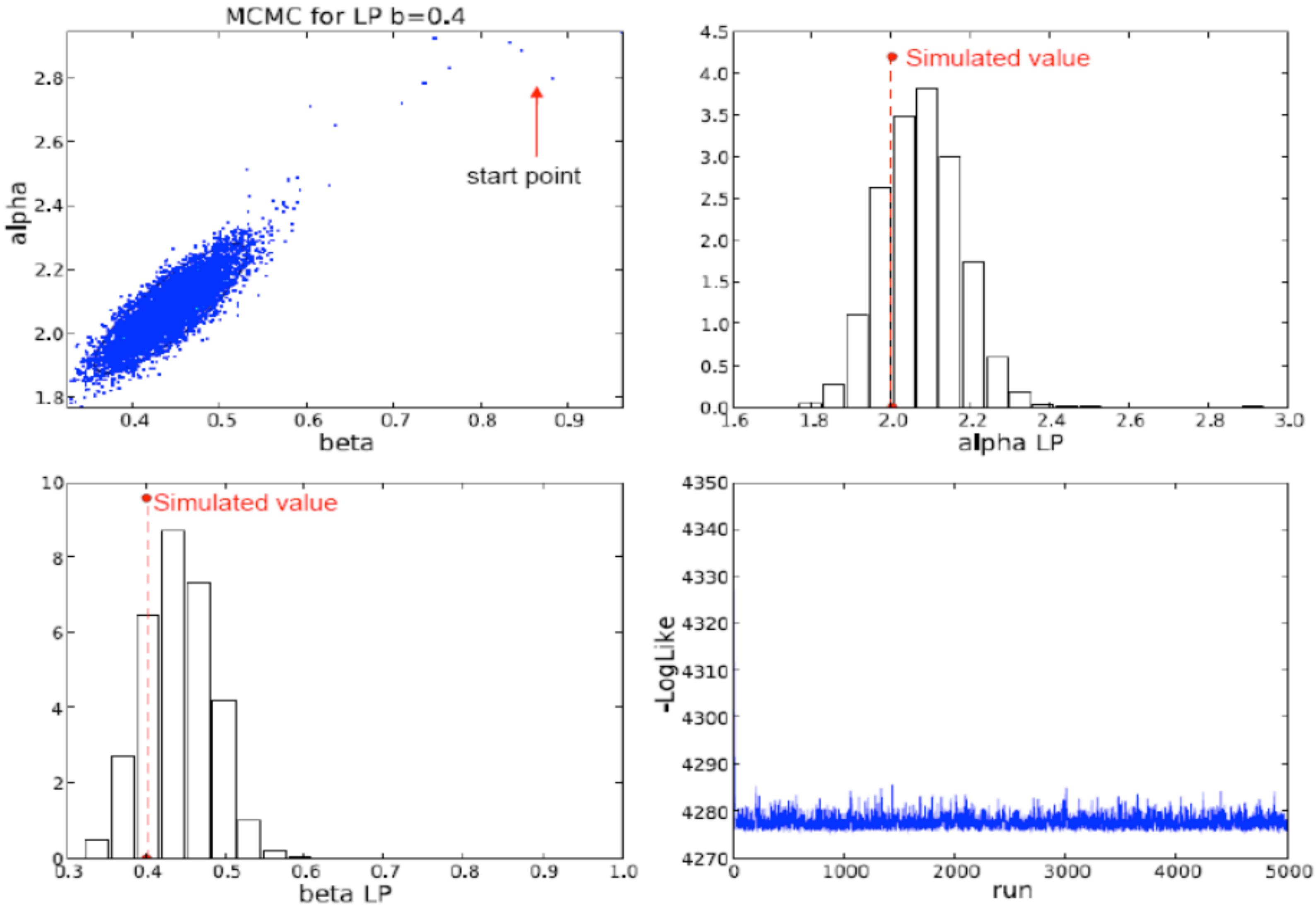

gtobssim
LP+bkg
$\frac{dN}{dE} = N_0 \left(\frac{E}{E_b}\right)^{-(\alpha+\beta\log(E/E_b))}$
MCMC
LP
$\frac{dN}{dE} = N_0 \left(\frac{E}{E_0}\right)^{\gamma}$
MCMC
PL
$\frac{dN}{dE} = N_0 \times \begin{cases} (E/E_b)^{\gamma_1} & \text{if } E < E_b \\ (E/E_b)^{\gamma_2} & \text{otherwise} \end{cases}$
MCMC
BKN
N NULL LP sources
N NULL PL sources
N NULL BKN sources
gtlike
build ML stat.
build LRTstat.
gtlike
build ML stat.
build LRTstat
gtlike
build ML stat.
build LRTstat.

