

Model Source Probabilities for Event Data

Jim Chiang
KIPAC/SLAC/ISOC

Introduction and Background

- There is a new tool, `gtsrcprob`, that will assign to each event in an FT1 file, the probability that that event belongs to a particular source in an xml model definition.
- An early motivation for this tool was to enhance searches for periodicity using PSF information. Other methods include
 - Energy-dependent acceptance cones.
 - Weighting events by PSF value wrt source location.
 - * Even with perfect source location information, using the PSF alone is poorly defined. Probabilities can only be assigned concretely in the context of sky model.
 - See LK-57@JIRA.

Computing Source Probabilities for Each Event

For an event with measured energy ε' and direction \hat{p}' , the model densities are given by

$$M_i(\varepsilon', \hat{p}', t) = \int d\varepsilon d\hat{p} S_i(\varepsilon, \hat{p}) R(\varepsilon', \hat{p}'; \varepsilon, \hat{p}, t) \quad (1)$$

$$= \int d\varepsilon d\hat{p} S_i(\varepsilon, \hat{p}) A(\varepsilon, \hat{p}, t) P(\hat{p}'; \varepsilon, \hat{p}, t) D(\varepsilon'; \varepsilon, \hat{p}, t) \quad (2)$$

Here the integral is taken over all true energies ε and directions \hat{p} , R is the instrument response, and $S_i(\varepsilon, \hat{p})$ is photon spectral density, $dN/dEdAdtd\Omega$, of source i . The instrument response has been factored in terms of effective area, A , point spread function, P , and energy dispersion, D .

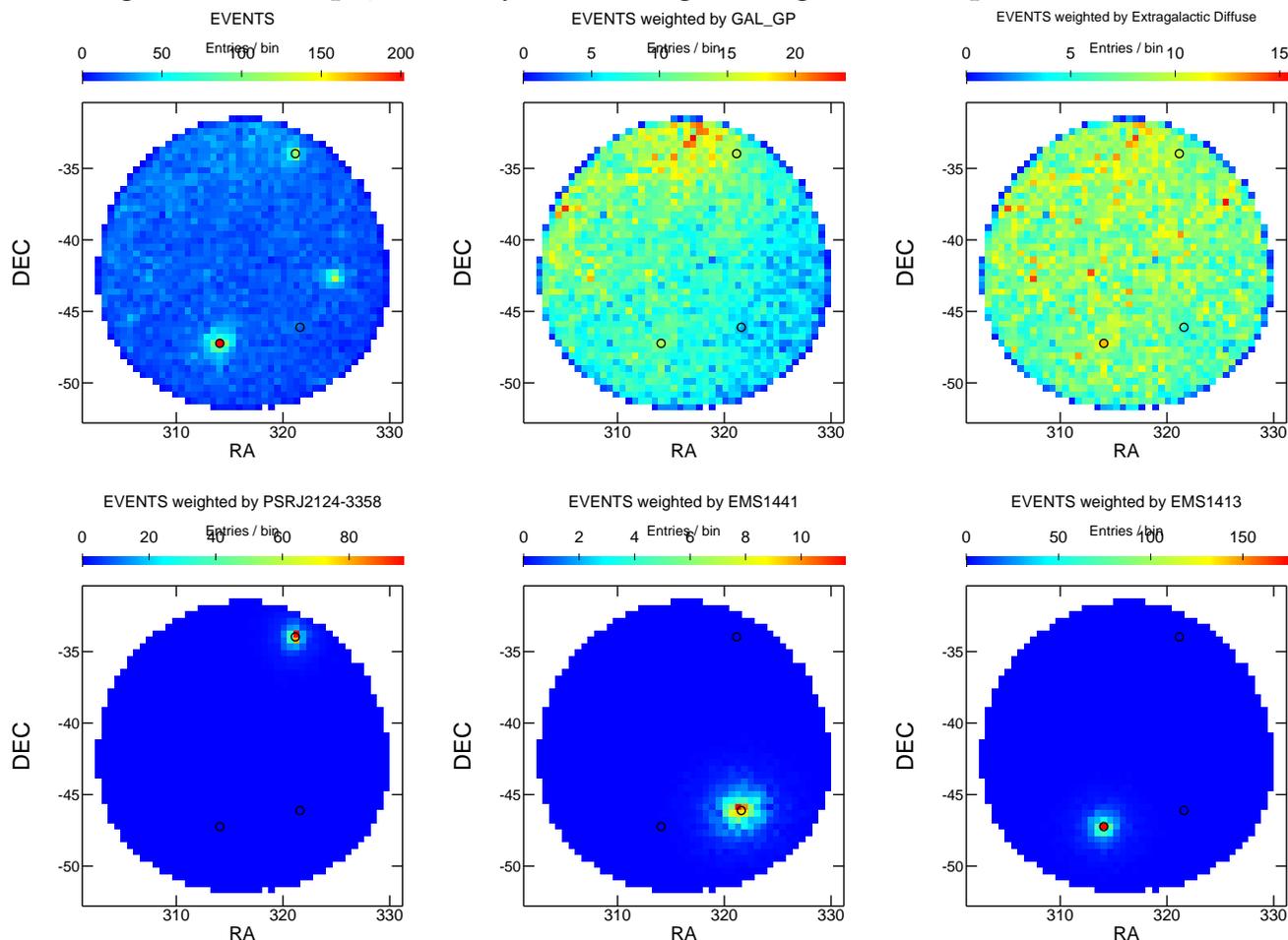
The probability that a specific event can be attributed to source i is then

