Science tool development directions

Science tool development directions

This page was started 31 October 2006; further comments added 22 Nov 2006 by SD; a Summary of Science Tool development directions was written on 23 Jan 2007

Inquiring minds would like to know what directions the science tools will take over the year remaining before launch (or the ~2 years before tools are delivered to guest investigators). The science tools in question for this discussion are those that will be part of the **standard set** that will be made available to guest investigators, the 'SAE'. The idea is to define what is **needed** and what is **desirable**, and **what fits within the time/person-power constraints**.

In the near term feedback from the GUC beta test will undoubtedly influence development work, although I'm not forecasting any major overhauls.

General

DLB (11/26/06)--In the beta test, the first activity after the users extracted was look at the data spatially and temporally. Consequently, users relied heavily on both fv and ds9. Our files lend themselves to this, and in the documentation we describe rudimentary uses of these tools to look at the data, but perhaps we should think a bit more about the interface to these tools.

Likelihood analysis

Some topics come to mind, pointed observations being one of them.

DLB (11/26/06)--We have discussed developing a new version of ModelEditor that edits and creates the XML model files for both likelihood and observation simulations. Since good source models are crucial for the likelihood analysis of any field that is not dominated by a single bright source, we must supply users with a robust, powerful version of ModelEditor. Also, I think we will find that users will want to create a simulated set of counts using gtobssim, which they will then analyze with likelihood, and for this they will want to use the same source model for both simulation and analysis.

DLB (11/26/06)--Do we understand what TS means, i.e., what the detection significance really is for a given TS? I know what it is supposed to be, but I recall that Jim did some simulations that showed that a detection was more significant for a given TS than theory said it should be.

DLB (11/26/06)--It would be useful to create and post running time benchmarks for various tools in the likelihood suite (this was suggested at the beta test). This would help users decide whether to do a binned or unbinned run. They would also be able to estimate how long their computer will be chugging away.

(Julie McEnery Nov 27)-- It would be handy to have a way to calculate an upper limit. I think that this will eventually be needed. (**Jim 11/29/06**: Computing an upper limit will, of course, require first understanding the null distribution, i.e., the TS distributions that David refers to above.)

Jim Chiang (11/29/06)--It has been proposed by Jean Ballet to compute a hybrid likelihood that combines an unbinned analysis for higher energy events that have better PSFs and are sparser, and a binned analyis for lower energies, where the PSF is not as good and binning would help speed up the execution without significant loss of accuracy. Alternatively, we could (also) implement a HEALPix binning scheme, based on Toby's work, that would provide similar computational savings.

Jim Chiang (11/29/06)--With regard to handling pointed observations, generalizing the Psf integrals to include a time-dependent zenith angle cut may turn out to be computationally intractable. As an alternative, Julie suggested making a cut on acceptance as a function of inclination angle wrt to the instrument z-axis. This would be straight-forward to implement in the exposure calculations, except that the cuts on the events would be applied to the **measured** inclinations, whereas our IRFs are defined as a function of **true**inclination. An alternative that would not make this sort of approximation can actually be applied within the current set of ScienceTools: Define a set of GTIs such that the instrument z-axis is within a specified angle of the spacecraft zenith. For pointed mode, this will likely decrease the on-source exposure time substantially, but this is a procedure that can be applied now.

GRB analysis

Is the current complement of tools complete?

DLB (11/26/06)--We should develop some XSPEC functions that parameterize the spectra expected in the GeV range. At the very least, we could use a version of a power law that can be normalized to an energy in the LAT range.

DLB (11/26/06)--Treatment of unbinned counts needs to be added to gtburstfit. Jeff Scargle has the methodology, which needs to be implemented.

(Julie McEnery Nov 27)-- I think that for brighter bursts we may need to account for variations in livetime fraction on timescales shorter than 30 s, or at least convince ourselves that this does not need to be done. If we don't do this, then the peak to peak flux measurements may turn out to be incorrect and bias joint spectral fits.

Pulsar analysis

From Masa: "In the pulsar tools area, the major works are:
1) to implement blind search tool (A4 tool),
2) to create a new tool to plot a pulse profile (w/ periodicity test results),
3) and to introduce a method that contains the full functionality of the tool, so that a software developer (or a Python script) can call it as a part of their software."
[He has updated the current status page for pulsar tools with these items, including some more details.]

"Another major task is to develop a system to ingest and distribute pulsar ephemerides database. Other than that, we have several minor improvements in mind, such as, improving gtephcomp output, implementing more time systems, and technical refactoring for easy maintenance."