- We simulate a source with a log-parabolic (steady) spectral distribution plus galactic and extra-galactic bkg. The source flux was chosen for three different cases: low (1*10^-6) medium (2.5*10^-6) high (5*10^-6). We call this simulation the "REAL OBSERVATION"
- Run the MCMC chain and sample N realizations for each of our test models. Each accepted step in the chain is stored to build the parameters posterior distribution These are the samples for the "NULL" models
- Fit these each N samples of each NULL sources data set with PL, BKN and LP
- We build the statistic of the log-Likelihood and compare it with the values from the "real observation"
- We build the statistic of the LRT for the NULL and the ALTERNATIVE models and compare it with that from "real observation

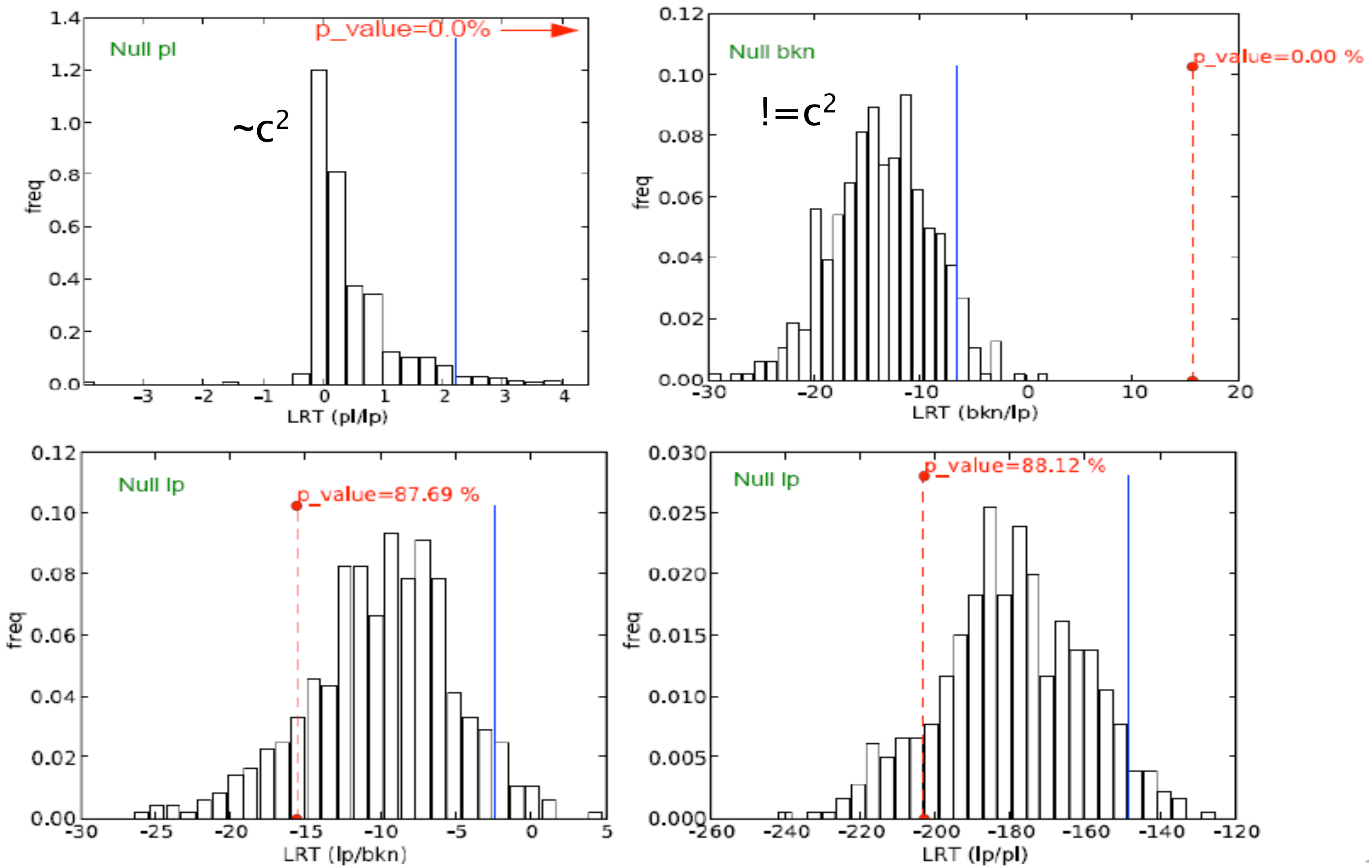
Simulated data:Weekly integration for a bright source b=0.4