$$p_i = \frac{M_i}{\sum_j M_j} \quad (3)$$

This is what `gtsrcprob` computes for each source for each event.

Weighted Counts Maps

In addition to weighting events by these probabilities for periodicity searches, one can also weight them in making counts maps, thereby obtaining “images” of a particular source component.



Using gtsrcprob

This is available in ScienceTools-09-16-01.

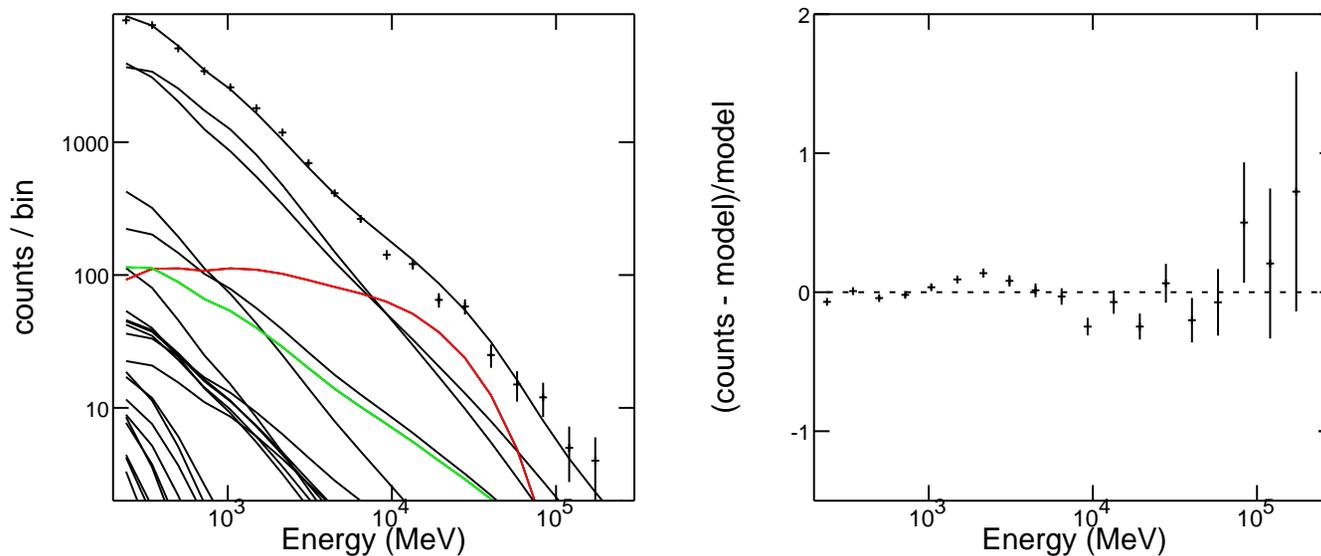
```
% gtsrcprob srclist=srclist
Event file[] ft1_in.fits
Spacecraft file[] ft2.fits
output filename[] ft1_out.fits
Source model file[] my_model.xml
Response functions to use[P6_V3_DIFFUSE]
%
```

