Spectral Analysis of GRB 090926181: Data Processing and Fitting Methodology

Gamma-Ray Burst Analysis Pipeline

October 22, 2025

Abstract

This document explains the methodology implemented in `analisis_espectral_multiple.ipynb` for the spectral analysis of Gamma-Ray Burst (GRB) 090926181 using Fermi-GBM data. The notebook covers lightcurve creation and visualization, background estimation, and spectral fitting using the Band function. The analysis combines data from multiple detectors (NaI and BGO) to achieve comprehensive spectral characterization of the burst.

1 Introduction

The analysis focuses on GRB 090926181, a gamma-ray burst observed by the Fermi Gamma-ray Burst Monitor (GBM). The notebook processes CSPEC (Coded Spectral) data from multiple detectors to extract temporal and spectral information. The methodology involves:

- Loading and organizing multi-detector data
- Creating energy-resolved lightcurves
- Background estimation and subtraction
- Spectral extraction and fitting with physical models

2 Data Loading and Organization

2.1 File Structure and Path Configuration

The analysis begins by establishing the data paths for the GRB event:

```
object_no = '090926181'
object_name = f'bn{object_no}'
common_str = f'datos/{object_no}/glg_cspec_'
filepaths = [
    f"{common_str}b0_{object_name}_v00.pha", #0
    f"{common_str}n0_{object_name}_v00.pha", #1
    f"{common_str}n1_{object_name}_v00.pha", #2
    f"{common_str}n3_{object_name}_v00.pha", #3
    f"{common_str}n6_{object_name}_v00.pha", #4
    f"{common_str}n7_{object_name}_v00.pha"  #5
]
```

The code loads CSPEC data from multiple detectors:

- **b0**: BGO detector (high-energy range, 325-35000 keV)
- **n0, n1, n3, n6, n7**: NaI detectors (low-energy range, 8-900 keV)

2.2 Data Objects

The CSPEC files are loaded as `GbmPhaii` objects, which contain time-tagged photon counts organized in energy channels:

```
from gdt.missions.fermi.gbm.phaii import GbmPhaii
cspec_b0 = GbmPhaii.open(filepaths[0])
cspec_n3 = GbmPhaii.open(filepaths[3])
cspec_n6 = GbmPhaii.open(filepaths[4])
cspec_n7 = GbmPhaii.open(filepaths[5])
```

3 Lightcurve Creation and Visualization

3.1 Energy-Resolved Lightcurves

The notebook creates lightcurves in different energy ranges to study the temporal evolution of the burst across the spectrum:

```
# Define energy ranges
range1 = (8.0, 50.0)    # Low energy
range2 = (50.0, 300.0)  # Medium energy
range3 = (300.0, 900.0) # High energy
t_r = (-33, 102)        # Time range

# Create lightcurves for each range
lc_data1 = cspec_n3.to_lightcurve(time_range=t_r, energy_range=range1)
lc_data2 = cspec_n3.to_lightcurve(time_range=t_r, energy_range=range2)
lc_data3 = cspec_n3.to_lightcurve(time_range=t_r, energy_range=range3)
```

The `to_lightcurve()` method integrates photon counts over specified energy bins and time intervals, creating count rate vs. time data. This allows for:

- Identification of different emission episodes
- Spectral evolution analysis (hard-to-soft behavior)
- Determination of optimal time intervals for spectral analysis

3.2 Multi-Detector Collection

To facilitate simultaneous analysis of multiple detectors, the data is organized into a `GbmDetectorCollection`:

```
from gdt.missions.fermi.gbm.collection import GbmDetectorCollection
cspecs = GbmDetectorCollection.from_list([cspec_n3, cspec_n6, cspec_n7, cspec_b0])
```

4 Background Estimation

4.1 Background Selection

Background radiation is estimated from time intervals before and after the burst:

```
view_range = (-50, 100)  # Analysis window
bkgd_range = [(-50, -10), (30, 100)]  # Background intervals
energy_range_nai = (8, 900)     # NaI energy range
energy_range_bgo = (325, 35000) # BGO energy range
src_range = (1, 2)              # Source interval for spectral analysis
```

4.2 Background Fitting

A polynomial background model is fitted to the background intervals:

```
from gdt.core.background.fitter import BackgroundFitter
from gdt.core.background.binned import Polynomial

# Initialize background fitters for each detector
backfitters = [BackgroundFitter.from_phaii(cspec, Polynomial,
                time_ranges=bkgd_range) for cspec in cspecs]
```

```
backfitters = GbmDetectorCollection.from_list(backfitters, dets=cspecs.detector())

# Perform 0th order polynomial fit
backfitters.fit(order=0)
```

The background fit quality is assessed using the reduced χ^2 statistic:

$$\chi^2_\nu = \frac{\chi^2}{\text{dof}} = \frac{\sum_i \frac{(O_i - E_i)^2}{\sigma_i^2}}{N - p} \tag{1}$$

where O_i are observed counts, E_i are expected counts, σ_i are uncertainties, N is the number of data points, and p is the number of fit parameters.

