Project 5

X射线和伽马射线数据联合分析

杨俊、吴沁宇、孙惠

天关卫星时代的X射线天文学 (2025) @ 云南昆明

X-ray and gamma-ray missions

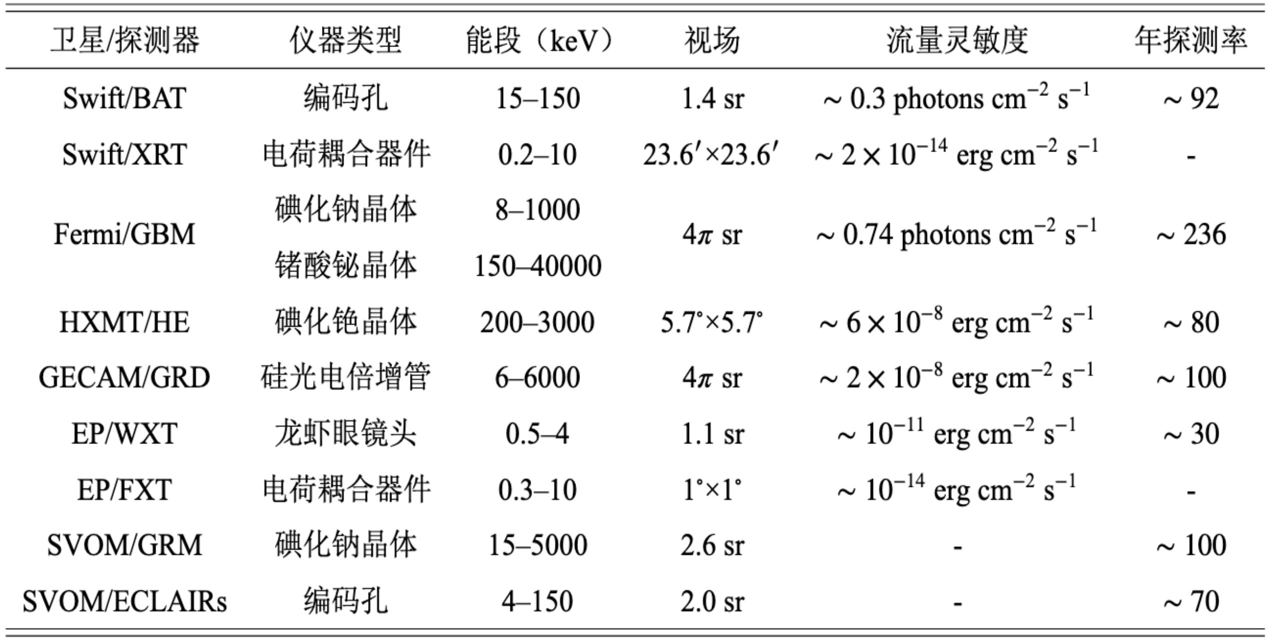
- Fermi Gamma-Ray Space Telescope (*Fermi*)
  - GBM (8 keV – 50 MeV) → prompt emission (236 per year)
  - LAT (GeV) → High-energy emission

- The Neil Gehrels Swift Observatory (*Swift*)
  - BAT (15 – 350 keV) → prompt emission (92 per year)
  - XRT (0.3 – 10 keV) → X-ray afterglow
  - UVOT (Ultraviolet and Optical) → afterglow

- Hard X-ray Modulation Telescope (*Insight*-HXMT)
  - HE (20 – 350 keV; 80 per year)
  - ME (8 – 35 keV)
  - LE (1 – 12 keV)

- Gravitational wave high-energy Electromagnetic Counterpart All-sky Monitor (GECAM)
  - B + C + D (6 keV – 5 MeV; > 70 per year)

- Einstein Probe (EP)
  - WXT (0.5-4.0 keV) + FXT (0.5-10 keV)



卫星/探测器	仪器类型	能段（keV）	视场	流量灵敏度	年探测率
Swift/BAT	编码孔	15–150	1.4 sr	~ 0.3 photons $\mathrm{cm}^{-2}\ \mathrm{s}^{-1}$	~ 92
Swift/XRT	电荷耦合器件	0.2–10	$23.6' \times 23.6'$	$\sim 2 \times 10^{-14}$ erg $\mathrm{cm}^{-2}\ \mathrm{s}^{-1}$	-
Fermi/GBM	碘化钠晶体	8–1000	4π sr	~ 0.74 photons $\mathrm{cm}^{-2}\ \mathrm{s}^{-1}$	~ 236
	锗酸铋晶体	150–40000			
HXMT/HE	碘化铯晶体	200–3000	$5.7^\circ \times 5.7^\circ$	$\sim 6 \times 10^{-8}$ erg $\mathrm{cm}^{-2}\ \mathrm{s}^{-1}$	~ 80
GECAM/GRD	硅光电倍增管	6–6000	4π sr	$\sim 2 \times 10^{-8}$ erg $\mathrm{cm}^{-2}\ \mathrm{s}^{-1}$	~ 100
EP/WXT	龙虾眼镜头	0.5–4	1.1 sr	$\sim 10^{-11}$ erg $\mathrm{cm}^{-2}\ \mathrm{s}^{-1}$	~ 30
EP/FXT	电荷耦合器件	0.3–10	$1^\circ \times 1^\circ$	$\sim 10^{-14}$ erg $\mathrm{cm}^{-2}\ \mathrm{s}^{-1}$	-
SVOM/GRM	碘化钠晶体	15–5000	2.6 sr	-	~ 100
SVOM/ECLAIRs	编码孔	4–150	2.0 sr	-	~ 70

Part 1: gamma-ray data analysis

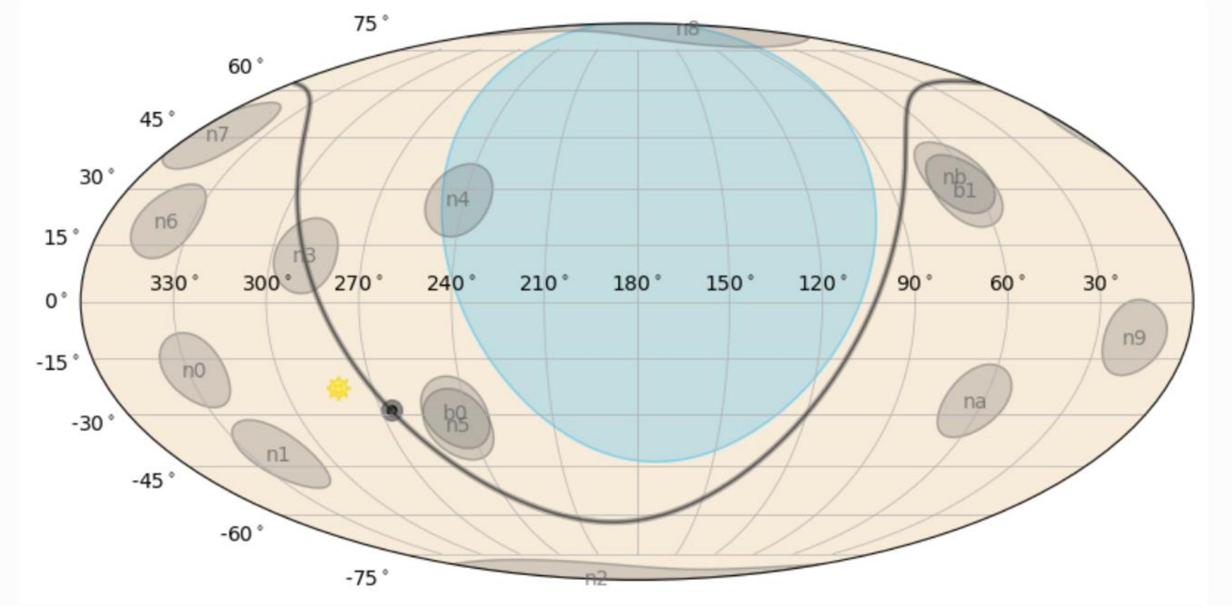
Gamma-ray data (e.g. Fermi/GBM)

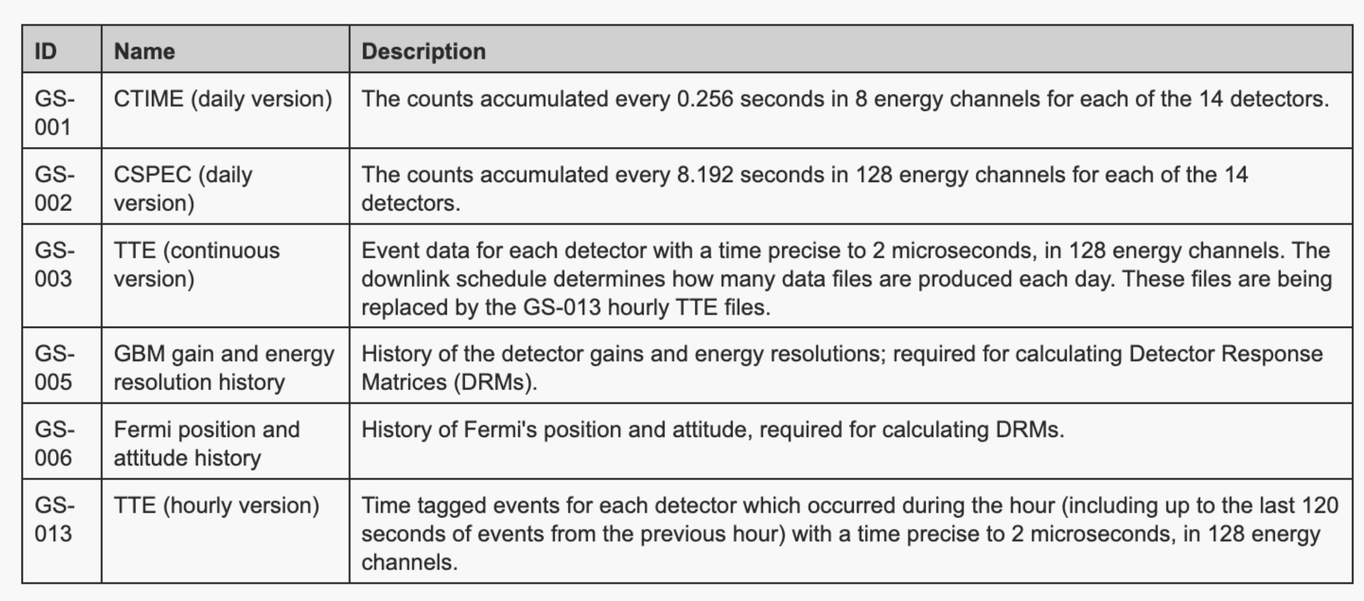
Gamma-ray: Fermi/GBM daily data


heasarc.gsfc.nasa.gov/FTP/fermi/data/gbm/daily/2024/02/19/current/

Index of /FTP/fermi/data/gbm/daily/2024/02/19/current

```
Name                           Last modified      Size  Description
Parent Directory                                     -
glg_cspec_b0_240219_v00.pha    19-Feb-2024 20:57  5.1M
glg_cspec_b1_240219_v00.pha    19-Feb-2024 20:58  5.1M
glg_cspec_n0_240219_v00.pha    19-Feb-2024 20:58  5.1M
glg_cspec_n1_240219_v00.pha    19-Feb-2024 20:58  5.1M
glg_cspec_n2_240219_v00.pha    19-Feb-2024 20:58  5.1M
glg_cspec_n3_240219_v00.pha    19-Feb-2024 20:58  5.1M
glg_cspec_n4_240219_v00.pha    19-Feb-2024 20:58  5.1M
glg_cspec_n5_240219_v00.pha    19-Feb-2024 20:58  5.1M
glg_cspec_n6_240219_v00.pha    19-Feb-2024 20:58  5.1M
glg_cspec_n7_240219_v00.pha    19-Feb-2024 20:58  5.1M
glg_cspec_n8_240219_v00.pha    19-Feb-2024 20:58  5.1M
glg_cspec_n9_240219_v00.pha    19-Feb-2024 20:58  5.1M
glg_cspec_na_240219_v00.pha    19-Feb-2024 20:58  5.1M
glg_cspec_nb_240219_v00.pha    19-Feb-2024 20:58  5.1M
glg_ctime_b0_240219_v00.pha    19-Feb-2024 20:58   11M
glg_ctime_b1_240219_v00.pha    19-Feb-2024 20:58   11M
glg_ctime_n0_240219_v00.pha    19-Feb-2024 20:59   11M
glg_ctime_n1_240219_v00.pha    19-Feb-2024 20:59   11M
glg_ctime_n2_240219_v00.pha    19-Feb-2024 20:59   11M
glg_ctime_n3_240219_v00.pha    19-Feb-2024 20:59   11M
glg_ctime_n4_240219_v00.pha    19-Feb-2024 20:59   11M
glg_ctime_n5_240219_v00.pha    19-Feb-2024 20:59   11M
glg_ctime_n6_240219_v00.pha    19-Feb-2024 20:59   11M
glg_ctime_n7_240219_v00.pha    19-Feb-2024 20:59   11M
glg_ctime_n8_240219_v00.pha    19-Feb-2024 20:59   11M
glg_ctime_n9_240219_v00.pha    19-Feb-2024 21:00   11M
glg_ctime_na_240219_v00.pha    19-Feb-2024 21:00   11M
glg_ctime_nb_240219_v00.pha    19-Feb-2024 21:00   11M
glg_poshist_all_240219_v00.fit 19-Feb-2024 21:05  8.8M
glg_spechist_b0_240219_v00.fit 19-Feb-2024 21:00   14K
glg_spechist_b1_240219_v00.fit 19-Feb-2024 21:00   14K
```





ID	Name	Description
GS-001	CTIME (daily version)	The counts accumulated every 0.256 seconds in 8 energy channels for each of the 14 detectors.
GS-002	CSPEC (daily version)	The counts accumulated every 8.192 seconds in 128 energy channels for each of the 14 detectors.
GS-003	TTE (continuous version)	Event data for each detector with a time precise to 2 microseconds, in 128 energy channels. The downlink schedule determines how many data files are produced each day. These files are being replaced by the GS-013 hourly TTE files.
GS-005	GBM gain and energy resolution history	History of the detector gains and energy resolutions; required for calculating Detector Response Matrices (DRMs).
GS-006	Fermi position and attitude history	History of Fermi's position and attitude, required for calculating DRMs.
GS-013	TTE (hourly version)	Time tagged events for each detector which occurred during the hour (including up to the last 120 seconds of events from the previous hour) with a time precise to 2 microseconds, in 128 energy channels.