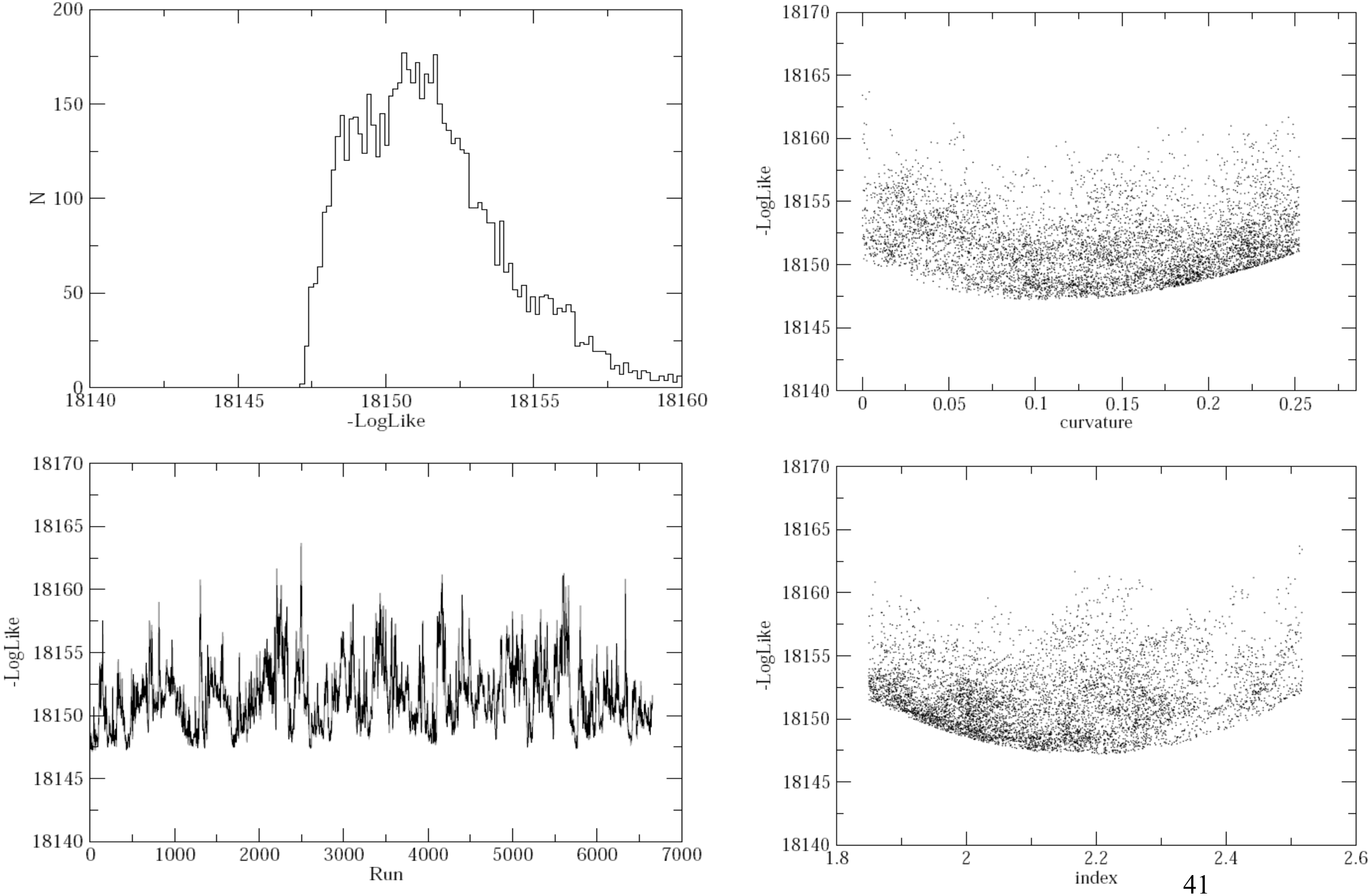
Convergence test



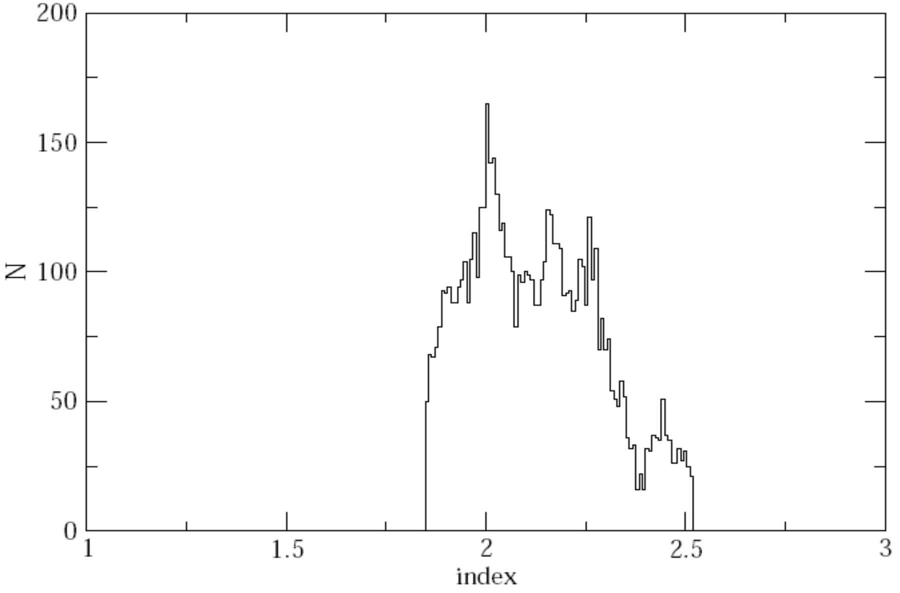
Weekly integration for a bright source b=0.4

