Fermi
Gamma-ray
Space Telescope

High-level LAT Simulations

Seth Digel
KIPAC/SLAC

Fermi Summer School 2012
Lewes, Delaware

Outline

- Introduction: Why simulate data?
- Monte Carlo simulations for LAT data
- Specifying simulations
  - About time
- Outputs from gtobssim
- Walkthrough of the gtobssim parameter file
- Advice and gotchas

Why simulate data?

- Monte Carlo simulations were vital for the development of he LAT instrument concept, event reconstruction, background rejection, even classification, and IRFs
- Monte Carlo simulations were also vital for the validation of the high-level analysis
  - To test and characterize the performance of source detection, localization, etc. while knowing the 'truth'
  - To evaluate false detection rates, source significance
- Or you could evaluate effects of different observing strategies, if you could make up your own FT2 file (with gtorbsim)

Monte Carlo simulation for LAT data

- Monte Carlo event simulation is ‘Monte Carlo’ in more than one sense of the word
- Event generation (scheduling is same for Gleam and gtobssim)
  - γ-rays are generated randomly, incident on a sphere that surrounds the LAT (cross sectional area = 6 m^2 for Gleam, max(A_{eff}) for gtobssim)
  - The γ-rays are generated with rates consistent with the sources in the specification (XML, more later) of the model
  - The code does this by ‘polling’ each source to find out when it plans to next issue a γ-ray, advancing the time (and the spacecraft location and orientation) to the time of the next γ-ray, generating it, and going around the loop again

Monte Carlo simulation for LAT data (2)

- **Event detection (full instrument simulation, Gleam)**
  - **As you heard from Luca on Thursday, uses GEANT4 particle interaction code and simulates the trigger and readout, followed by reconstruction**
  - **This is slow, more so at high energies**
  - **For this simulator long runs for astronomy, especially including incident charged particle and Earth limb gamma-ray backgrounds, have been very rare**

Monte Carlo simulation for LAT data (3)

- **Event detection (fast simulation, gtobssim)**
  - **The instrument is represented by the Instrument Response Functions**
  - **A generated event is accepted based on the effective area for the incident direction and energy relative to the cross sectional area for the event generation sphere***
  - **Then front vs. back is decided on based on their relative effective areas**
  - **Then the PSF and energy dispersion are used to assign the event a measured direction and a true direction**
  - **This is very fast and speed is also not dependent on energy**

* Livetime fraction and rate-dependent inefficiency are also applied, possibly rejecting the event.

Defining a gtobssim simulation

- Basically, you need a source model and a pointing/attitude/ livetime history
  - The source models are defined as XML file(s) [see later]
  - The pointing history is an FT2 file

Pointing history (FT2) for observation simulation

- **If you provide an input FT2 file, the pointing history and livetime fraction will be respected in the simulation**
- **If you do not provide an FT2 file, an idealized 'step rocking' observing pattern and default orbit will be used**
  - **You specify the rocking angle and (constant) livetime fraction**
  - **The output will include an FT2 file**
- **Alternatively, you could use gtorbsim**
  - **To simulate, say, a pointed observation***

* Pointed observations are not simulated with high fidelity but certainly to a good first approximation

Source model specifications

- **First question: Can the source model files be the same as gtlikelihood uses? No. The ModelEditor tool can translate formats for source specifications that have exact mapping between the two**
- **Example source specification for a point source:**

```
<source name="vela" flux="0.00928">
    <spectrum escale="MeV">
        <particle name="gamma"> <power_law emin="30.0" emax="100000." gamma="1.62"/>
        </particle>
        <celestial_dir ra="128.73" dec="-45.2"/>
    </spectrum>
</source>
```

If broken power-law, ebreak and gamma2 tags are accepted

  - **N.B. this is not really Vela**
  - **General rule: fluxes are integrated over the energy range specified and are in m^{-2} s^{-1}**
- **This is one of the originally-developed sources. Most of the new sources are 'SpectrumClass' and have a simpler but less readable specification**

More on source model specifications

- **Example 'generic' source with a parameter list:**

```
<source name="periodic_source">
    <spectrum escale="MeV">
        <SpectrumClass name="PeriodicSource" params="0.1, 2.1, 1e3, 1, 0.75,
         30., 2e5"/>
    <galactic_dir l="0" b="0"/>
    </spectrum>
</source>
```

- **Want to know what those parameters are? See the documentation at the FSSC**

  http://fermi.gsfc.nasa.gov/ssc/data/analysis/scitools/help/gtobssim.txt

- **Options – Extended sources**
  - **GaussianSource (2-dim gaussian + power-law)**
  - **MapSource (2-dim image + power-law spectrum)**
  - **MapCube (3-dim image – 2 spatial one spectral)**
  - **Isotropic (power-law)**

More on source model specifications (2)

- **Options – Time-varying sources**
  - **SimpleTransient – Point source with power-law spectrum and a finite 'on' time**
  - **SpectralTransient – Point source with broken power-law or curved spectrum and detailed light curve – can handle multiple sources and can attenuate according to redshift and its own idea of the Extragalactic Background Light**
  - **PeriodicSource – Point source with power-law spectrum and sinusoidal variation of flux**
  - **Pulsar – Simple pulsar simulation, with P, Pdot, power-law spectrum, and user-specified light curve , does not 'decorrect' arrival times (so you do not need to run gtbary for analysis)**

About time

- **For the time-dependent sources time is expressed in MET, Mission Elapsed Time, which is the number of seconds elapsed since midnight, January 1, 2001 (MJD=51910).**
  - **Right now is approximately 360356000**
    - **Yes, MET is inconvenient to use directly***
  - **The start_date parameter of gtobssim allows you to specify your own reference date, so that the source specifications can be made with conveniently small time offsets**
  - **I would not hold my breath that the conversion is handled carefully enough to be accurate at the leap second level if for some reason your simulation depends on absolute time**

* N.B. xTime tool at HEASARC for time conversions:
http://heasarc.gsfc.nasa.gov/cgi-bin/Tools/xTime/xTime.pl?