Gamma-ray data (e.g. Fermi/GBM)

Gamma-ray: Fermi/GBM event data

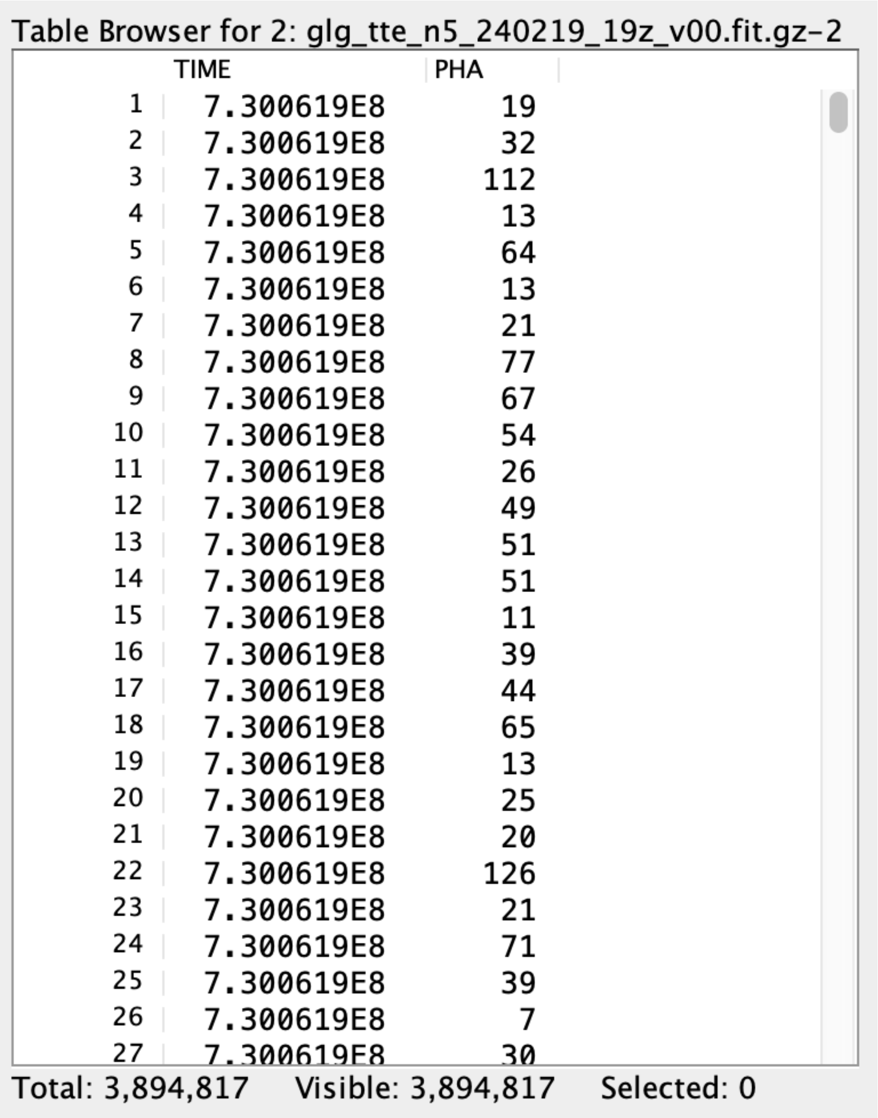


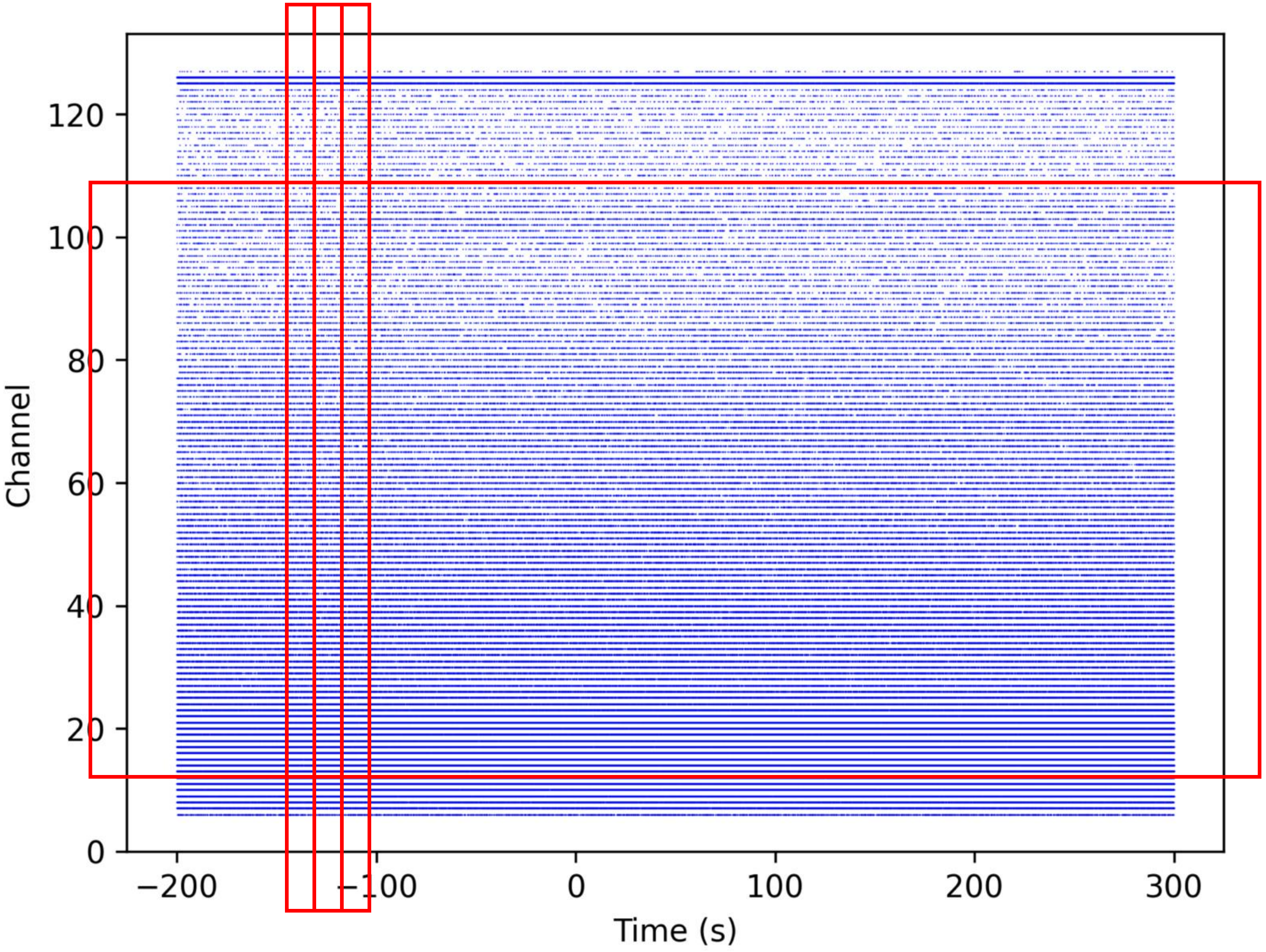
Table Browser for 2: glg_tte_n5_240219_19z_v00.fit.gz–2

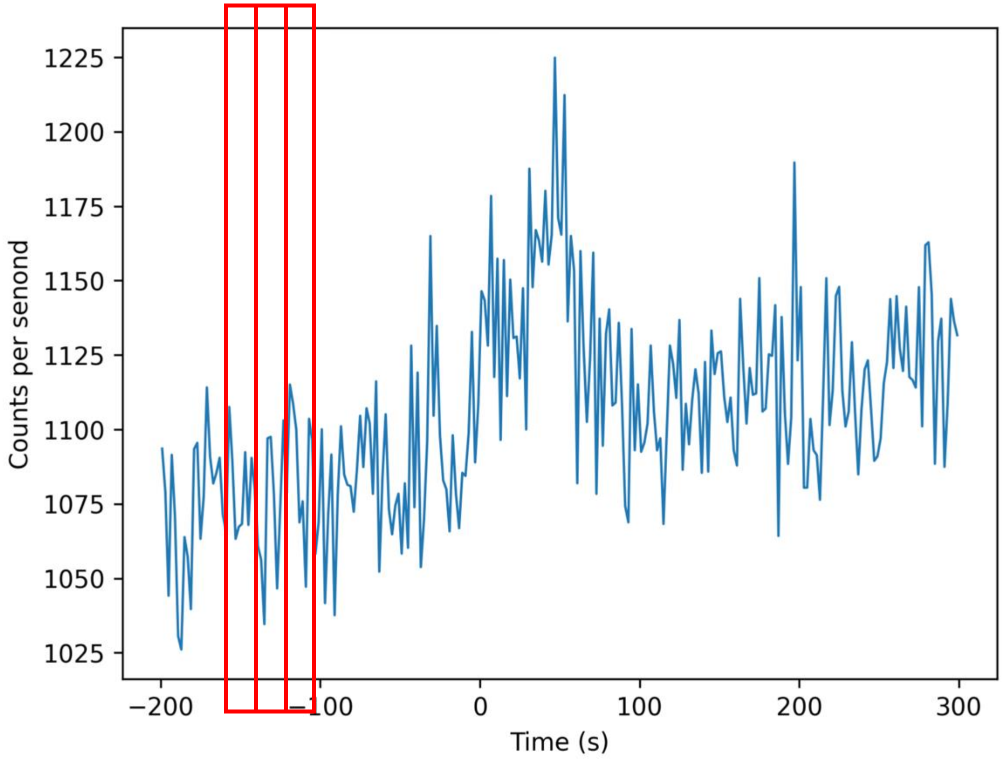
	TIME	PHA
1	7.300619E8	19
2	7.300619E8	32
3	7.300619E8	112
4	7.300619E8	13
5	7.300619E8	64
6	7.300619E8	13
7	7.300619E8	21
8	7.300619E8	77
9	7.300619E8	67
10	7.300619E8	54
11	7.300619E8	26
12	7.300619E8	49
13	7.300619E8	51
14	7.300619E8	51
15	7.300619E8	11
16	7.300619E8	39
17	7.300619E8	44
18	7.300619E8	65
19	7.300619E8	13
20	7.300619E8	25
21	7.300619E8	20
22	7.300619E8	126
23	7.300619E8	21
24	7.300619E8	71
25	7.300619E8	39
26	7.300619E8	7
27	7.300619E8	30

Total: 3,894,817 Visible: 3,894,817 Selected: 0

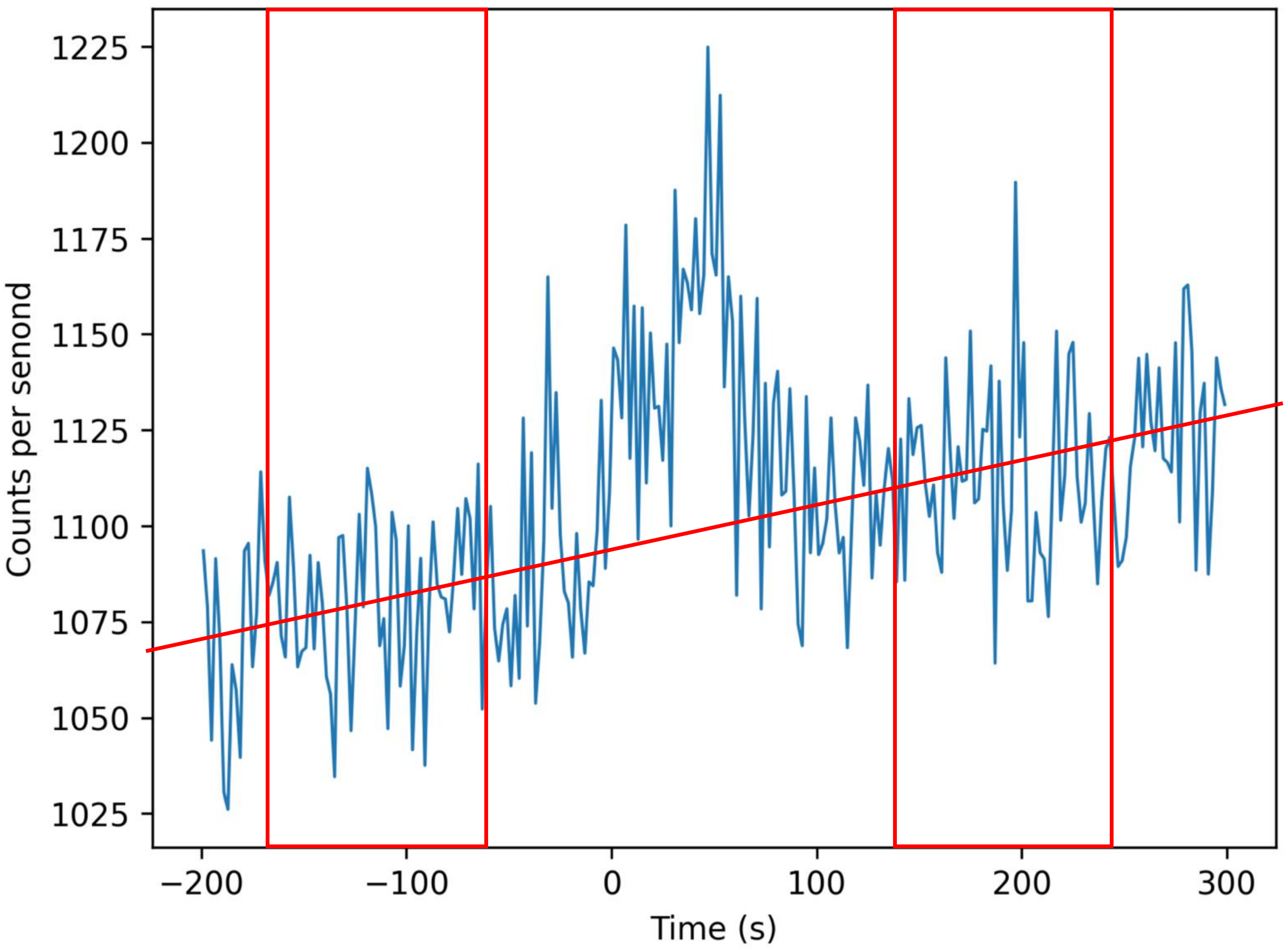
参数	类型	单位	解释
TIME	8 字节实数	s	光子事件的到达时间
PHA	2 字节整数	-	脉冲高度分析器通道编号

Gamma-ray data analysis (e.g. Fermi/GBM) — light curve

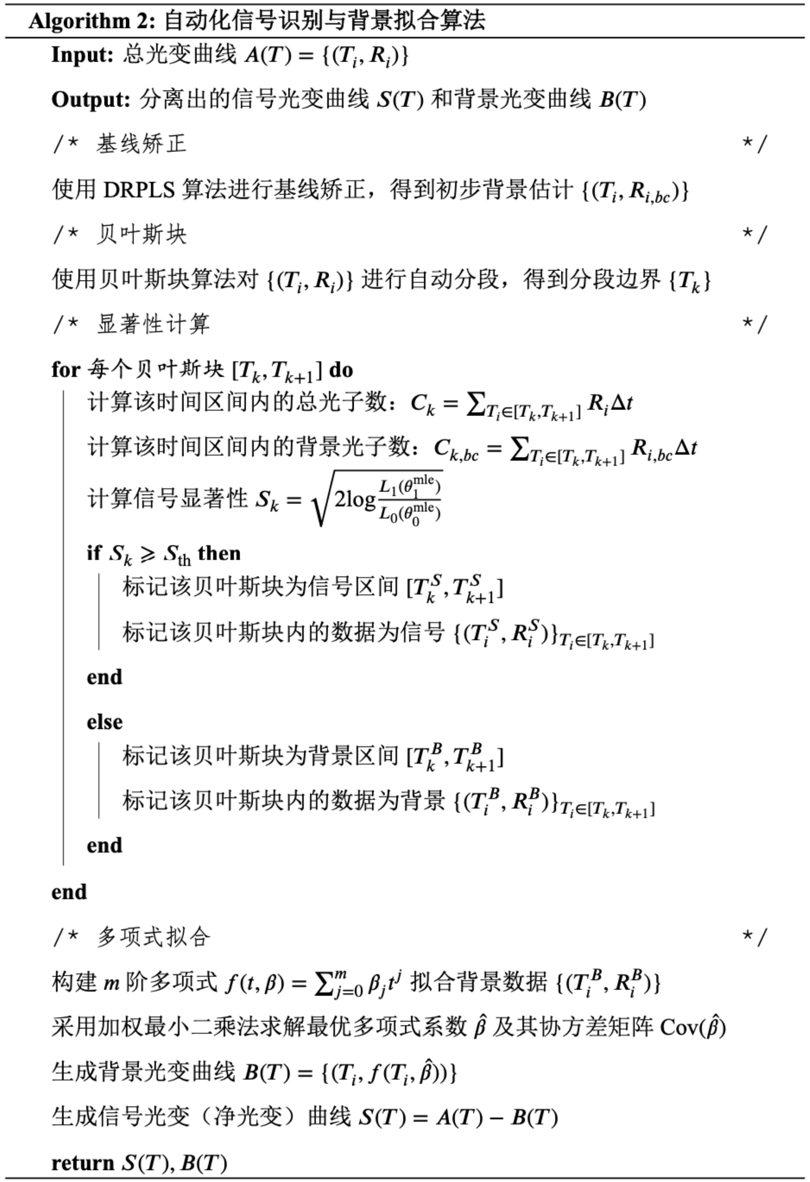




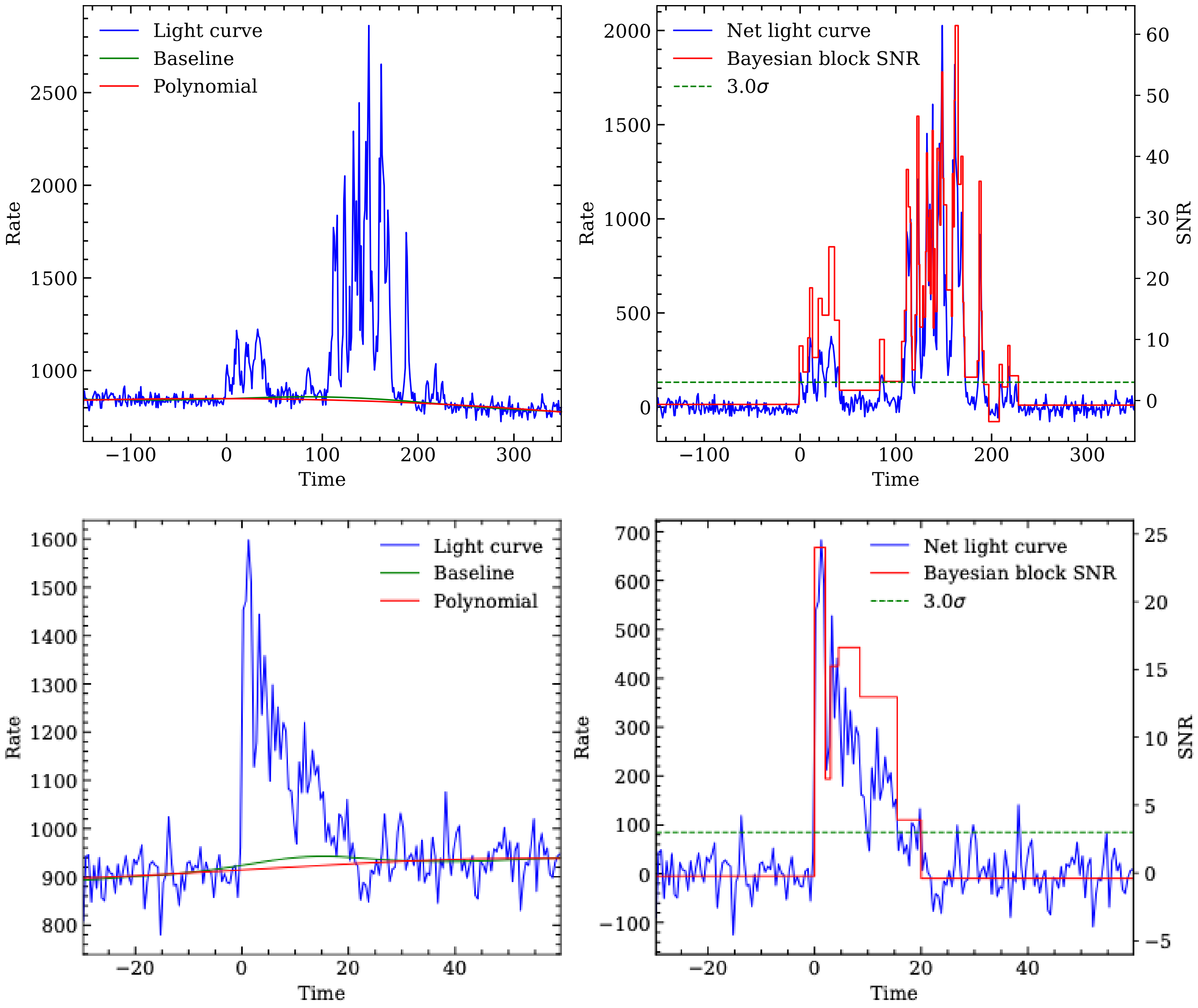
Gamma-ray data analysis (e.g. Fermi/GBM) — background