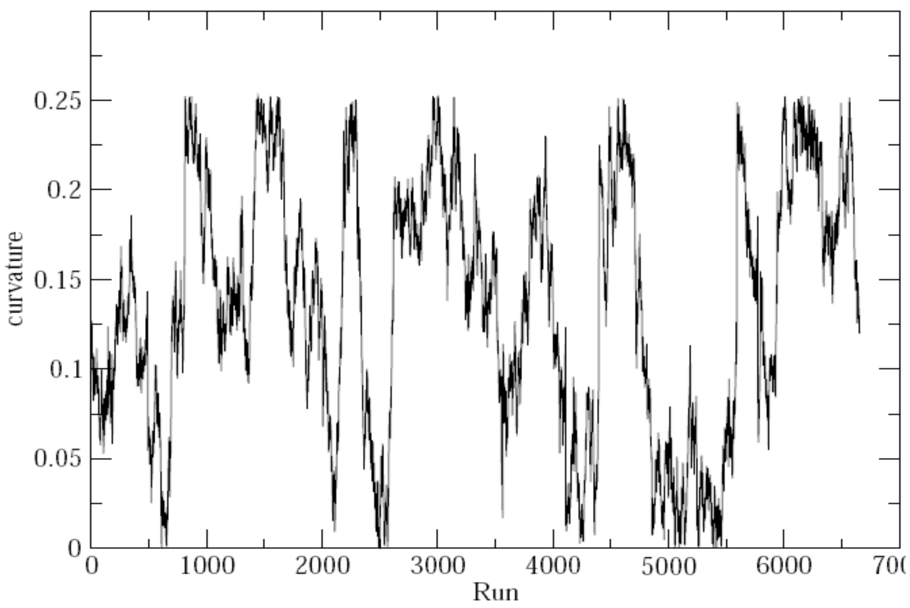


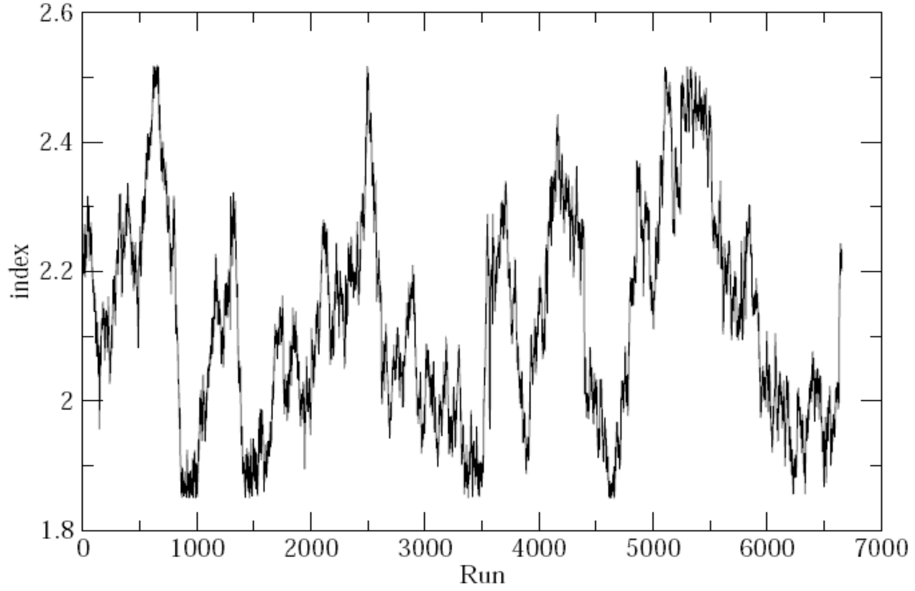
Investigating the parameter space: MCMC real data