Outputs from gtobssim

- FT1 files usable with the Science Tools
- The FT1 files include 2 columns not found in the science data
  - MC_SRC_ID for each γ-ray identifies the source that it came from
  - You can also filter on this column, e.g., with the fselect FTOOL
  - MCENERGY for each γ-ray provides the true energy (as opposed to the measured energy in the ENERGY column)
- The corresponding file <prefix>_srcIds.txt has the mapping of MC_SRC_ID number to source name from the input XML source specification
  - For each source it also lists the numbers of generated and accepted γ-rays

Walk through the parameter file (1)

Any parameter with an 'h' is 'hidden' (not prompted for)

```
infile,f,a,"none",,,"File of flux-style source definitions"
srclist,fr,a,"source_names.txt",,,"File containing list of source names"
scfile,f,a,"none",,,"Pointing history file"
sctable,s,h,"SC_DATA",,,"Spacecraft data extension"
evroot,s,a,"test",,,"Prefix for output files"
evtable,s,h,"EVENTS",,,"Event data extension"

simtime,r,a,86400,,,"Simulation time (seconds)"
ltfrac,r,h,0.9,,,"Livetime fraction"
tstart,r,a,INDEF,,,Simulation start time (seconds wrt MET 0)
nevents,b,h,no,,,Use simulation time as number of events
maxtime,r,h,3.155e8,,,Maximum simulation time (seconds)
startdate,s,h,"2001-01-01 00:00:00",,,"Mission start"
```

- **infile can be a file that is a list of other XML files, too; "none" means use the default libraries [flux-style is LAT team jargon]**
- **srclist is the file that lists the sources to be included from the libraries**
- **scfile is the FT2 file**
- **If you want to you can simulate by # of events (instead of time) (but that is not commonly done)**

Walk through the parameter file (2)

Indispensible if you are combining simulations

```
offset,i,h,0,,,"Source ID offset"
rockangle,r,h,INDEF,,,Rocking angle (degrees)

use_ac,b,a,no,,,"Apply acceptance cone?"
ra,r,a,0,-360,360,"RA of cone center (degrees)"
dec,r,a,0,-90,90,"Dec of cone center (degrees)"
radius,r,a,20,0,180,"Acceptance cone radius (degrees)"

emin,r,h,0,,,"Minimum event energy (MeV)"
emax,r,h,1e6,,,"Maximum event energy (MeV)"
edisp,b,h,yes,,,"Apply energy dispersion?"

irfs,s,a,"P7SOURCE_V6",,,"Response functions"
area,r,h,1,,,"LAT cross-sectional area"

maxrows,i,h,1000000,,,"Maximum number of rows in FITS files"
seed,i,a,293049,,,"Random number seed”

chatter,        i, h, 2, 0, 4, "Output verbosity"
clobber,        b, h, yes, , , "Overwrite existing output files"
debug,          b, h, no, , , "Activate debugging mode"
gui,            b, h, no, , , "GUI mode activated"
mode,           s, h, "ql", , , "Mode of automatic parameters"
```

- **An ‘acceptance cone’ and energy range can be specified if you want to, and the maximum number of events per file [but this does not save any time]**
- **Setting gui to yes will tell the science tools to prompt for the parameter files values via a GUI**

Advice and gotchas

- Again: units of flux are photons m^{-2} s^{-1} for historical reasons
- For diffuse sources (maps and cubes) the integral flux is specified in the XML file
  - The units of the FITS map or cube used to describe the source do not matter at all - the integral is renormalized at run time based on the integral flux specified in the XML file
- In order to not introduce artificial roll-offs of spectra at the extremes of the energy range simulated, you should specify a broader energy range for the simulation than you plan to analyze, and use gtselect to trim to the desired range
- If you are simulating a real analysis do not forget to run the same gtmktime selection that you ran for your real FT1 file
- If you are making multiple simulations of the same field do not forget to change the random seed for each run

Advice and gotchas (2)

- Simulating residual cosmic rays at the gtobssim level is not feasible except via the approximation that they are part of an isotropic source
  - That is, you need to assume that the acceptance is the same for residual background as for gamma rays (as we do for high-level analysis)
- Similarly, if you generate data with a given diffuse emission model and then use that same model in your likelihood analysis, then for better or worse your model will be perfect

Advice and gotchas (3)

- Depending on what you want to study you could consider making separate simulation runs for different components of the model
  - Specify an MC_SRC_ID offset so the sources do not overlap in ID number
  - Then just list the FT1 files as inputs to subsequent analysis steps
- Another strategy is to run a simulation with an existing (real) FT2 file and then combine the FT1 file with the FT1 file for the corresponding time interval in the flight data
  - You will want to be sure that both files have the same GTI from gtmktime
  - N.B. By design gtobssim does not set EVENT_CLASS which for flight data is a bit field for the mostly-nested event classes: it is YOUR responsibility to not mix event classes and IRFs between simulation and analysis
- Play with gtobssim