Gamma-ray data analysis (e.g. Fermi/GBM) — background



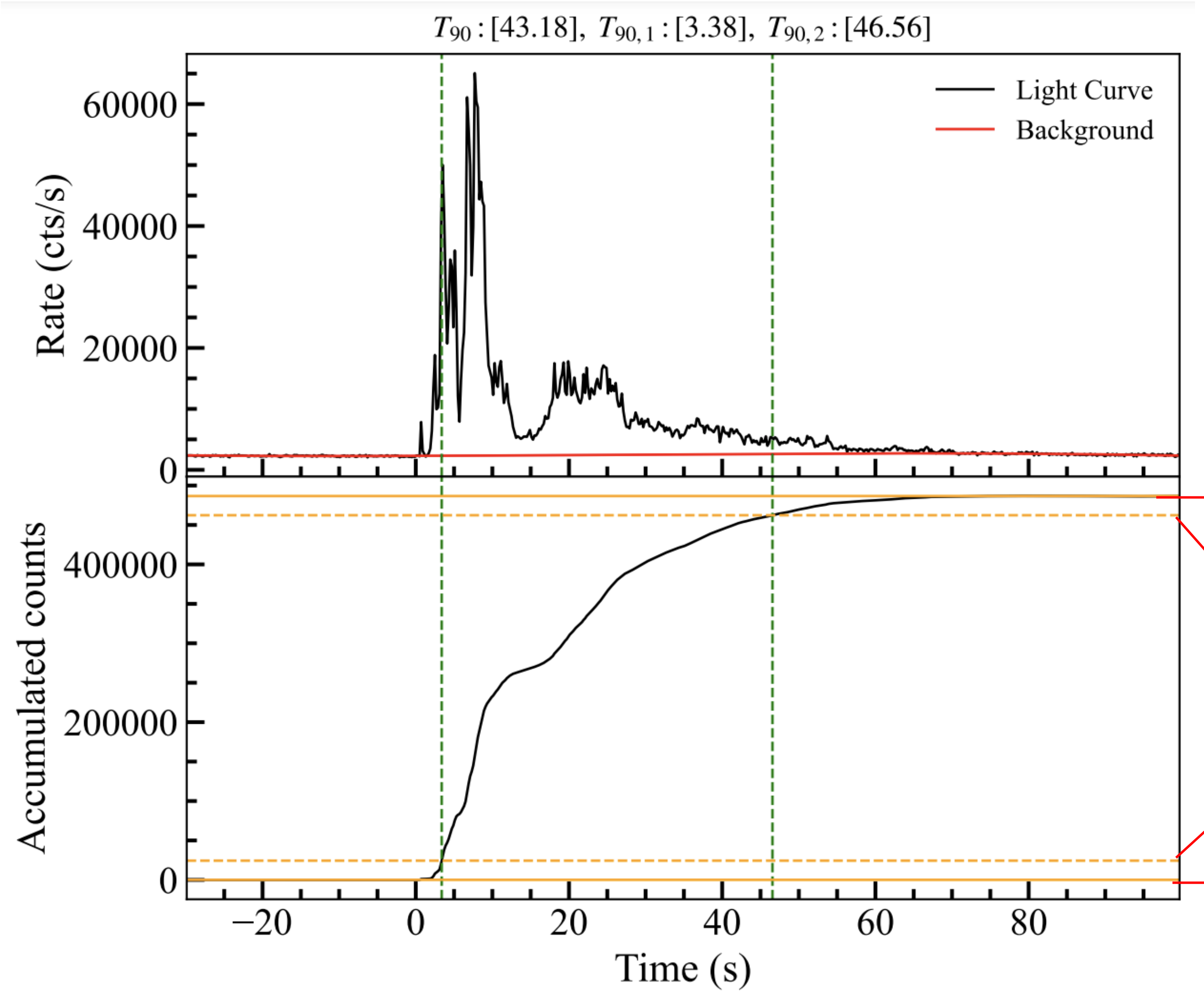
```
Algorithm 2: 自动化信号识别与背景拟合算法
Input: 总光变曲线 $A(T) = \{(T_i, R_i)\}$
Output: 分离出的信号光变曲线 $S(T)$ 和背景光变曲线 $B(T)$
/* 基线矫正                                                    */
使用 DRPLS 算法进行基线矫正，得到初步背景估计 $\{(T_i, R_{i,bc})\}$
/* 贝叶斯块                                                    */
使用贝叶斯块算法对 $\{(T_i, R_i)\}$ 进行自动分段，得到分段边界 $\{T_k\}$
/* 显著性计算                                                  */
for 每个贝叶斯块 $[T_k, T_{k+1}]$ do
    计算该时间区间内的总光子数：$C_k = \sum_{T_i \in [T_k, T_{k+1}]} R_i \Delta t$
    计算该时间区间内的背景光子数：$C_{k,bc} = \sum_{T_i \in [T_k, T_{k+1}]} R_{i,bc} \Delta t$
    计算信号显著性 $S_k = \sqrt{2\log\frac{L_1(\theta_1^{\mathrm{mle}})}{L_0(\theta_0^{\mathrm{mle}})}}$
    if $S_k \geqslant S_{\mathrm{th}}$ then
        标记该贝叶斯块为信号区间 $[T_k^S, T_{k+1}^S]$
        标记该贝叶斯块内的数据为信号 $\{(T_i^S, R_i^S)\}_{T_i \in [T_k, T_{k+1}]}$
    end
    else
        标记该贝叶斯块为背景区间 $[T_k^B, T_{k+1}^B]$
        标记该贝叶斯块内的数据为背景 $\{(T_i^B, R_i^B)\}_{T_i \in [T_k, T_{k+1}]}$
    end
end
/* 多项式拟合                                                  */
构建 $m$ 阶多项式 $f(t, \beta) = \sum_{j=0}^{m} \beta_j t^j$ 拟合背景数据 $\{(T_i^B, R_i^B)\}$
采用加权最小二乘法求解最优多项式系数 $\hat{\beta}$ 及其协方差矩阵 $\mathrm{Cov}(\hat{\beta})$
生成背景光变曲线 $B(T) = \{(T_i, f(T_i, \hat{\beta}))\}$
生成信号光变（净光变）曲线 $S(T) = A(T) - B(T)$
return $S(T), B(T)$
```



Gamma-ray data analysis (e.g. Fermi/GBM) — duration

Algorithm 3: 持续时间 T_{90} 的计算算法

```
Input: 总光变曲线 $A(T) = \{(T_i, R_i)\}$，蒙特卡洛模拟次数 $N$
Output: $T_{90}$ 及其误差范围
/* 自动化信号识别与背景拟合                                          */
提取背景光变曲线 $B(T) = \{(T_i, R_i^B)\}$ 和净光变曲线 $S(T) = \{(T_i, R_i^S)\}$
确定信号区间 $\{T_i^S\}$ 和背景区间 $\{T_i^B\}$
/* 计算 $T_{90}$                                                       */
计算净光子累积计数曲线：$F(T) = \sum_{T_i \leqslant T} S(T_i)\Delta t$
计算 0% 流量水平：$F_0 = \langle F(T_i) \rangle_{T_i \in \{T_i^B\},\, T_i < t_{\mathrm{start}}}$
计算 100% 流量水平：$F_{100} = \langle F(T_i) \rangle_{T_i \in \{T_i^B\},\, T_i > t_{\mathrm{end}}}$
计算 n% 流量水平：$F_n = n\%(F_{100} - F_0) + F_0$
确定 $t_5$ 和 $t_{95}$ 使得：$F(t_5) = F_5$，$F(t_{95}) = F_{95}$
计算 $T_{90}$：$T_{90} = t_{95} - t_5$
/* 蒙特卡洛模拟                                                        */
for $j = 1$ 到 $N$ do
    使用泊松随机抽样生成模拟总光变曲线：$A^{\mathrm{sim}}(T_i) \sim \pi(A(T_i))$
    使用高斯随机抽样生成模拟背景光变曲线：
     $B^{\mathrm{sim}}(T_i) \sim \mathcal{N}(B(T_i), \sigma_B^2(T_i))$
    计算模拟净光变曲线：$S^{\mathrm{sim}}(T_i) = A^{\mathrm{sim}}(T_i) - B^{\mathrm{sim}}(T_i)$
    计算模拟净光变曲线的 $T_{90}^{\mathrm{sim}}$
end
计算 $T_{90}^{\mathrm{sim}}$ 分布的 $1\sigma$ 置信区间 $[T_{90,\mathrm{low}}, T_{90,\mathrm{high}}]$
return $T_{90}$, $[T_{90,\mathrm{low}}, T_{90,\mathrm{high}}]$
```