Observation simulation

Will an orbit/attitude simulator with at least semi-realistic attitude profiles/knowledge of constraints really be part of the SAE? Should we assume that pointing history files will be made available (or generated on request) for various scenarios?

Are any important source types missing? Is simulating residual background at the gtobssim level important, and is it feasible?

Jim Chiang (11/29/06)--A few months ago, I proposed a gtobssim-like tool that would take a livetime cube as input, thereby avoiding the necessity of querying for the arrival time of each incident photon and then processing individually through the IRFs. This tool would not use the flux package and so would need a new infrastructure for folding the incident source fluxes through the instrument response. In addition, for the interval covered by the input livetime cube, all sources would be steady and the variation of exposure to a given location on the sky would not be imprinted on the data.

Infrastructure

GUI(s)?

Utilities

In the past, I've argued that we need a utility in the SAE for examining/displaying the IRFs.

Question: Did anyone else try out the event display tool during DC2? It was impressive and fun to play with, but I didn't *need* it.

(Julie McEnery, Nov 27) It would be very nice to have tools to correctly merge event data files. This cropped up as a common need in DC2 and more recently when finishing up the 1 year gtobssim run. To start with it might be nice if gtselect could accept file lists, although I appreciate that a final solution (to correctly take care of GTI) may need to be more sophisticated.

Jim Chiang, 11/29/06--I have a python script that I use in ASP that I believe does this merging of FT1 files correctly using the **fmerge** FTOOL: ft1merge.py As the comments in this code indicate, this script assumes that input data are partitioned in time and that all other DSS cuts are the same among the files. Given that **fmerge** already exists as an FTOOL and that a such a simple script using **fmerge** can do the job, I think it is worth reconsidering whether all ScienceTools run from the command line need to be compiled st_apps. After all, standard FTOOLs contain perl scripts that perform much the same kind of functionality in terms of driving other standalone FTOOLs.

Other issues

Delivery of science tools to the GSSC

In terms of delivery, a long time ago the SSC-LAT working group, or whatever we called ourselves, declared that it would be the group that decided when a given tool was 'ready' for delivery. I think that we probably don't need to re-convene the group, but I'd like your opinions about whether the tools should pass some not-yet-written battery of tests beyond the unit tests for the packages before they are accepted. And whether, during the mission the GSSC will be issuing incremental releases of the SAE tools at the same rate that the LAT team 'delivers' them.

Real life

I think that Jim has generalized the IRF look up to allow for time dependence of the IRFs. I don't know how likely we are to need to have response functions be time dependent - e.g., owing to something like a hardware failure - but at least in principle we could want to make analyses (with Likelihood or gtrspgen) that span these changes. The only obvious problem would be with live time cubes.

Also, we'll need to figure out how we'll *really* assign ID numbers to events.

How we'll handle the residual backgrounds in the data is still being grappled with. Even the 'irreducible' component is not all that small at low energies. The orbit and attitude dependence of the background (and residual background) complicates modeling, but probably we should deliver some sort of reliable model for residual backgrounds just as we deliver a model of the diffuse gamma-ray emission.

DLB (11/26/06)--I don't know whether we've ever made an official decision on this, but I think we should recognize that Mac OS X must now be one of the supported platforms.

DLB (11/26/06)--We need to automate and speed up the creation of builds on all platforms. It took a long time to get the SAE running on all the different platforms for the beta test, and we ended up with various tools not working on different platforms. Half our beta testers had Macs, yet we did not have the time to test the Mac version as thoroughly as we'd have liked.

Live time and pointing history

The 'accumulated livetime since start of mission' is looking quite difficult to obtain - at one time it was going to be easy. I think that the need for it is sufficiently small that we should consider omitting it from the FT1 file.
The FT2 file will need to continue to have accumulated live times for each interval of time, for exposure calculations. The attitude and position information that we'll get from the spacecraft, and will want to use for L1 processing, are not in anything like FT2 format - among the differences are the frequency of updates (much greater), the use of quaternions, the availability of angular velocities, and the asynchronicity of the attitude and position information. Do we want to change the FT2 format to more closely relate to what comes in the telemetry?

The short answer is no, if only because the files would be very large and not offer any useful advantages in terms of, say, accuracy of the exposure calculation.

I wonder sometimes what the GBM is doing regarding position/attitude information; their FT2 equivalent uses quaternions, but I think that position and atitude are interpolated to the same point in time.

Time Series

DLB (11/26/06)--The lightcurve functionality in gtbin just bins the counts in an FT1 file in time. The lightcurve therefore does not compensate for the exposure. For those of us who look at gamma-ray burst lightcurves this is OK; indeed to plan further analysis this is what we want. However, those looking at sources on longer timescales (i.e., over orbits) may want to correct for exposure; AGN people have requested this. For the counts from a point source this makes sense, but this can be problematic when the counts originate from a large region. So...should we create/add exposure correction?