4.3 Background Interpolation

The fitted background is interpolated across the entire time range:

```
# Interpolate background fits
bkgds = backfitters.interpolate_bins(cspecs.data()[0].tstart,
                                     cspecs.data()[0].tstop)
bkgds = GbmDetectorCollection.from_list(bkgds, dets=cspecs.detector())
```

5 Spectral Analysis

5.1 Spectrum Extraction

Count spectra are extracted for the source time interval, with background subtraction:

```
# Extract count spectra
data_specs = cspecs.to_spectrum(time_range=src_range)
# Extract time-integrated background
bkgd_specs = bkgds.integrate_time(*src_range)
# Apply energy selection
src_specs = cspecs.to_spectrum(time_range=src_range,
                               nai_kwargs={'energy_range': energy_range_nai},
                               bgo_kwargs={'energy_range': energy_range_bgo})
```

The spectra are converted to PHA (Pulse Height Analyzer) format for spectral fitting:

```
phas = cspecs.to_pha(time_ranges=src_range,
                     nai_kwargs={'energy_range': energy_range_nai},
                     bgo_kwargs={'energy_range': energy_range_bgo})
```

5.2 Response Matrix Loading

Instrument response matrices (RSP2 files) account for detector efficiency and energy resolution:

```
from gdt.missions.fermi.gbm.response import GbmRsp2

# Load response files
rsp_n3 = GbmRsp2.open(filepaths_rsp[0])
rsp_n6 = GbmRsp2.open(filepaths_rsp[1])
rsp_n7 = GbmRsp2.open(filepaths_rsp[2])
rsp_b0 = GbmRsp2.open(filepaths_rsp[3])

rsps = GbmDetectorCollection.from_list([rsp_n3, rsp_n6, rsp_n7, rsp_b0])

# Interpolate at spectrum central time
rsps_interp = [rsp.interpolate(pha.tcent) for rsp, pha in zip(rsps, phas)]
```

6 Spectral Fitting

6.1 Band Function

The Band function is used to model the GRB spectrum:

$$N(E) = A \begin{cases} \left(\frac{E}{E_{\text{pivot}}}\right)^{\alpha} \exp\left(-\frac{E}{E_0}\right) & E \leq (\alpha - \beta)E_0 \\ \left[\frac{(\alpha-\beta)E_0}{E_{\text{pivot}}}\right]^{\alpha-\beta} \exp(\beta - \alpha) \left(\frac{E}{E_{\text{pivot}}}\right)^{\beta} & E > (\alpha - \beta)E_0 \end{cases} \tag{2}$$

Parameters:

- A: Amplitude (ph s^{-1} cm^{-2} keV^{-1})
- α: Low-energy photon index
- β: High-energy photon index
- E_0: Characteristic energy (keV)
- $E_{\text{peak}} = (2 + \alpha)E_0$: Peak energy in νF_ν spectrum

6.2 Fitting Procedure

The spectral fitting uses the PGSTAT statistic (Poisson-Gaussian statistic appropriate for low-count regimes):

```python
from gdt.core.spectra.fitting import SpectralFitterPgstat
from gdt.core.spectra.functions import Band

# Initialize fitter
specfitter = SpectralFitterPgstat(phas, bkgds.to_list(), rsps_interp, method='TNC')

# Initialize Band function
band = Band()

# Perform fit
specfitter.fit(band, options={'maxiter': 1000})
```

The fitting process minimizes the PGSTAT statistic using the TNC (Truncated Newton) optimization algorithm. The fit quality is evaluated using:

- PGSTAT/DOF ratio (should be close to 1 for good fits)
- Parameter confidence intervals (90% asymmetric errors)
- Residual analysis

7 Results and Visualization

7.1 Lightcurve Analysis

The multi-energy lightcurves reveal:

- Main emission episode lasting approximately 20 seconds
- Spectral hardening during peak intensity
- Different temporal behavior in various energy bands

7.2 Spectral Fit Results

The Band function fit yields the following parameters (example values):

- Amplitude: $A = 8.54 \times 10^{-2}$ ph s^{-1} cm^{-2} keV^{-1}
- Peak energy: $E_{\text{peak}} = 291$ keV
- Low-energy index: $\alpha = -0.40$
- High-energy index: $\beta = -2.00$ (at upper bound)
- PGSTAT/DOF = 256.14/480

The high-energy index reaching its upper bound suggests the spectrum doesn't require a high-energy break within the observed range.

7.3 Model Visualization

The fitted model is plotted with data points and residuals:

```
from gdt.core.plot.model import ModelFit
modelplot = ModelFit(fitter=specfitter)
plt.ylim(1e-4, 200)
plt.xlim(7.15, 4000)
```

The plot shows:

- Count spectrum with statistical errors
- Fitted Band function model
- Data/model residuals to assess fit quality
- Combined NaI and BGO detector coverage

8 Conclusion

The spectral analysis notebook provides a comprehensive pipeline for GRB spectral analysis using Fermi-GBM data. Key strengths include:

- Multi-detector approach for broad energy coverage (8 keV - 35 MeV)
- Robust background estimation using polynomial fitting
- Physical spectral modeling with the Band function
- Statistical rigor using appropriate fitting methods

The methodology can be adapted for other GRB events and extended to include additional spectral models or temporal-resolved spectroscopy.