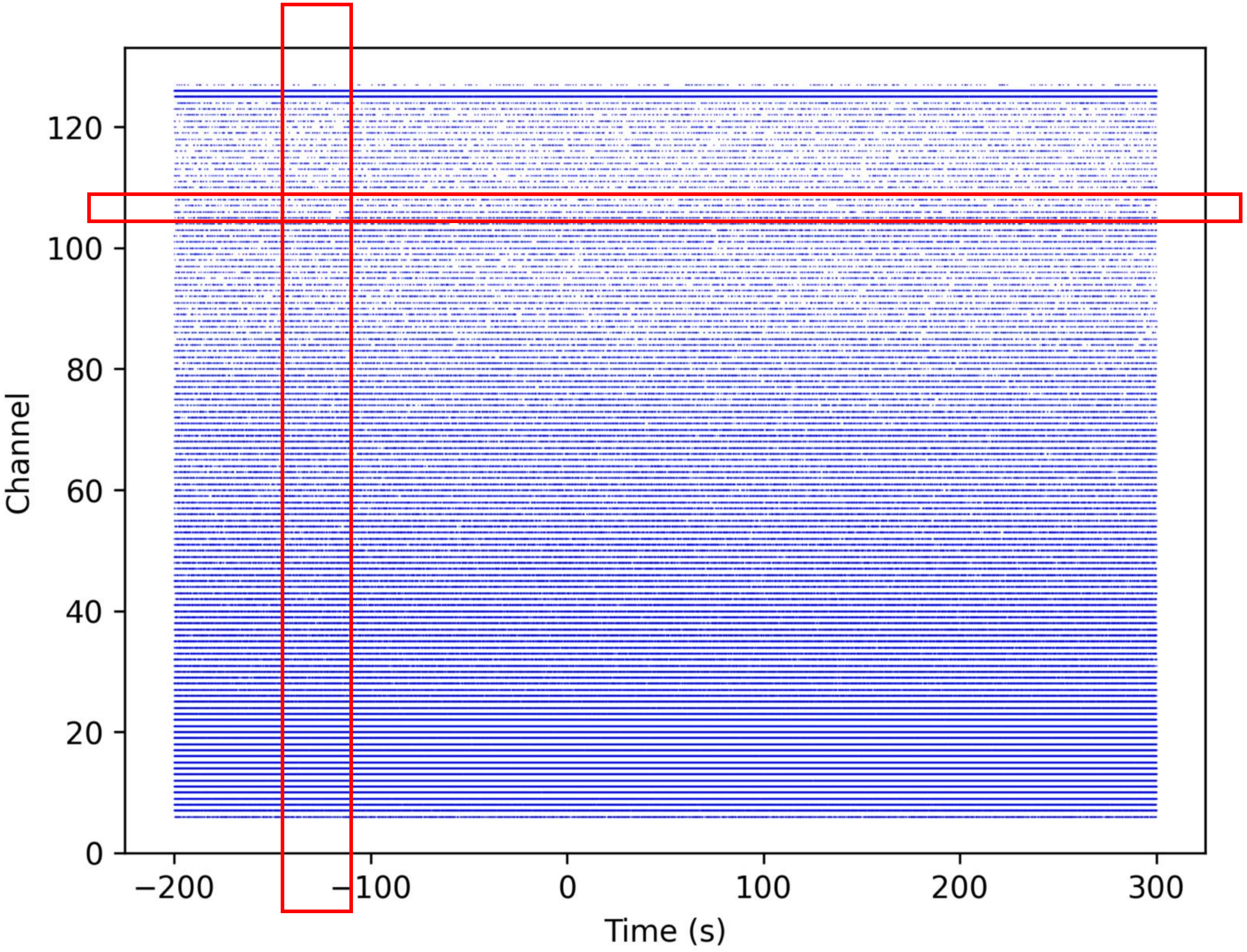
100% level

95% level

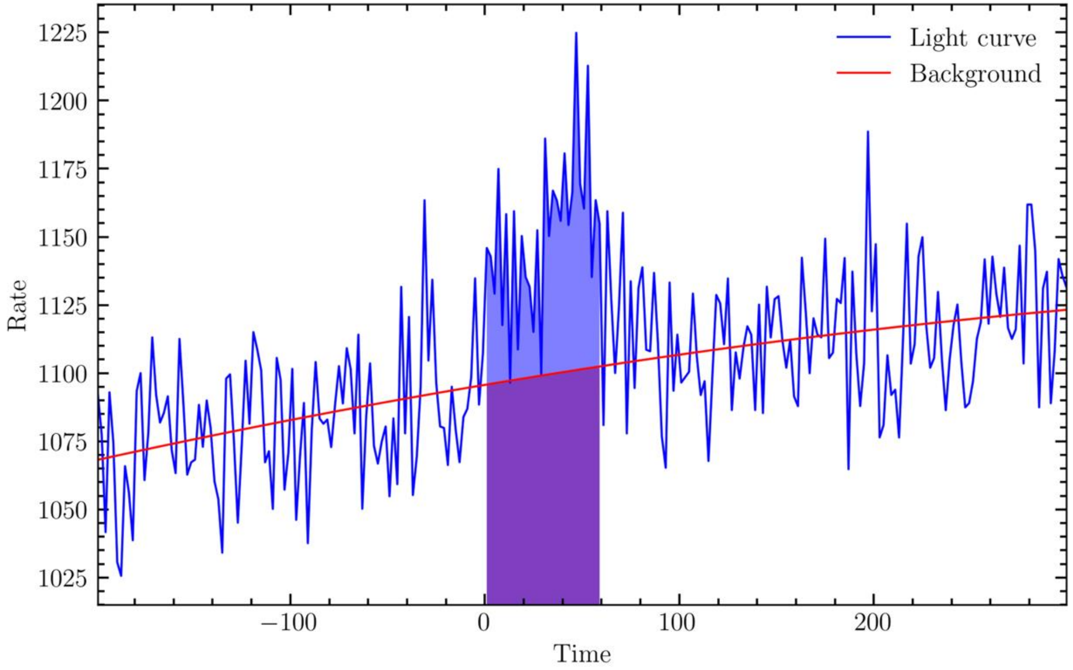
5% level

0% level

Gamma-ray data analysis (e.g. Fermi/GBM) — spectrum

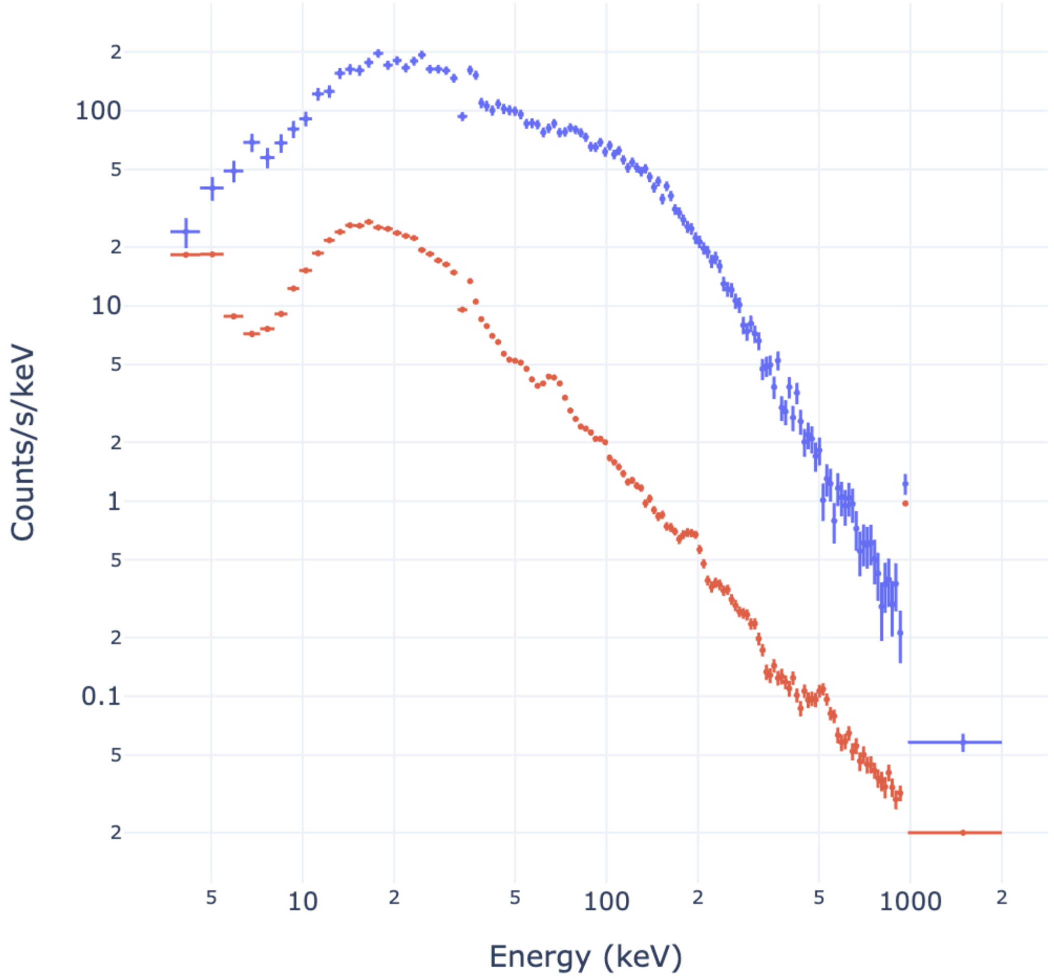


total and background spectrum



Gamma-ray data analysis (e.g. Fermi/GBM) —spectrum

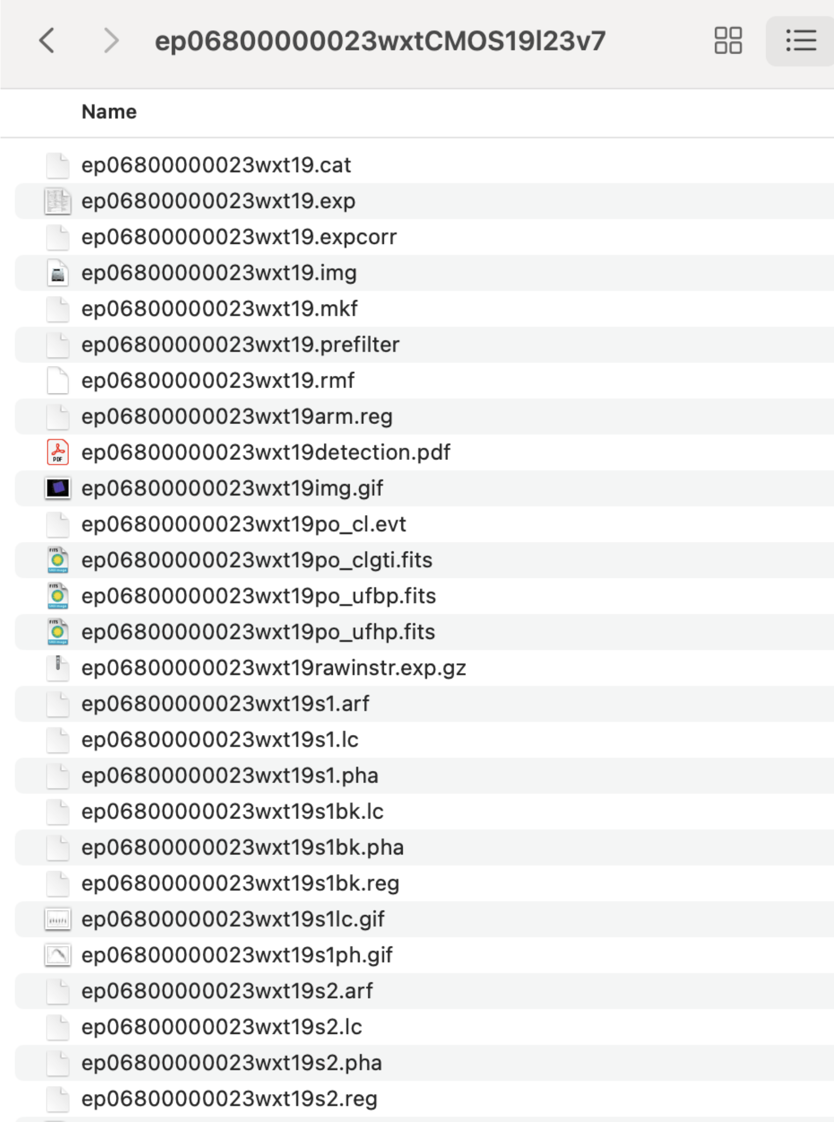
```
Algorithm 4: 伽马射线背景能谱提取算法
Input: 光子事件列表 ${(t_i, p_i)}$，时间切片 $[T_l, T_r]$
Output: 背景能谱 ${(I, C_I^B, \sigma_{C_I^B})}$
提取总光变曲线 $A(T) = \{(T_i, R_i)\}$
应用信号识别与背景拟合算法，确定信号区间 $\{T_i^S\}$ 和背景区间 $\{T_i^B\}$
for 每一个能道 $I \in \{1, 2, ..., N\}$ do
    提取光子事件列表 $\{(t_i, p_i) \mid p_i = I\}$
    提取光变曲线 $A_I(T) = \{(T_{i,I}, R_{i,I})\}$
    使用多项式拟合背景区间 $\{T_i^B\}$ 内的 $A_I(T)$
    获取背景光变曲线的多项式模型 $B_I(t)$ 及其误差 $\sigma_{B_I}(t)$
    计算背景光子数：$C_I^B = \int_{T_l}^{T_r} B_I(t)dt$
    计算背景光子数的统计误差：$\sigma_{C_I^B} = \int_{T_l}^{T_r} \sigma_{B_I}(t)dt$
end
return $\{(I, C_I^B, \sigma_{C_I^B})\}$
```



Part 2: X-ray data analysis

X-ray data (e.g. EP-WXT)

X-ray: EP-WXT

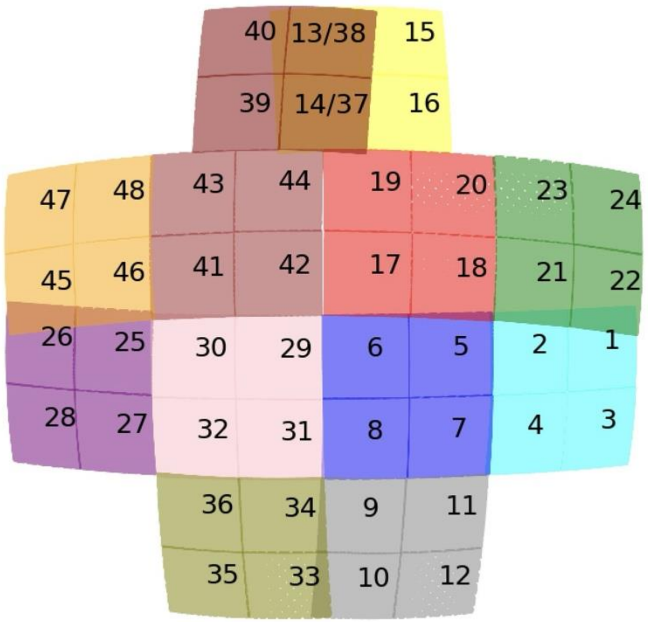


ep06800000023wxtCMOS19l23v7

Name

- ep06800000023wxt19.cat
- ep06800000023wxt19.exp
- ep06800000023wxt19.expcorr
- ep06800000023wxt19.img
- ep06800000023wxt19.mkf
- ep06800000023wxt19.prefilter
- ep06800000023wxt19.rmf
- ep06800000023wxt19arm.reg
- ep06800000023wxt19detection.pdf
- ep06800000023wxt19img.gif
- ep06800000023wxt19po_cl.evt
- ep06800000023wxt19po_clgti.fits
- ep06800000023wxt19po_ufbp.fits
- ep06800000023wxt19po_ufhp.fits
- ep06800000023wxt19rawinstr.exp.gz
- ep06800000023wxt19s1.arf
- ep06800000023wxt19s1.lc
- ep06800000023wxt19s1.pha
- ep06800000023wxt19s1bk.lc
- ep06800000023wxt19s1bk.pha
- ep06800000023wxt19s1bk.reg
- ep06800000023wxt19s1lc.gif
- ep06800000023wxt19s1ph.gif
- ep06800000023wxt19s2.arf
- ep06800000023wxt19s2.lc
- ep06800000023wxt19s2.pha
- ep06800000023wxt19s2.reg

ep[obsID]wxt[number][pp]_cl.evt	event file
ep[obsID]wxt[number].mkf	make filter file
ep[obsID]wxt[number].img	image
ep[obsID]wxt[number].exp	exposure map
ep[obsID]wxt[number].expcorr	corrected exposure map
ep[obsID]wxt[number].cat	source catalog
ep[obsID]wxt[number]arm.reg	arm region file
ep[obsID]wxt[number]s[sourceID].lc	source light curve
ep[obsID]wxt[number]s[sourceID]bk.lc	background light curve
ep[obsID]wxt[number]s[sourceID].pha	source spectrum
ep[obsID]wxt[number]s[sourceID]bk.pha	background spectrum
ep[obsID]wxt[number]s[sourceID].arf	ancillary response file
ep[obsID]wxt[number].rmf	response matrix



X-ray data (e.g. EP-WXT)