The optional parameter `srclist` specifies an ascii file that lists the sources for which the user wants to compute probabilities. The default behavior will compute for all sources in the xml model file. The probabilities are added as columns labeled by the name of the source.

NB: The output FT1 file cannot be the input FT1 file, i.e., a new FT1 file must be created.

Weighted Counts Spectra

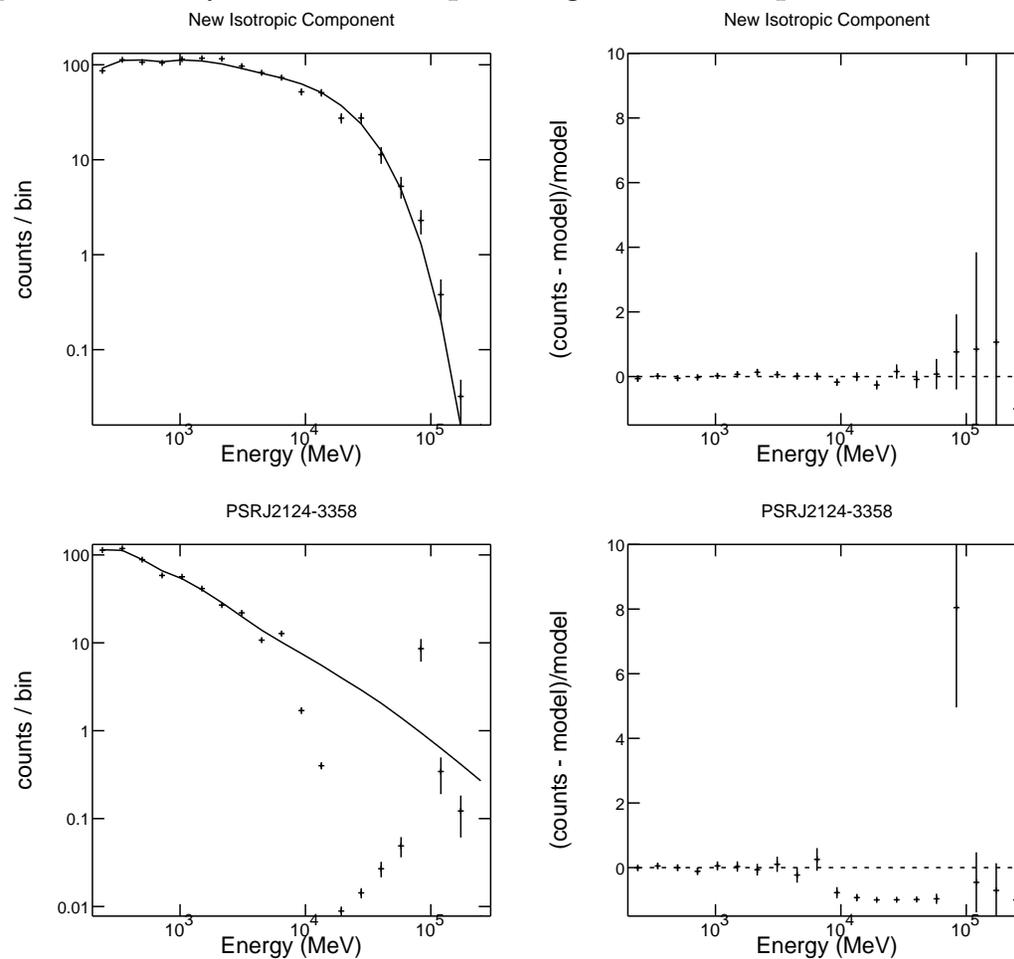
A far more useful application would be to produce source probability weighted counts spectra. Plots of the model components and model sum on top of the **total observed** counts spectrum has been available from `pyLikelihood`:



There are 38 sources (35 point, 3 diffuse) in this model fit. The red curve is the model spectrum for a “new isotropic component” modeled as an exponentially cutoff power-law; the green curve is PSR J2124-3358 modeled simply as a power-law.

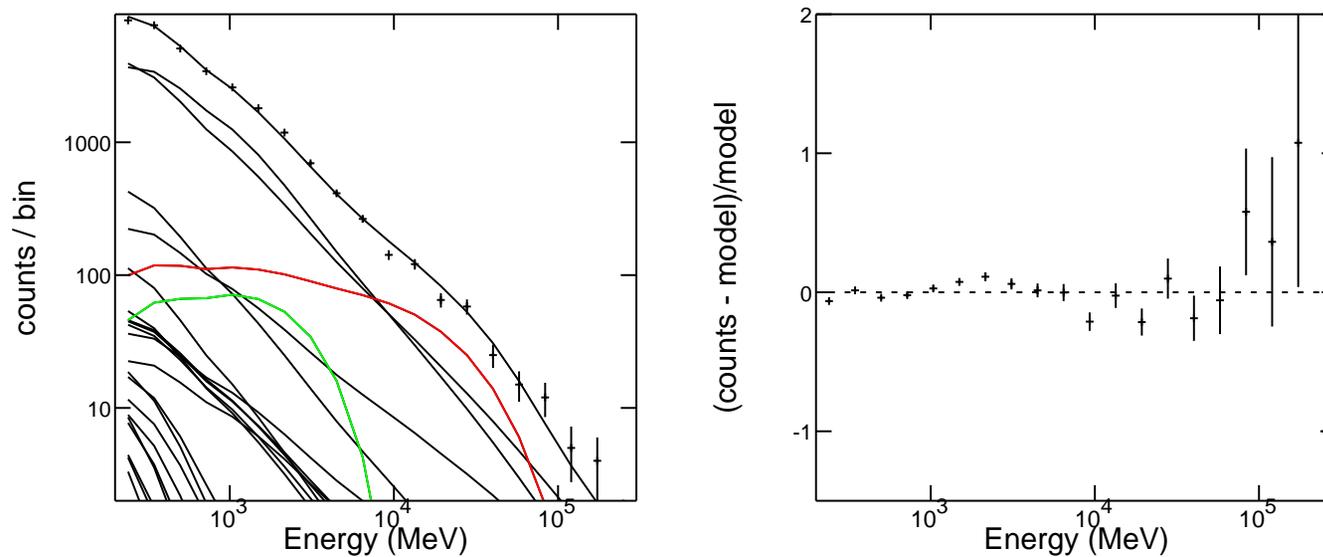
Weighted Counts Spectra

By applying the source probability weighting, a representation of the counts spectra is obtained that can be compared directly to the corresponding model component:



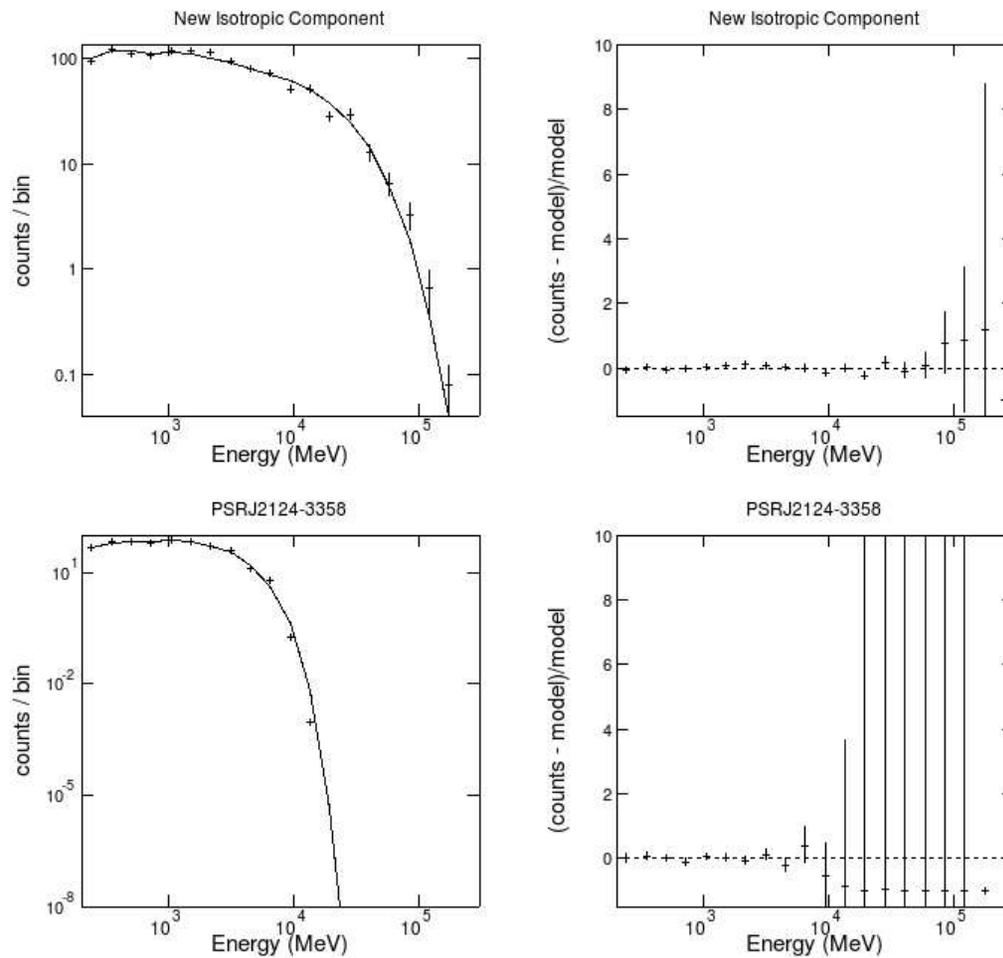
Weighted Counts Spectra

Refit these data, but use a super-exponential cutoff power-law to model the spectrum of PSR J2124-3358. The fit to the total counts spectrum appears largely unchanged:



Weighted Counts Spectra

However, the weighted spectra agree more substantially with the model components:



Weighted Counts Spectra

The weighted counts spectra will be available in `pyLikelihood` without having to run `gtsrcprob` separately:

```
like = unbinnedAnalysis(...)

# Produce the standard total counts spectrum.
like.plot()

# Overlay the individual model components.
like.plotSource('New Isotropic Component', 'red')
like.plotSource('PSRJ2124-3358', 'green')

# New commands to create the weighted counts spectra.
like.plotSourceFit('New Isotropic Component')
like.plotSourceFit('PSRJ2124-3358')
```

Notes and Caveats

- Error bars for weighted counts are not quite right. Currently using

$$\langle p_i \rangle = \left(\sum_i p_i \right) / N \quad (4)$$

$$\tilde{N} = \sum_{i=1}^N p_i = \langle p_i \rangle N \quad (5)$$

$$\sigma_{\tilde{N}} = \langle p_i \rangle \sqrt{N} \quad (6)$$

$$= \tilde{N} / \sqrt{N} \quad (7)$$

- Even with correct errors, one should not try to use χ^2 with the weighted spectra to evaluate goodness-of-fit, because the “data” are not independent of the model.
- It should be possible create the same sort of weighted spectra for binned likelihood.