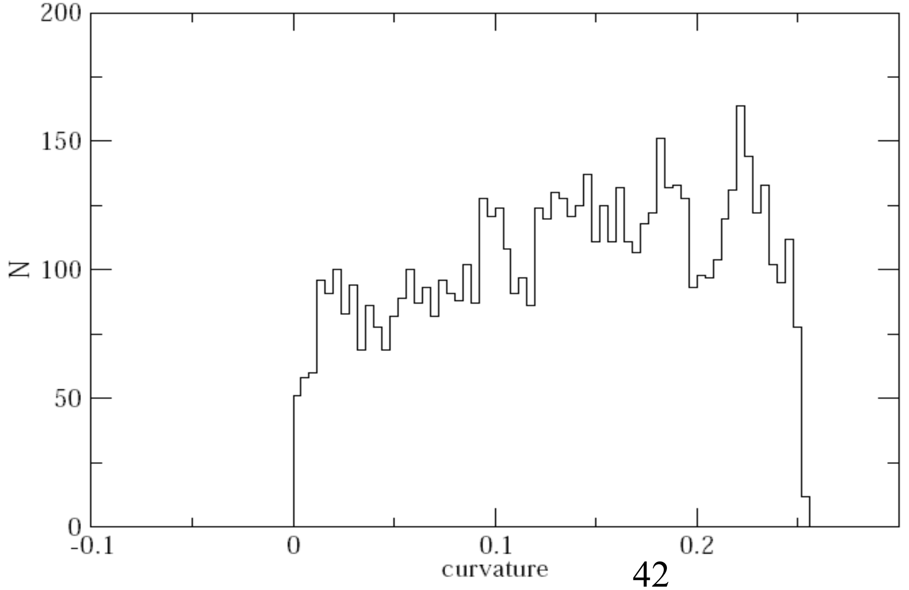


Investigating the parameter space: MCMC, real data

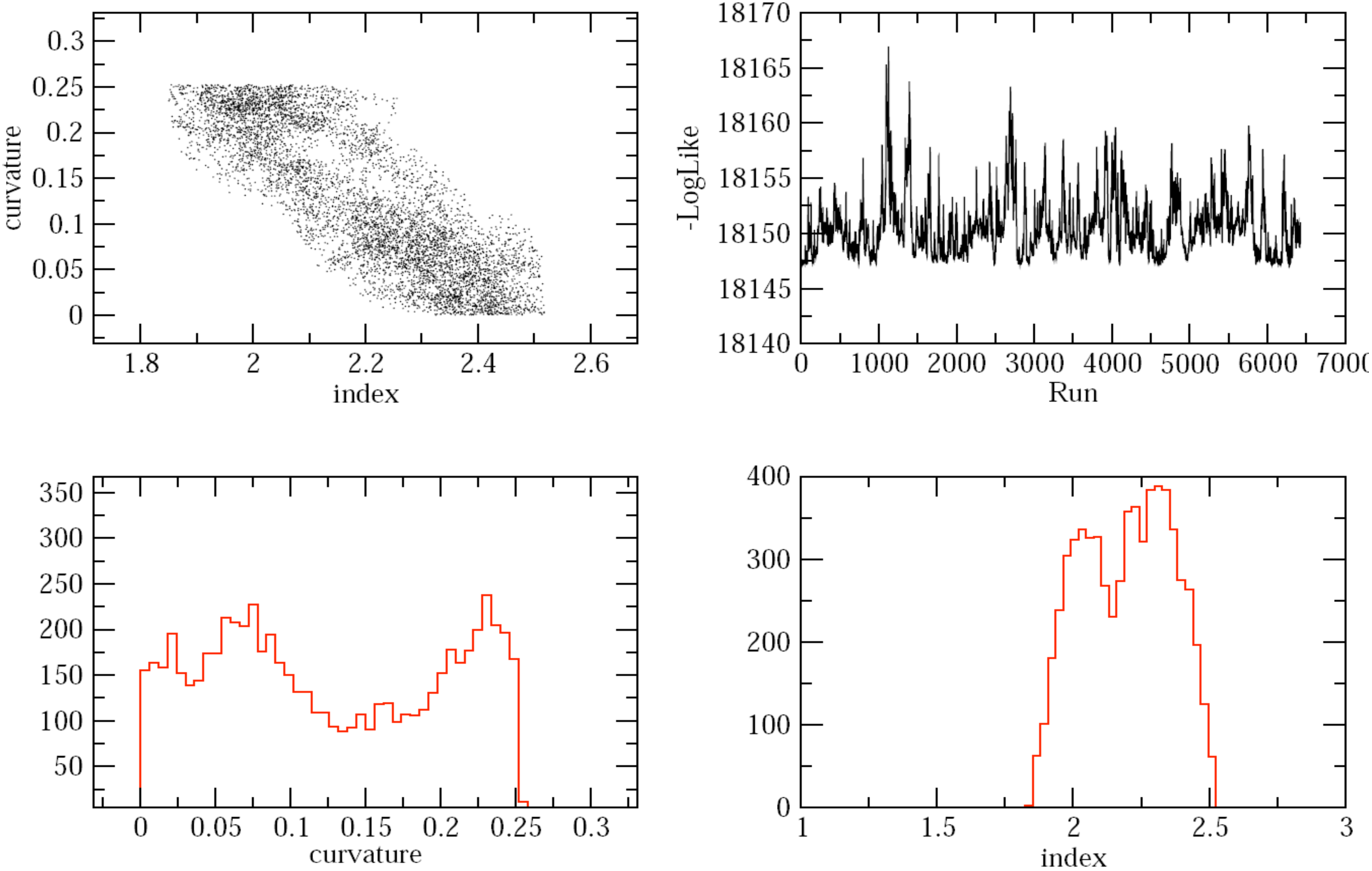








Investigating the parameter space: MCMC



$$\underset{\substack{\uparrow\\ \text{Posterior}}}{p(\boldsymbol{\theta}\,|\,\boldsymbol{d},I)} \propto \overset{\substack{\text{Prob. of the data}\\ \downarrow}}{p(\boldsymbol{d}\,|\,\boldsymbol{\theta},I)}\,\underset{\substack{\uparrow\\ \text{Prior factor}}}{p_0(\boldsymbol{\theta},I)} \propto p(\boldsymbol{d}\,|\,\boldsymbol{\theta},I) = \overset{\substack{\text{Likelihood.}\\ \downarrow}}{\mathcal{L}(\boldsymbol{\theta};\boldsymbol{d})}\,,$$

1. Select a new trial point $\boldsymbol{x}^*$, chosen according to a symmetric *proposal* pdf $q(\boldsymbol{\theta}^*\,|\,\boldsymbol{\theta}_t)$.
2. Calculate the *acceptance probability*

$$A(\boldsymbol{x}^*\,|\,\boldsymbol{x}_t) = \min\left[1,\,\frac{\tilde{p}(\boldsymbol{\theta}^*)}{\tilde{p}(\boldsymbol{\theta}_t)}\right]\,. \tag{118}$$

3. Accept $\boldsymbol{x}^*$ with probability $A(\boldsymbol{x}_t,\boldsymbol{x}^*)$, i.e.
   - if $\tilde{p}(\boldsymbol{\theta}^*) \geq \tilde{p}(\boldsymbol{\theta}_t)$, then accept $\boldsymbol{x}^*$;
   - if $\tilde{p}(\boldsymbol{\theta}^*) < \tilde{p}(\boldsymbol{\theta}_t)$, extract a uniform random number between 0 and 1 and accept $\boldsymbol{x}^*$ if the random number is less then $\tilde{p}(\boldsymbol{\theta}^*)/\tilde{p}(\boldsymbol{\theta}_t)$.

If the point is accepted, then $\boldsymbol{x}_{t+1} = \boldsymbol{x}^*$. Otherwise $\boldsymbol{x}_{t+1} = \boldsymbol{x}_t$