X-ray: EP-WXT clean event data

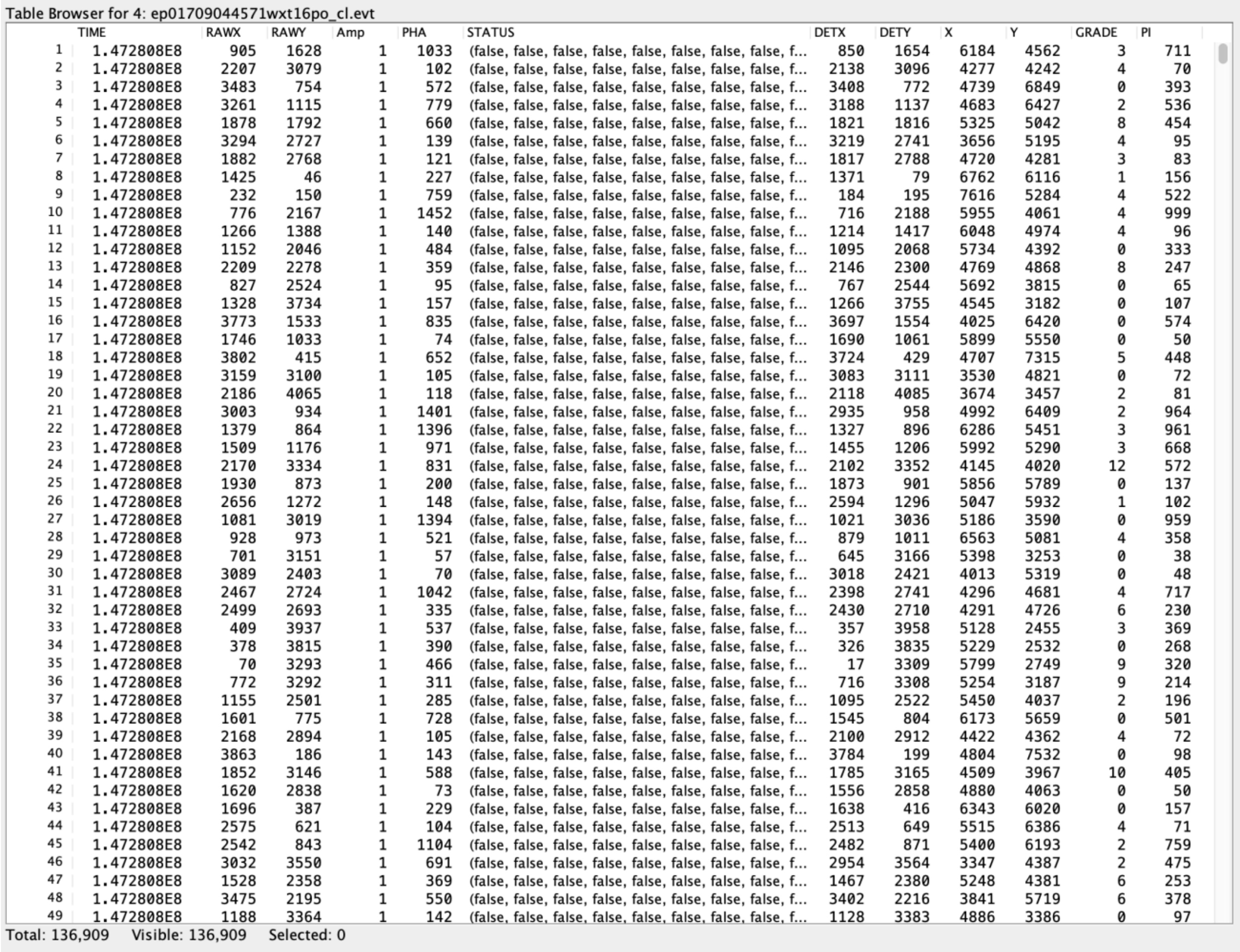


Table Browser for 4: ep01709044571wxt16po_cl.evt

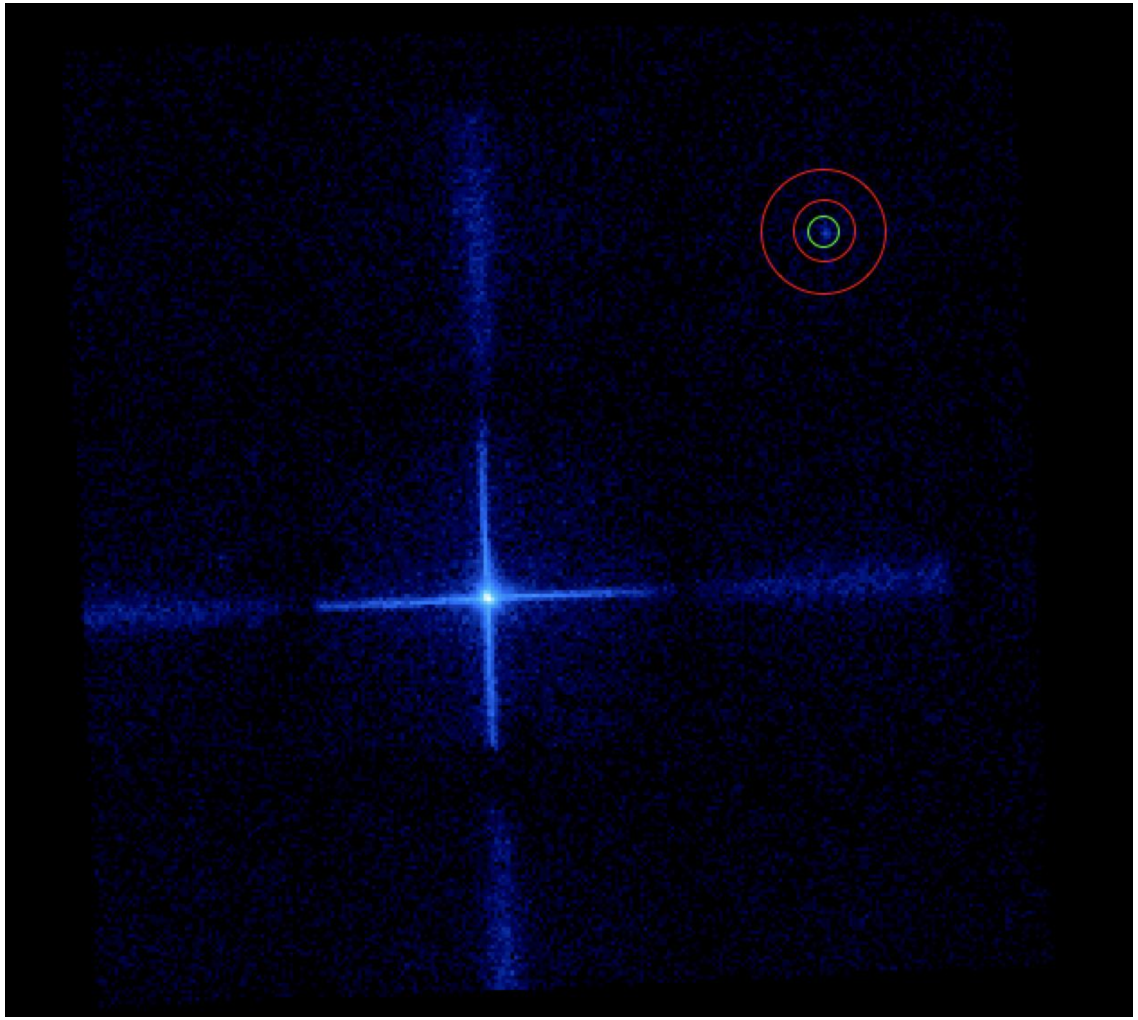
	TIME	RAWX	RAWY	Amp	PHA	STATUS	DETX	DETY	X	Y	GRADE	PI
1	1.472808E8	905	1628	1	1033	(false, false, false, false, false, false, false, false, f...	850	1654	6184	4562	3	711
2	1.472808E8	2207	3079	1	102	(false, false, false, false, false, false, false, false, f...	2138	3096	4277	4242	4	70
3	1.472808E8	3483	754	1	572	(false, false, false, false, false, false, false, false, f...	3408	772	4739	6849	0	393
4	1.472808E8	3261	1115	1	779	(false, false, false, false, false, false, false, false, f...	3188	1137	4683	6427	2	536
5	1.472808E8	1878	1792	1	660	(false, false, false, false, false, false, false, false, f...	1821	1816	5325	5042	8	454
6	1.472808E8	3294	2727	1	139	(false, false, false, false, false, false, false, false, f...	3219	2741	3656	5195	4	95
7	1.472808E8	1882	2768	1	121	(false, false, false, false, false, false, false, false, f...	1817	2788	4720	4281	3	83
8	1.472808E8	1425	46	1	227	(false, false, false, false, false, false, false, false, f...	1371	79	6762	6116	1	156
9	1.472808E8	232	150	1	759	(false, false, false, false, false, false, false, false, f...	184	195	7616	5284	4	522
10	1.472808E8	776	2167	1	1452	(false, false, false, false, false, false, false, false, f...	716	2188	5955	4061	4	999
11	1.472808E8	1266	1388	1	140	(false, false, false, false, false, false, false, false, f...	1214	1417	6048	4974	4	96
12	1.472808E8	1152	2046	1	484	(false, false, false, false, false, false, false, false, f...	1095	2068	5734	4392	0	333
13	1.472808E8	2209	2278	1	359	(false, false, false, false, false, false, false, false, f...	2146	2300	4769	4868	8	247
14	1.472808E8	827	2524	1	95	(false, false, false, false, false, false, false, false, f...	767	2544	5692	3815	0	65
15	1.472808E8	1328	3734	1	157	(false, false, false, false, false, false, false, false, f...	1266	3755	4545	3182	0	107
16	1.472808E8	3773	1533	1	835	(false, false, false, false, false, false, false, false, f...	3697	1554	4025	6420	0	574
17	1.472808E8	1746	1033	1	74	(false, false, false, false, false, false, false, false, f...	1690	1061	5899	5550	0	50
18	1.472808E8	3802	415	1	652	(false, false, false, false, false, false, false, false, f...	3724	429	4707	7315	5	448
19	1.472808E8	3159	3100	1	105	(false, false, false, false, false, false, false, false, f...	3083	3111	3530	4821	0	72
20	1.472808E8	2186	4065	1	118	(false, false, false, false, false, false, false, false, f...	2118	4085	3674	3457	2	81
21	1.472808E8	3003	934	1	1401	(false, false, false, false, false, false, false, false, f...	2935	958	4992	6409	2	964
22	1.472808E8	1379	864	1	1396	(false, false, false, false, false, false, false, false, f...	1327	896	6286	5451	3	961
23	1.472808E8	1509	1176	1	971	(false, false, false, false, false, false, false, false, f...	1455	1206	5992	5290	3	668
24	1.472808E8	2170	3334	1	831	(false, false, false, false, false, false, false, false, f...	2102	3352	4145	4020	12	572
25	1.472808E8	1930	873	1	200	(false, false, false, false, false, false, false, false, f...	1873	901	5856	5789	0	137
26	1.472808E8	2656	1272	1	148	(false, false, false, false, false, false, false, false, f...	2594	1296	5047	5932	1	102
27	1.472808E8	1081	3019	1	1394	(false, false, false, false, false, false, false, false, f...	1021	3036	5186	3590	0	959
28	1.472808E8	928	973	1	521	(false, false, false, false, false, false, false, false, f...	879	1011	6563	5081	4	358
29	1.472808E8	701	3151	1	57	(false, false, false, false, false, false, false, false, f...	645	3166	5398	3253	0	38
30	1.472808E8	3089	2403	1	70	(false, false, false, false, false, false, false, false, f...	3018	2421	4013	5319	0	48
31	1.472808E8	2467	2724	1	1042	(false, false, false, false, false, false, false, false, f...	2398	2741	4296	4681	4	717
32	1.472808E8	2499	2693	1	335	(false, false, false, false, false, false, false, false, f...	2430	2710	4291	4726	6	230
33	1.472808E8	409	3937	1	537	(false, false, false, false, false, false, false, false, f...	357	3958	5128	2455	3	369
34	1.472808E8	378	3815	1	390	(false, false, false, false, false, false, false, false, f...	326	3835	5229	2532	0	268
35	1.472808E8	70	3293	1	466	(false, false, false, false, false, false, false, false, f...	17	3309	5799	2749	9	320
36	1.472808E8	772	3292	1	311	(false, false, false, false, false, false, false, false, f...	716	3308	5254	3187	9	214
37	1.472808E8	1155	2501	1	285	(false, false, false, false, false, false, false, false, f...	1095	2522	5450	4037	2	196
38	1.472808E8	1601	775	1	728	(false, false, false, false, false, false, false, false, f...	1545	804	6173	5659	0	501
39	1.472808E8	2168	2894	1	105	(false, false, false, false, false, false, false, false, f...	2100	2912	4422	4362	4	72
40	1.472808E8	3863	186	1	143	(false, false, false, false, false, false, false, false, f...	3784	199	4804	7532	0	98
41	1.472808E8	1852	3146	1	588	(false, false, false, false, false, false, false, false, f...	1785	3165	4509	3967	10	405
42	1.472808E8	1620	2838	1	73	(false, false, false, false, false, false, false, false, f...	1556	2858	4880	4063	0	50
43	1.472808E8	1696	387	1	229	(false, false, false, false, false, false, false, false, f...	1638	416	6343	6020	0	157
44	1.472808E8	2575	621	1	104	(false, false, false, false, false, false, false, false, f...	2513	649	5515	6386	4	71
45	1.472808E8	2542	843	1	1104	(false, false, false, false, false, false, false, false, f...	2482	871	5400	6193	2	759
46	1.472808E8	3032	3550	1	691	(false, false, false, false, false, false, false, false, f...	2954	3564	3347	4387	2	475
47	1.472808E8	1528	2358	1	369	(false, false, false, false, false, false, false, false, f...	1467	2380	5248	4381	6	253
48	1.472808E8	3475	2195	1	550	(false, false, false, false, false, false, false, false, f...	3402	2216	3841	5719	6	378
49	1.472808E8	1188	3364	1	142	(false, false, false, false, false, false, false, false, f...	1128	3383	4886	3386	0	97

Total: 136,909 Visible: 136,909 Selected: 0

参数	类型	单位	解释
TIME	8 字节实数	s	光子事件的到达时间
X	2 字节整数	pixel	光子在天球坐标系中的 X 坐标
Y	2 字节整数	pixel	光子在天球坐标系中的 Y 坐标
RAWX	2 字节整数	pixel	光子在原始探测器坐标系中的 X 坐标（未线性化前）
RAWY	2 字节整数	pixel	光子在原始探测器坐标系中的 Y 坐标（未线性化前）
DETX	2 字节整数	pixel	光子在线性化探测器坐标系中的 X 坐标
DETY	2 字节整数	pixel	光子在线性化探测器坐标系中的 Y 坐标
PHA	4 字节整数	-	脉冲高度分析器通道编号
PI	4 字节整数	-	脉冲不变通道编号
GRADE	2 字节整数	-	光子事件的等级
STATUS	16 位无类型	-	光子事件的数据质量标志

X-ray data (e.g. EP-WXT) — filter region

Use ds9 to set source and background regions



Source region:
circle (radius: 67 pix)

Background region:
annulus (radius: 134-268 pix)

X-ray data (e.g. EP-WXT) — extract event

Use xselcet software of heasoft

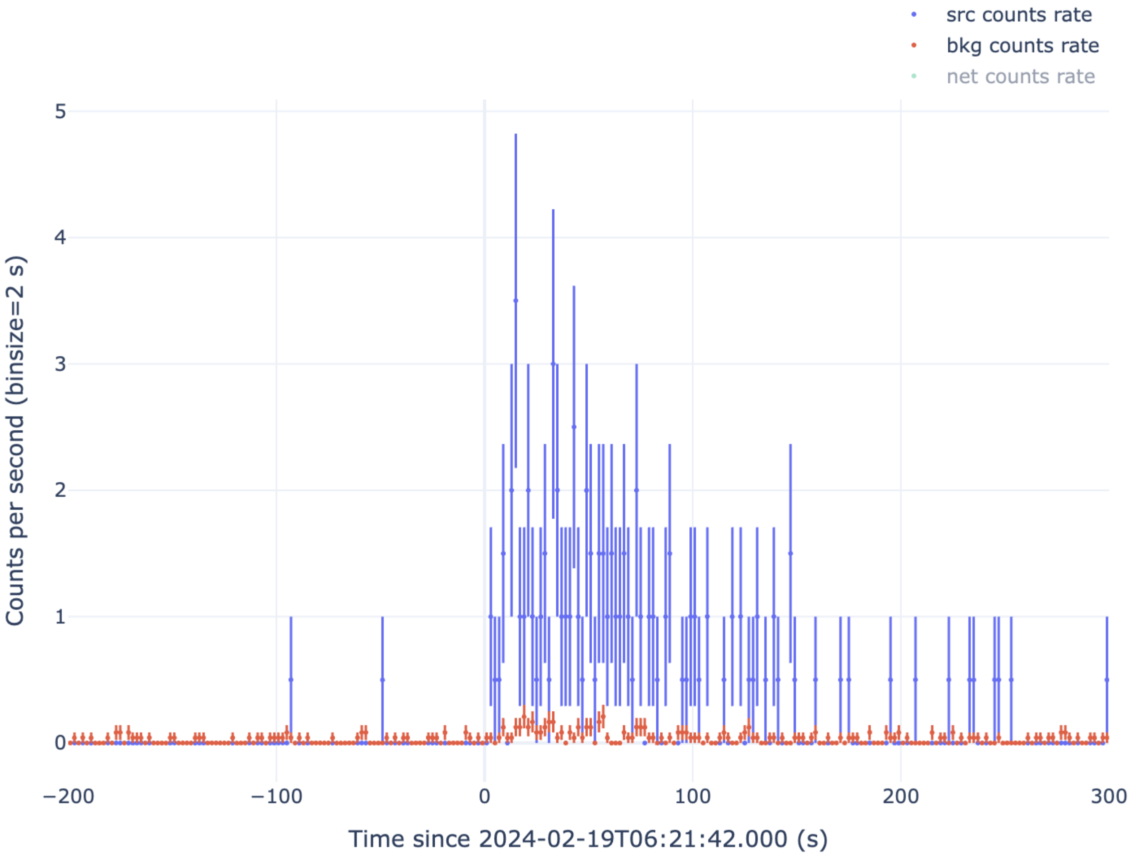
```
> Enter session name > xsel
xsel:SUZAKU > read events
> Enter the Event file dir > ./
> Enter Event file list > ep13600003859wxt44po_cl.evt
Got new mission: EP
> Reset the mission ? > yes
xsel:EP-WXT-PHOTON > filter region ep13600003859wxt44s2.reg
xsel:EP-WXT-PHOTON > extract events
xsel:EP-WXT-PHOTON > save events src.evt
> Use filtered events as input data file ? > no
xsel:EP-WXT-PHOTON > clear events
xsel:EP-WXT-PHOTON > clear region
xsel:EP-WXT-PHOTON > filter region ep13600003859wxt44s2bk.reg
xsel:EP-WXT-PHOTON > extract events
xsel:EP-WXT-PHOTON > save events bkg.evt
> Use filtered events as input data file ? > no
xsel:EP-WXT-PHOTON > clear events
xsel:EP-WXT-PHOTON > clear region
```

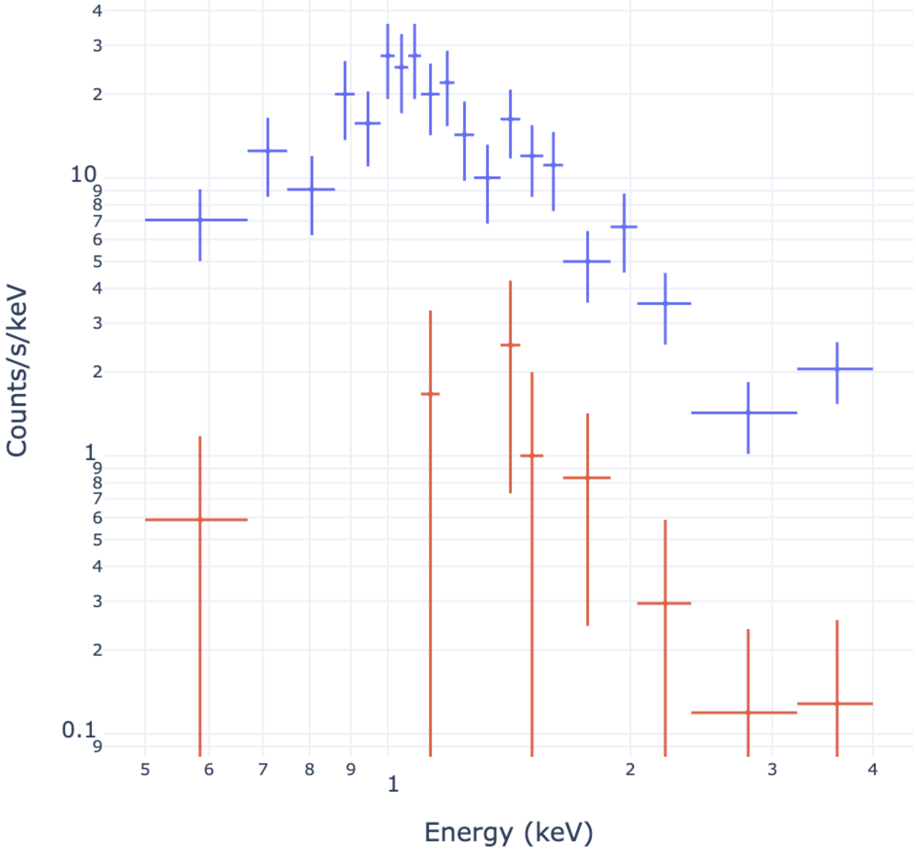
X-ray data (e.g. EP-WXT) — extract curve and spectrum