Jim Chiang (11/29/06)--For my DC2 analysis of the Solar flare, I wrote an exposure tool that does such an exposure correction as a function of time at a specific location on the sky. For the Solar flare, I made a comparison of the light curve obtained using gtbin and the output of this exposure tool versus the flux estimates from a full likelihood fit of the fluxes. At least for bright sources such as the DC2 Solar flare, the comparison was quite favorable. See slides 8 and 9 of my DC2 close-out talk. I will make this exposure tool into a proper ScienceTool.

DLB(11/26/06)--gtbin was meant to be a simple binning tool, i.e., accumulating counts into energy/temporal/spatial bins. It does not calculate uncertainties for the number of counts. However, showing uncertainties would be appropriate in displaying the products, particularly when the counts are divided by livetime or exposure, or undergo some other transformation. So, how should we proceed?

XSPEC analysis of strong sources

DLB (11/26/06)--Since strong sources may be analyzed with XSPEC, we might want to create XSPEC functions suited for ~GeV range. For example, it would be useful to have a version of the standard power law, broken power law, etc., functions normalized to a GeV. When data at a GeV is fit by functions normalized at 1 keV, the normalization is highly correlated with the other parameters

SAE Enhancements From the GSSC beta test

These are comments compiled by Dave Davis from members of the GLAST Users Committee who participated in the GUC Beta Test in November.

DS9 regions:
gtselect should be able to use ds9 regions for selection
and the SAE tools should recognize ds9 region files.
Proposed method:
Use gtbin to make a projected map
ds9 can then be used to select and exclude regions.
Is there a library to convert .reg files --> DSS regions?
do we need to translate the projected circle --> spherical
circle?
Is this sufficient?

Default values:
gtselect should add keywords that downstream programs can use for
default values. Maybe this should be the DSS keywords if we can
determine which set to use if there are multiple dss
keywords. Alternately we might be able to use the RA_FOV and
DEC_FOV to set the field center for later programs.
1) How do other FTOOLS handle this?

- keywords?
  2) How to impliment
- INDEF
- DEFAULT

The Tools need to start with reasonable defaults.
e.g. it has been suggested that for gtlikelihood
gui=yes and save output file should be on by default.
1) What are reasonable values?

- Most FTOOLS start with the last values of when the tool
  was run and do not inherit from the previous tool but they
  do read header info.

Another way to make the imputs more reasonable" is to make them
more compact so that the user can reuse parts of the command
file. One method would be to use the fselect expression
interface. This would allow queries like
"binary && mag <= 5.0" or to be more glast specific
"energy > 100. && ZENITH_ANGLE <20."
This also allows one to use the region file format
regfilter("region.reg", XPOS, YPOS)
and it allows flexible gti selections.
gtifilter("", TIME, "**START**", "**STOP**")

Parameters names should be consistent between the tools. This should
include the GUI.
1) Who should do this and how should it be split up
2) Present this at a science tools meeting

Names should be shorter where possible
gtlivetimecube -> gtltc , gtexp ...?
1) suggestions for names?
What should we consider tool long (> 8 characters)
2) Links for the programs

- How to handle parameter names

lightcurve analysis needs to be revisited and mapped out.
how to use exposure corrected data
how to use regions to exclude nearby sources.
Develop threads
1) What has already been done
quicklook analysis
publication quality analysis
2) Can we adapt existing scripts?

- evaluation the amount of work.

map analysis:
psf generation for a given source and spectral fit.
1) compare with source distribution
2) PSF subtraction
3) image restoration/deconvolution
Issues:
How to use the fitted parameters from gtlikelihood or Xsepc
a) read the xml fit parameters or the xspec fit parameters
b) convert to some fits format?
Need both 1-d psf to compare radial distribution and a 2-d psf
for psf subtraction/deconvolution

pulsar analysis:
Energy dependent and possibly PSF dependent cuts.
Overwriting the TIME column for the tools is not
optimal. (Change name? TIME_ORIG)

All threads need to be brought up to date.

Reference documentation needs significant updates.

GUI interface for the tools.
This is rather overarching.
1) what tools need to be in the GUI
2) What type of GUI
root gui's
xselect type (or even use xselect?)

The current tools GUI's needs to be refined. E.g. the save and "should I overwrite"
prompts need to be clearer.

XSPEC-like analysis should be explored.
The ability to get a Counts or "pseudo-counts" spectrum with an
appropriate rsp matrix would facilitate multi-mission analysis.