Use xselcet software of heasoft

```
> Enter session name > xsel
xsel:SUZAKU > read events
> Enter the Event file dir > ./
> Enter Event file list > ep13600003859wxt44po_cl.evt
Got new mission: EP
> Reset the mission ? > yes
xsel:EP-WXT-PHOTON > filter time scc
> Enter list of start and stop times > 130486902.0, 130487062.0
> Enter list of start and stop times > x
xsel:EP-WXT-PHOTON > filter region ep13600003859wxt44s2.reg
xsel:EP-WXT-PHOTON > extract curve
xsel:EP-WXT-PHOTON > save curve src.lc
xsel:EP-WXT-PHOTON > extract spectrum
xsel:EP-WXT-PHOTON > save spectrum src.pi
xsel:EP-WXT-PHOTON > clear region
xsel:EP-WXT-PHOTON > filter region ep13600003859wxt44s2bk.reg
xsel:EP-WXT-PHOTON > extract curve
xsel:EP-WXT-PHOTON > save curve bkg.lc
xsel:EP-WXT-PHOTON > extract spectrum
xsel:EP-WXT-PHOTON > save spectrum bkg.pi
xsel:EP-WXT-PHOTON > clear region
```

X-ray data (e.g. EP-WXT) — extract curve and spectrum





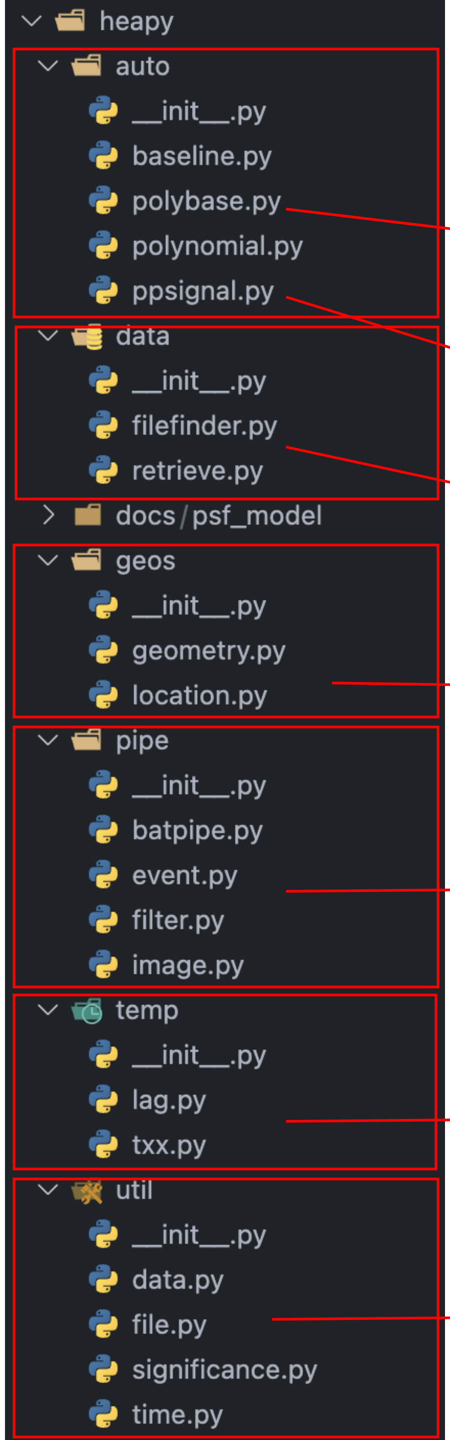
Part 3: heapy

https://github.com/jyangch/heapy

A unified toolkit for timing and spectral analysis of X-ray and gamma-ray transient data

Other options: heasoft, gbm data tools, gamma-ray data tools, threeml

heapy package structure



```
heapy
  auto
    __init__.py
    baseline.py
    polybase.py
    polynomial.py
    ppsignal.py
  data
    __init__.py
    filefinder.py
    retrieve.py
  docs / psf_model
  geos
    __init__.py
    geometry.py
    location.py
  pipe
    __init__.py
    batpipe.py
    event.py
    filter.py
    image.py
  temp
    __init__.py
    lag.py
    txx.py
  util
    __init__.py
    data.py
    file.py
    significance.py
    time.py
```

polybase.py: for the case of Gaussian background (e.g. Fermi/GBM)
bayesian block, SNR calculation, signal identification, background modelling

ppsignal.py: for the case of Poisson background (e.g. EP-WXT)
bayesian block, SNR calculation, signal identification


Retrieve in local → Retrieve in ftp server → Download → Event / Image

yes

geometry.py: spacecraft location in orbit and detector pointing information (only GBM)
location.py: localizing the real source (only GBM, not public)

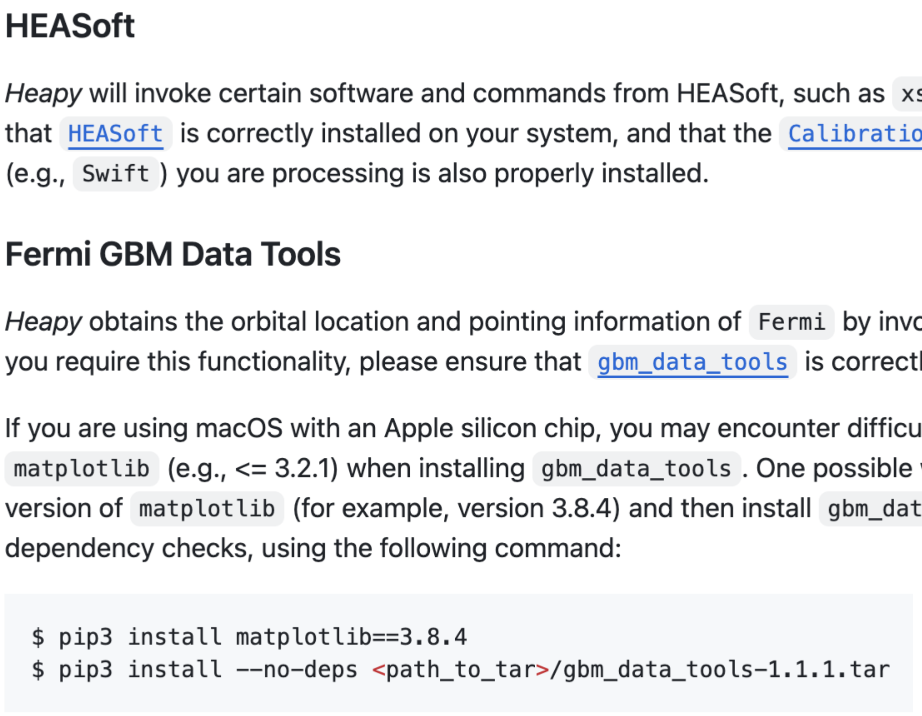
Extract light curve, spectrum, and response
batpipe.py: Swift/BAT pipeline implemented through calling heasoft commands
event.py: analysis pipeline for event list data (time+energy, e.g. Fermi/GBM, GECAM)
image.py: analysis pipeline for image data (time+energy+space, e.g. EP-WXT, Swift/XRT)

Temporal analysis for GRB X-ray and gamma-ray data
duration calculation, lag calculation

Some useful functions
time conversion, significance calculation

Spectral fit <- BaySpec

heapy installation: pip3 install heapyx



HEASoft

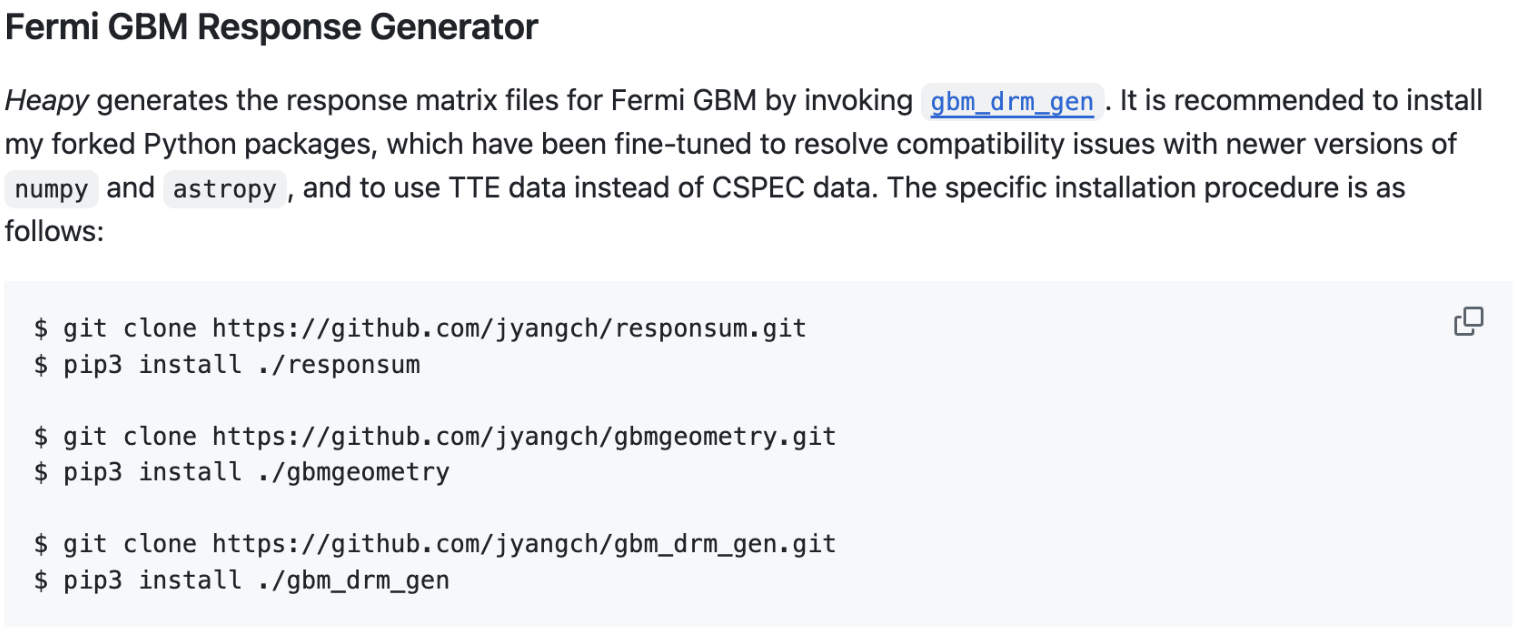
Heapy will invoke certain software and commands from HEASoft, such as `xselect` and `ximage`. Please ensure that `HEASoft` is correctly installed on your system, and that the `Calibration Database` (CALDB) for the mission (e.g., `Swift`) you are processing is also properly installed.

Fermi GBM Data Tools

Heapy obtains the orbital location and pointing information of `Fermi` by invoking `gbm_data_tools`. Therefore, if you require this functionality, please ensure that `gbm_data_tools` is correctly installed in advance.

If you are using macOS with an Apple silicon chip, you may encounter difficulties compiling older versions of `matplotlib` (e.g., <= 3.2.1) when installing `gbm_data_tools`. One possible workaround is to first install a newer version of `matplotlib` (for example, version 3.8.4) and then install `gbm_data_tools` without performing dependency checks, using the following command:

```
$ pip3 install matplotlib==3.8.4
$ pip3 install --no-deps <path_to_tar>/gbm_data_tools-1.1.1.tar
```

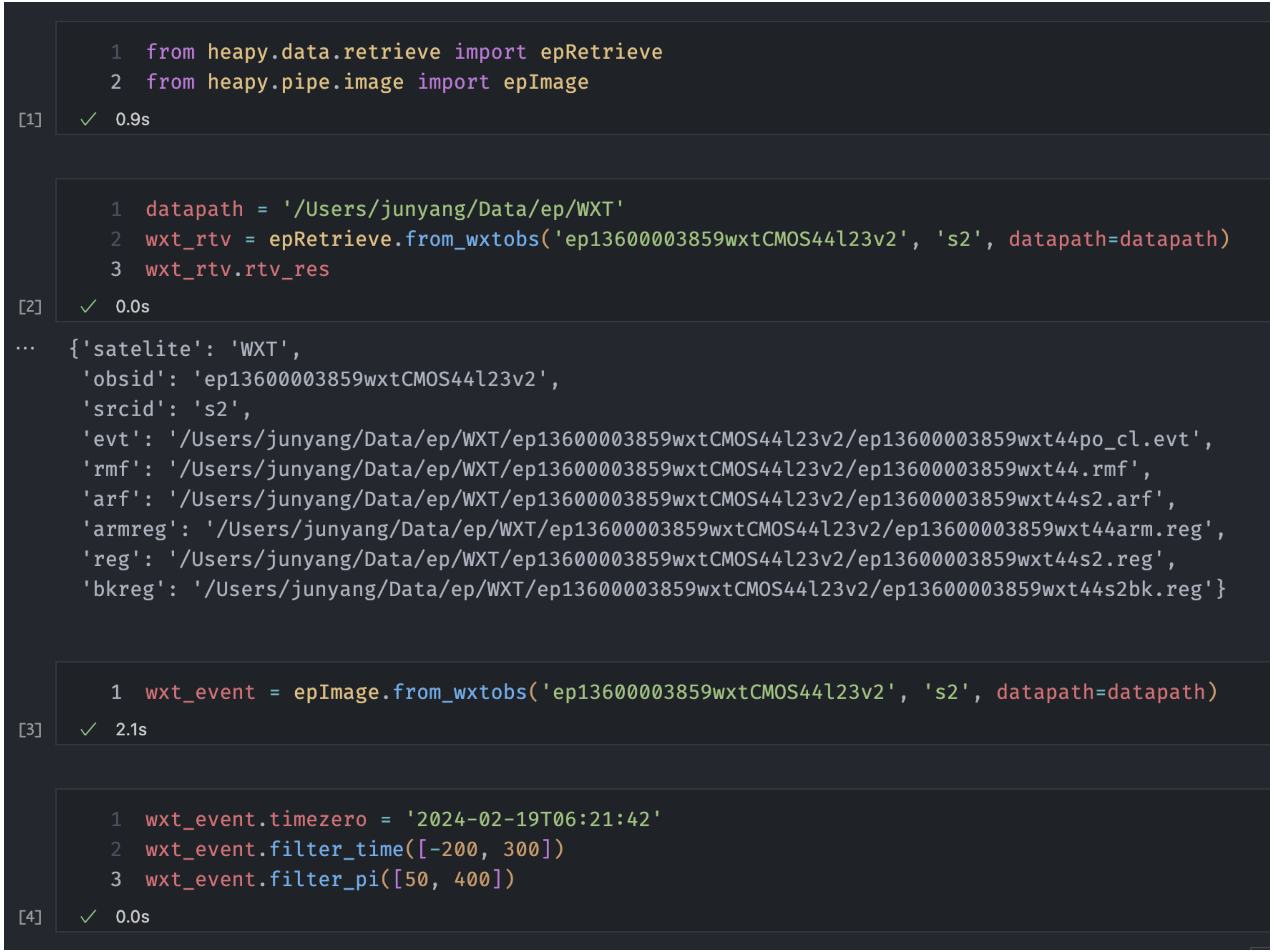


Fermi GBM Response Generator

Heapy generates the response matrix files for Fermi GBM by invoking `gbm_drm_gen`. It is recommended to install my forked Python packages, which have been fine-tuned to resolve compatibility issues with newer versions of `numpy` and `astropy`, and to use TTE data instead of CSPEC data. The specific installation procedure is as follows:

```
$ git clone https://github.com/jyangch/responsum.git
$ pip3 install ./responsum

$ git clone https://github.com/jyangch/gbmgeometry.git
$ pip3 install ./gbmgeometry

$ git clone https://github.com/jyangch/gbm_drm_gen.git
$ pip3 install ./gbm_drm_gen
```

heapy example — EP-WXT data analysis



```
from heapy.data.retrieve import epRetrieve
from heapy.pipe.image import epImage
```

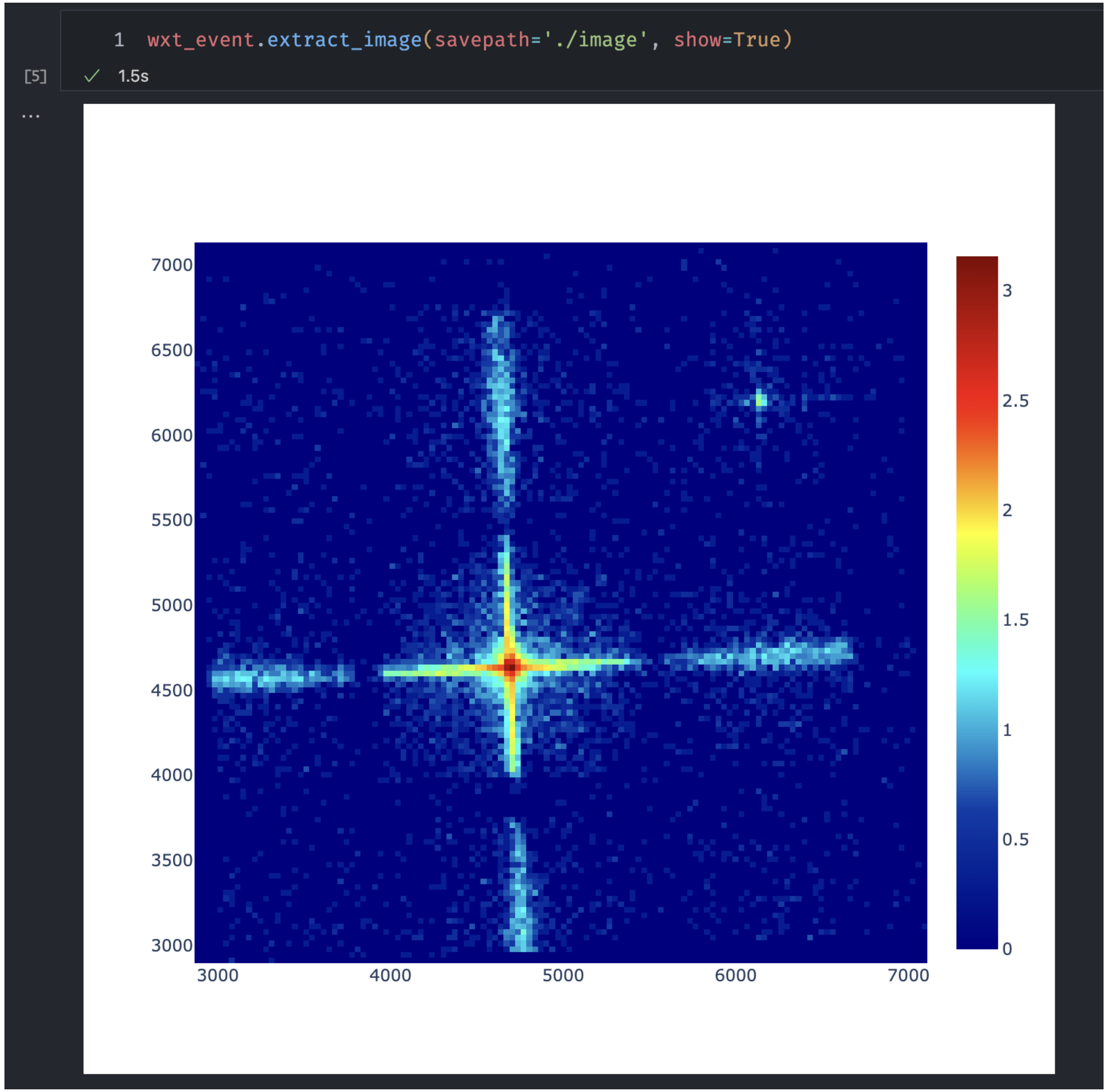
```
datapath = '/Users/junyang/Data/ep/WXT'
wxt_rtv = epRetrieve.from_wxtobs('ep13600003859wxtCMOS44l23v2', 's2', datapath=datapath)
wxt_rtv.rtv_res
```

```
{'satelite': 'WXT',
 'obsid': 'ep13600003859wxtCMOS44l23v2',
 'srcid': 's2',
 'evt': '/Users/junyang/Data/ep/WXT/ep13600003859wxtCMOS44l23v2/ep13600003859wxt44po_cl.evt',
 'rmf': '/Users/junyang/Data/ep/WXT/ep13600003859wxtCMOS44l23v2/ep13600003859wxt44.rmf',
 'arf': '/Users/junyang/Data/ep/WXT/ep13600003859wxtCMOS44l23v2/ep13600003859wxt44s2.arf',
 'armreg': '/Users/junyang/Data/ep/WXT/ep13600003859wxtCMOS44l23v2/ep13600003859wxt44arm.reg',
 'reg': '/Users/junyang/Data/ep/WXT/ep13600003859wxtCMOS44l23v2/ep13600003859wxt44s2.reg',
 'bkreg': '/Users/junyang/Data/ep/WXT/ep13600003859wxtCMOS44l23v2/ep13600003859wxt44s2bk.reg'}
```

```
wxt_event = epImage.from_wxtobs('ep13600003859wxtCMOS44l23v2', 's2', datapath=datapath)
```

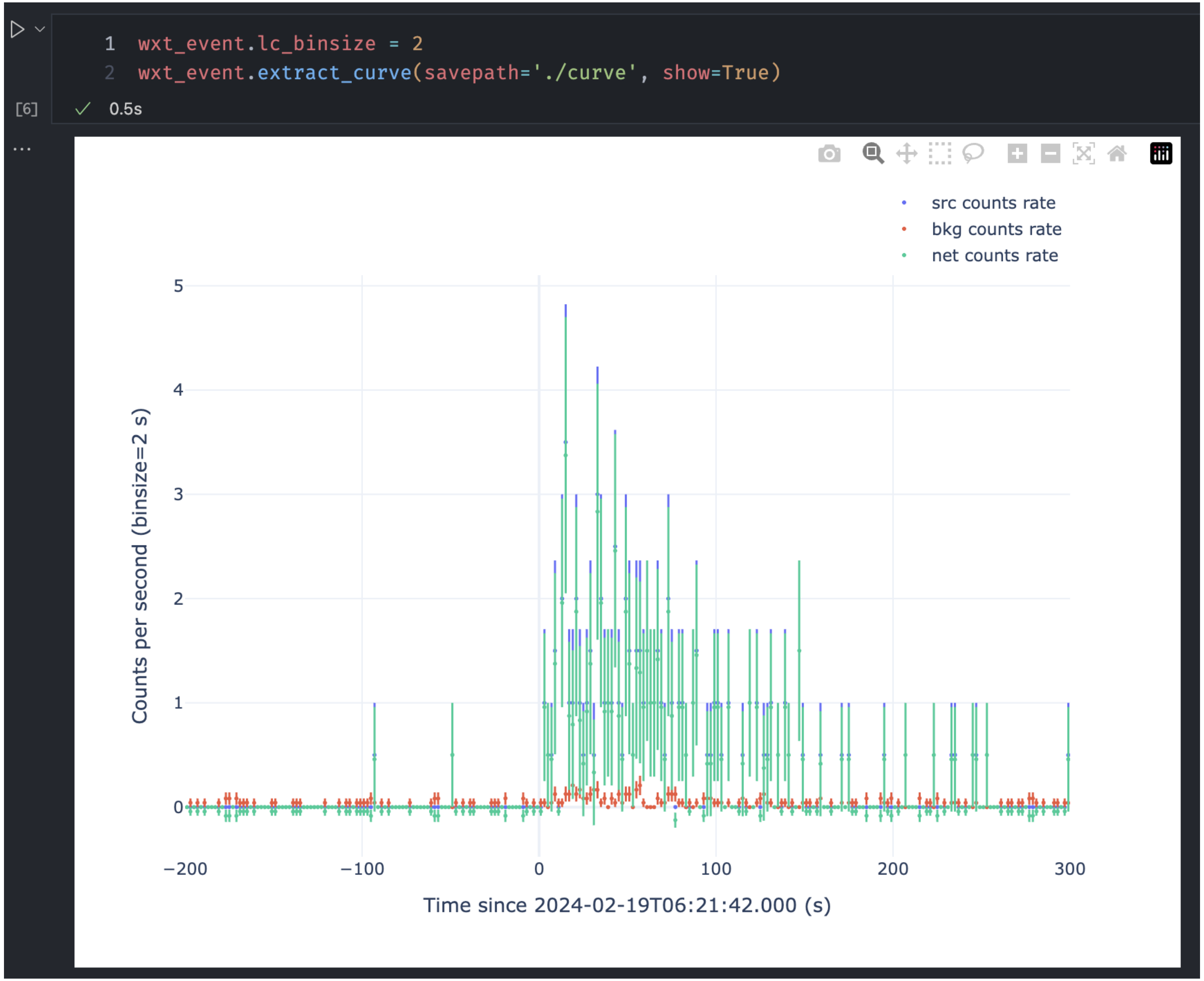
```
wxt_event.timezero = '2024-02-19T06:21:42'
wxt_event.filter_time([-200, 300])
wxt_event.filter_pi([50, 400])
```

heapy example — EP-WXT data analysis



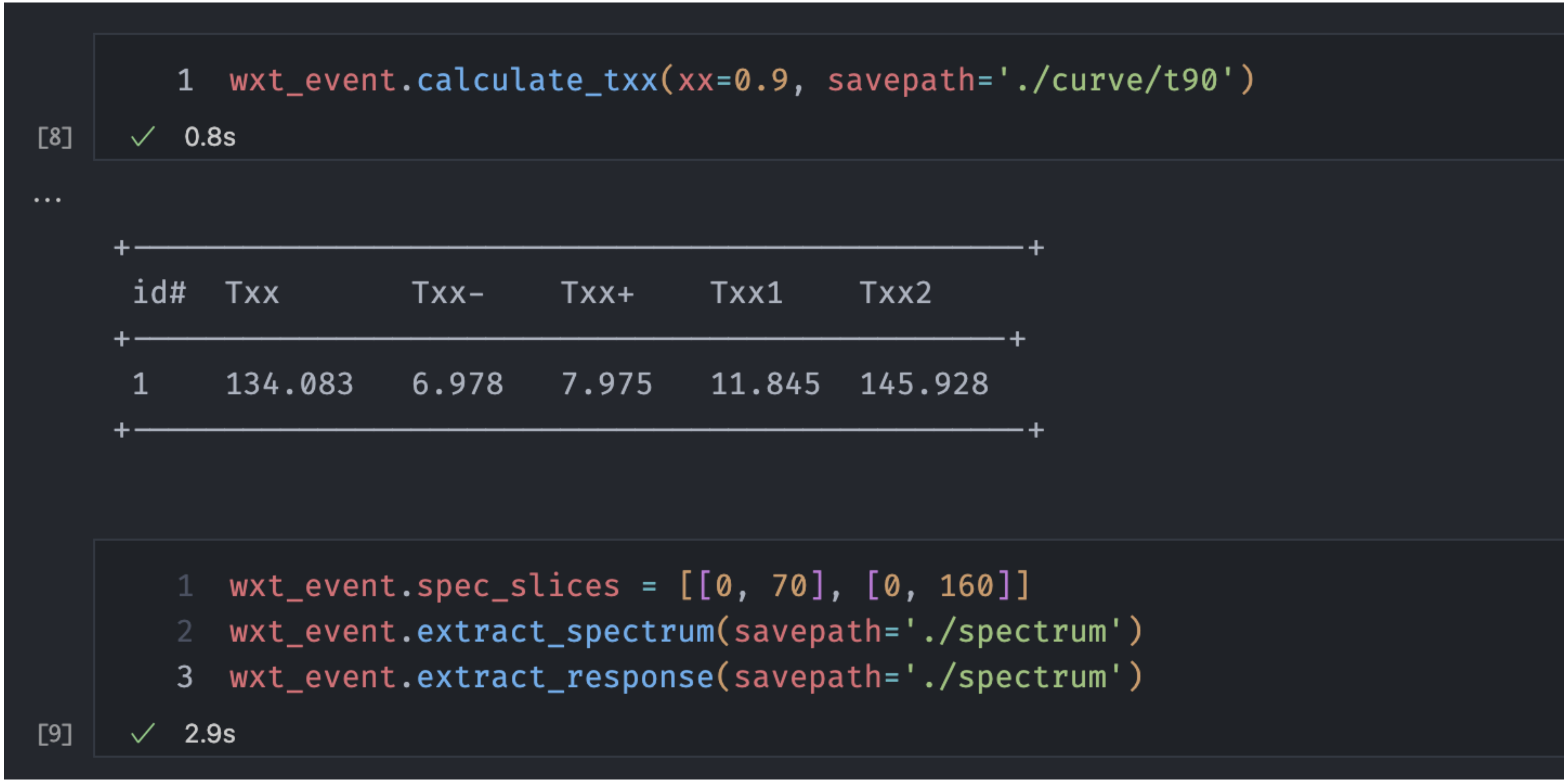
```
wxt_event.extract_image(savepath='./image', show=True)
```

heapy example — EP-WXT data analysis



```
wxt_event.lc_binsize = 2
wxt_event.extract_curve(savepath='./curve', show=True)
```

heapy example — EP-WXT data analysis

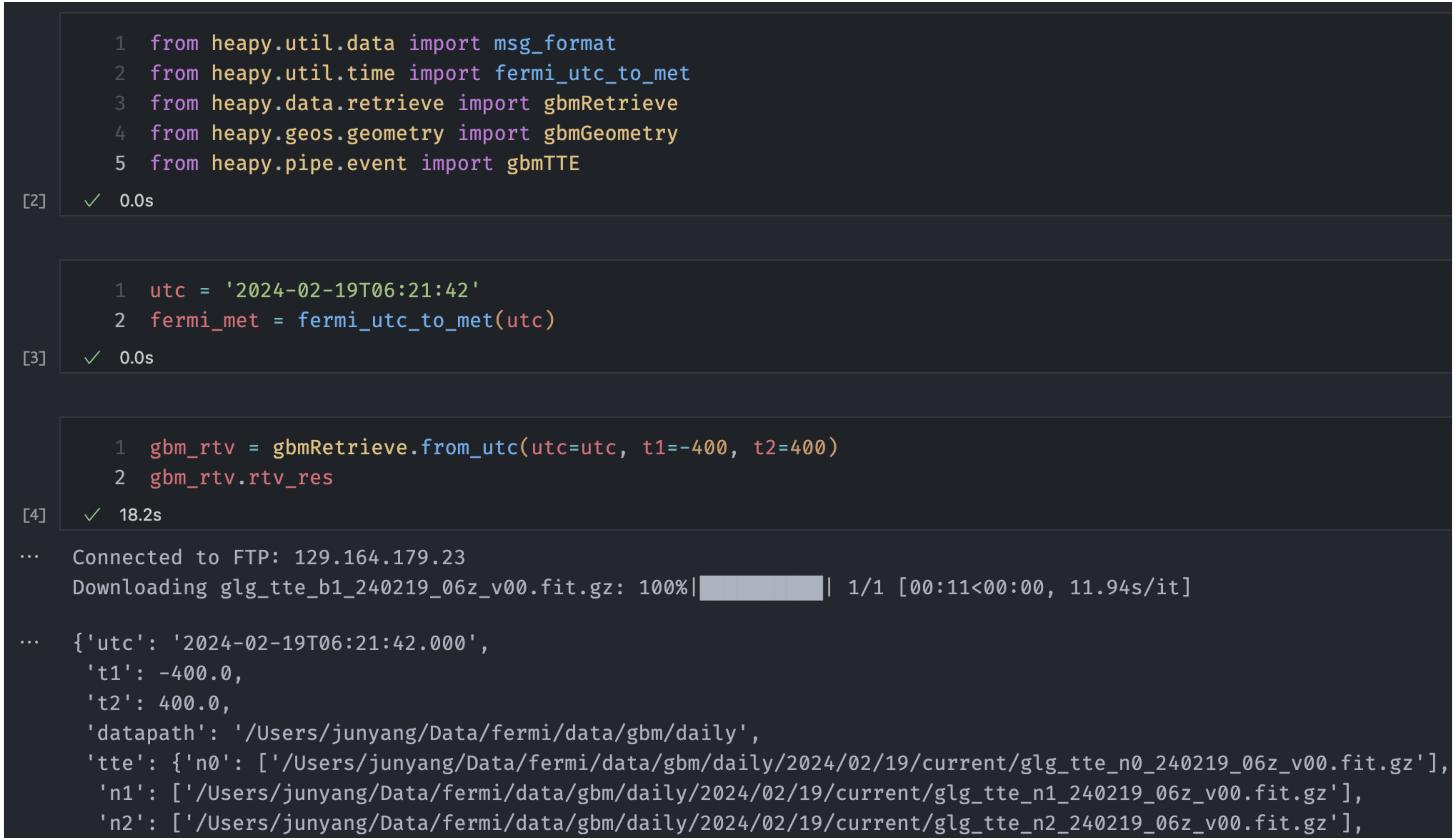


```
wxt_event.calculate_txx(xx=0.9, savepath='./curve/t90')
```

```
+──────────────────────────────────────────────+
 id#  Txx        Txx-     Txx+     Txx1     Txx2
+─────────────────────────────────────────────+
 1    134.083    6.978    7.975    11.845   145.928
+──────────────────────────────────────────────+
```

```
wxt_event.spec_slices = [[0, 70], [0, 160]]
wxt_event.extract_spectrum(savepath='./spectrum')
wxt_event.extract_response(savepath='./spectrum')
```

heapy example — Fermi/GBM data analysis



```
from heapy.util.data import msg_format
from heapy.util.time import fermi_utc_to_met
from heapy.data.retrieve import gbmRetrieve
from heapy.geos.geometry import gbmGeometry
from heapy.pipe.event import gbmTTE
```

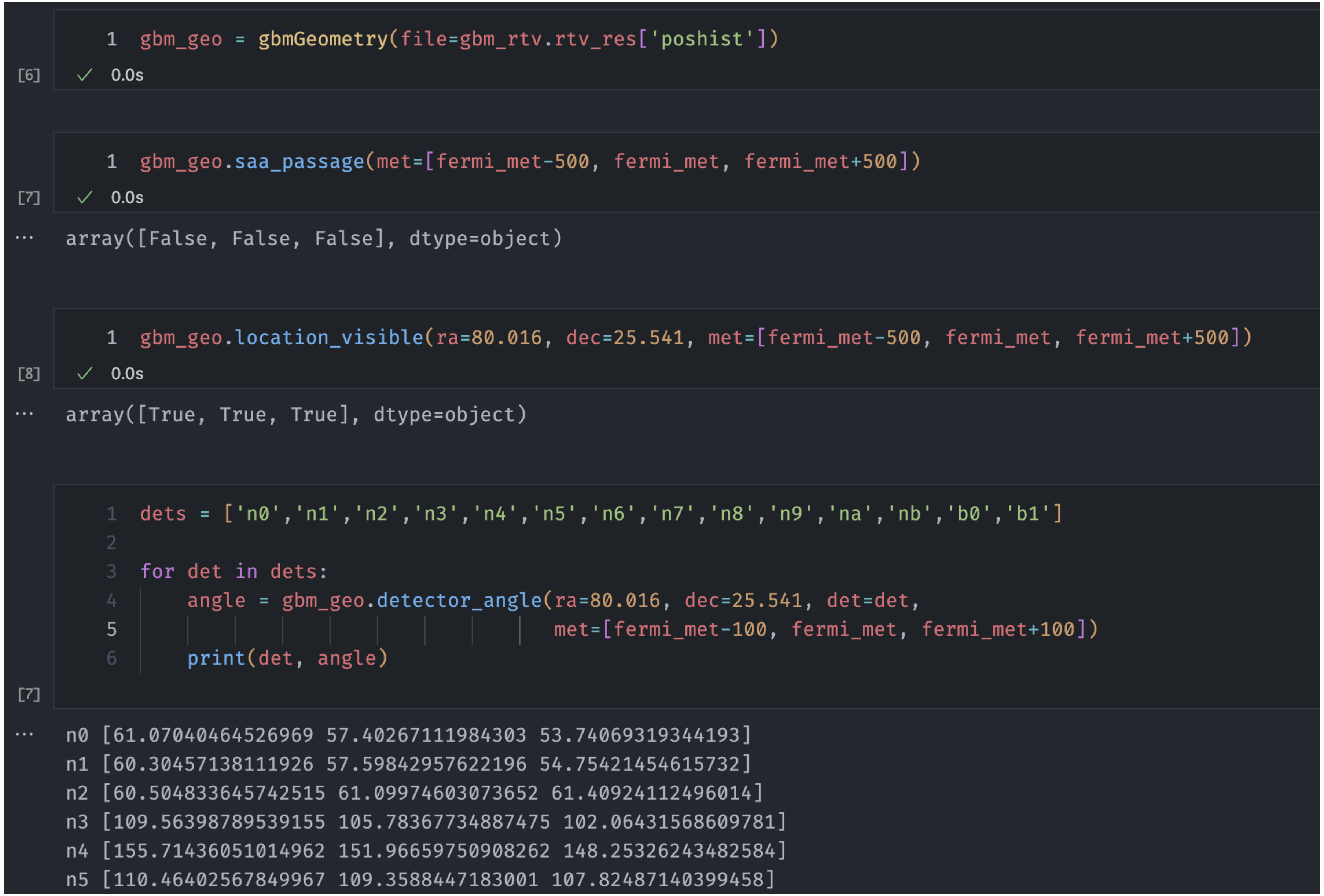
```
utc = '2024-02-19T06:21:42'
fermi_met = fermi_utc_to_met(utc)
```

```
gbm_rtv = gbmRetrieve.from_utc(utc=utc, t1=-400, t2=400)
gbm_rtv.rtv_res
```

```
Connected to FTP: 129.164.179.23
Downloading glg_tte_b1_240219_06z_v00.fit.gz: 100%|██████████| 1/1 [00:11<00:00, 11.94s/it]
```

```
{'utc': '2024-02-19T06:21:42.000',
 't1': -400.0,
 't2': 400.0,
 'datapath': '/Users/junyang/Data/fermi/data/gbm/daily',
 'tte': {'n0': ['/Users/junyang/Data/fermi/data/gbm/daily/2024/02/19/current/glg_tte_n0_240219_06z_v00.fit.gz'],
  'n1': ['/Users/junyang/Data/fermi/data/gbm/daily/2024/02/19/current/glg_tte_n1_240219_06z_v00.fit.gz'],
  'n2': ['/Users/junyang/Data/fermi/data/gbm/daily/2024/02/19/current/glg_tte_n2_240219_06z_v00.fit.gz'],
```

heapy example — Fermi/GBM data analysis



```python
gbm_geo = gbmGeometry(file=gbm_rtv.rtv_res['poshist'])
```

```python
gbm_geo.saa_passage(met=[fermi_met-500, fermi_met, fermi_met+500])
```

```
array([False, False, False], dtype=object)
```

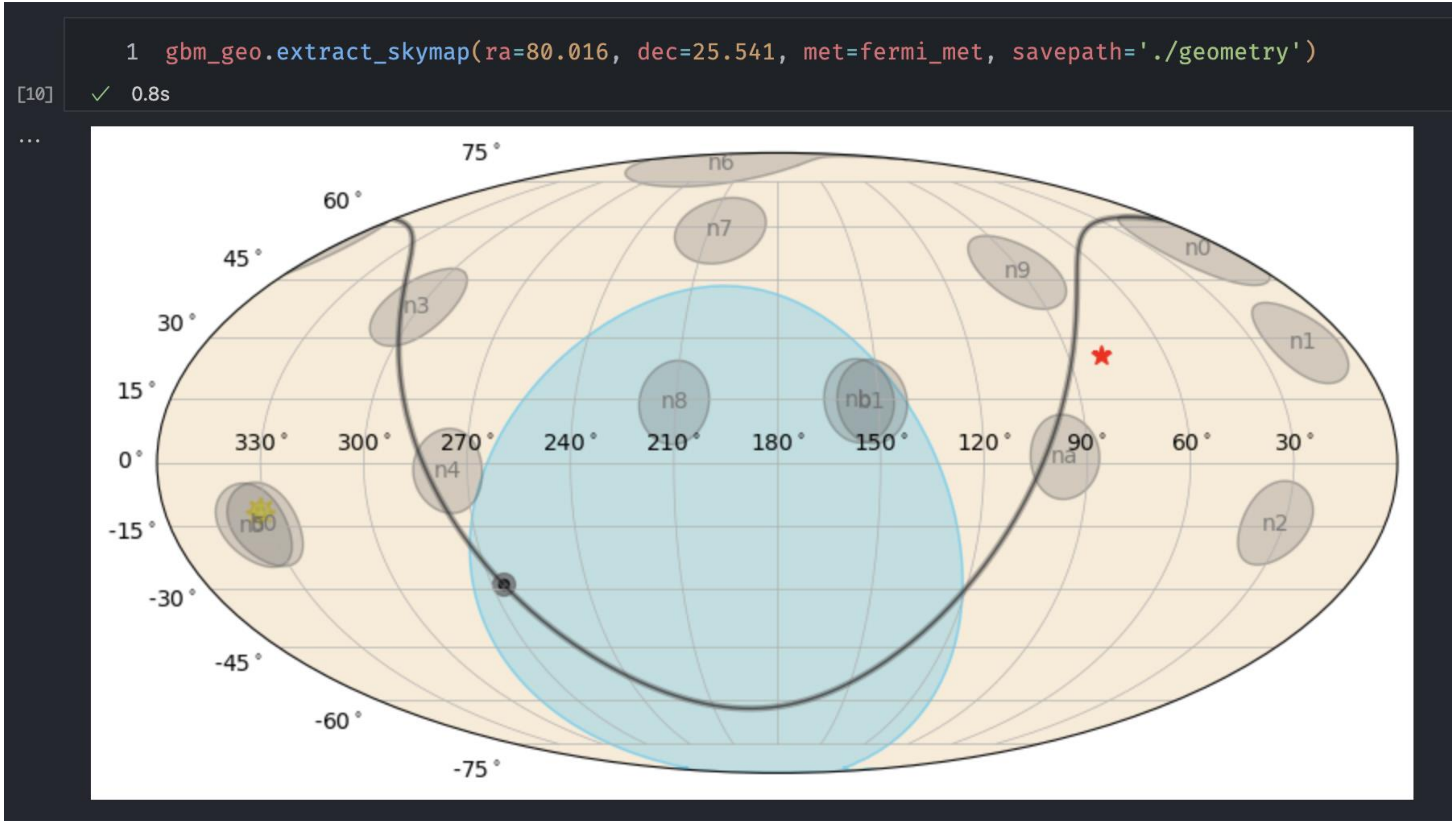
```python
gbm_geo.location_visible(ra=80.016, dec=25.541, met=[fermi_met-500, fermi_met, fermi_met+500])
```

```
array([True, True, True], dtype=object)
```

```python
dets = ['n0','n1','n2','n3','n4','n5','n6','n7','n8','n9','na','nb','b0','b1']

for det in dets:
    angle = gbm_geo.detector_angle(ra=80.016, dec=25.541, det=det,
                                   met=[fermi_met-100, fermi_met, fermi_met+100])
    print(det, angle)
```

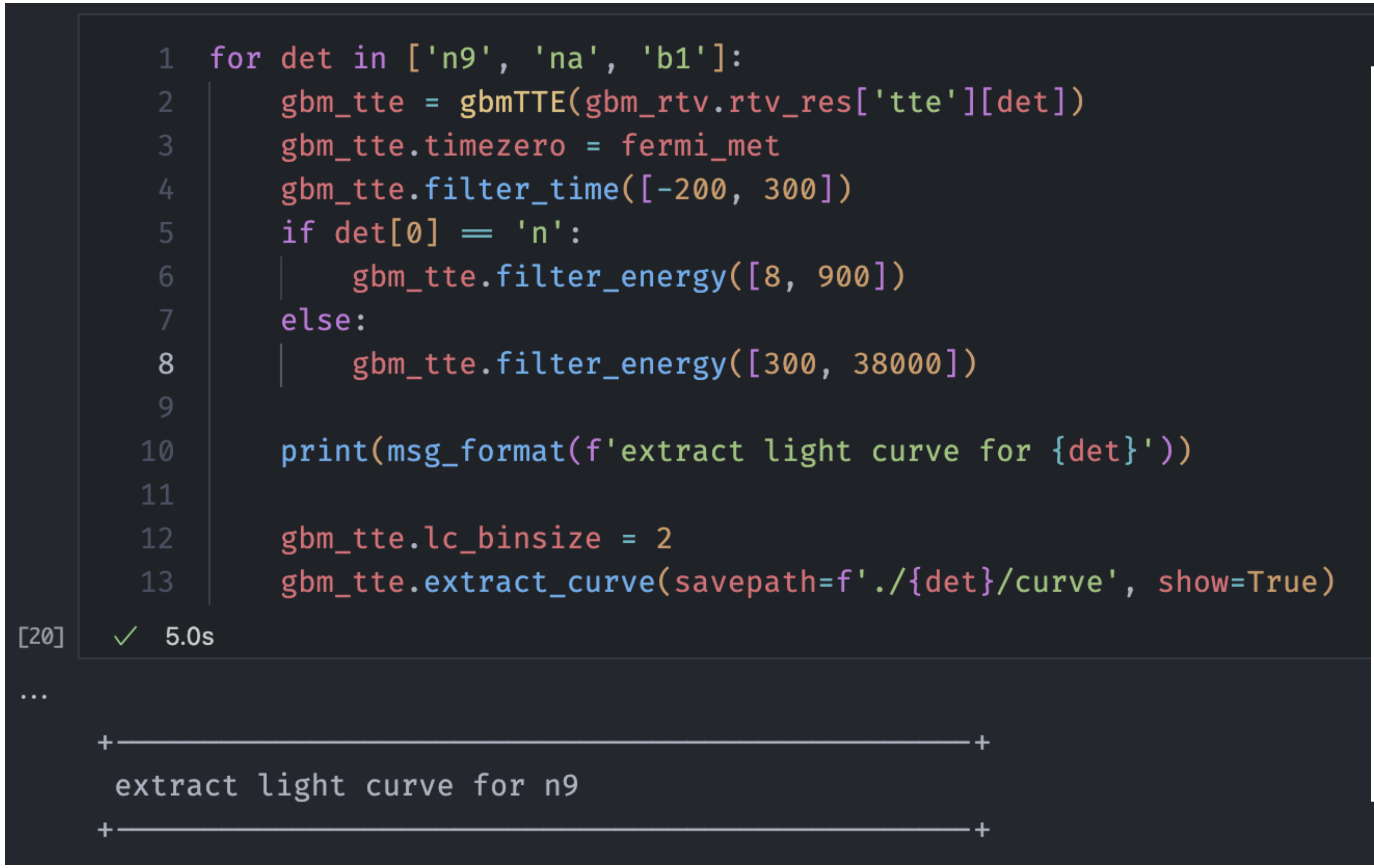
```
n0 [61.07040464526969 57.40267111984303 53.74069319344193]
n1 [60.30457138111926 57.59842957622196 54.75421454615732]
n2 [60.504833645742515 61.09974603073652 61.40924112496014]
n3 [109.56398789539155 105.78367734887475 102.06431568609781]
n4 [155.71436051014962 151.96659750908262 148.25326243482584]
n5 [110.46402567849967 109.3588447183001 107.82487140399458]
```

heapy example — Fermi/GBM data analysis



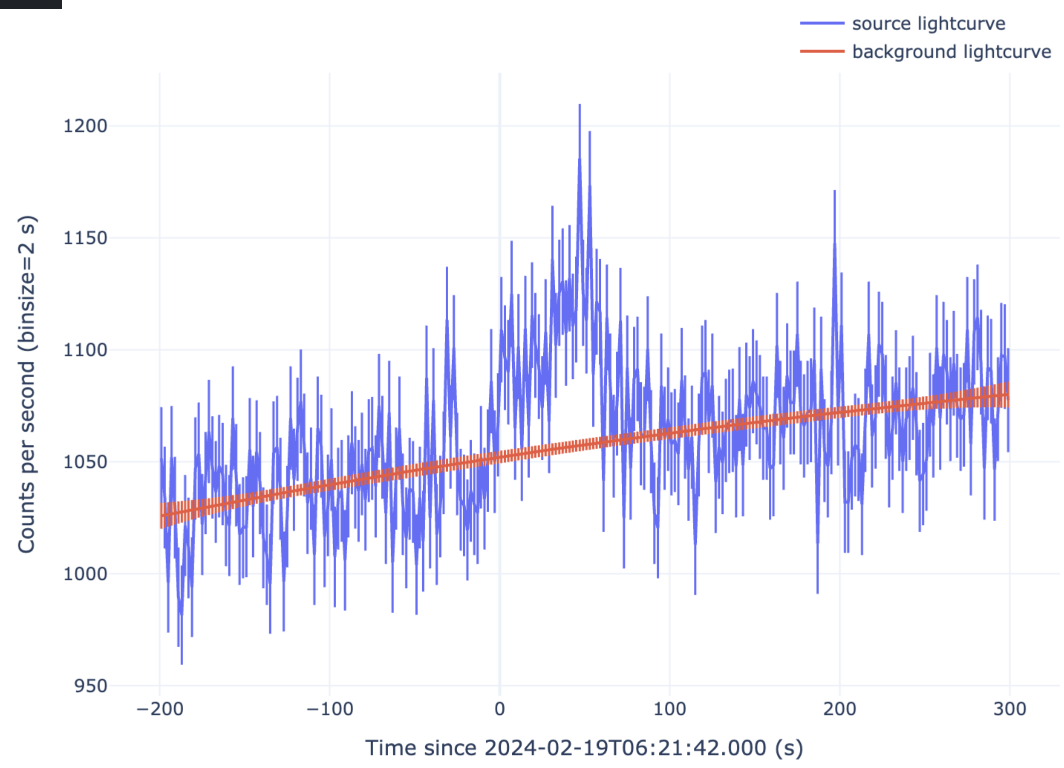
```
gbm_geo.extract_skymap(ra=80.016, dec=25.541, met=fermi_met, savepath='./geometry')
```

heapy example — Fermi/GBM data analysis

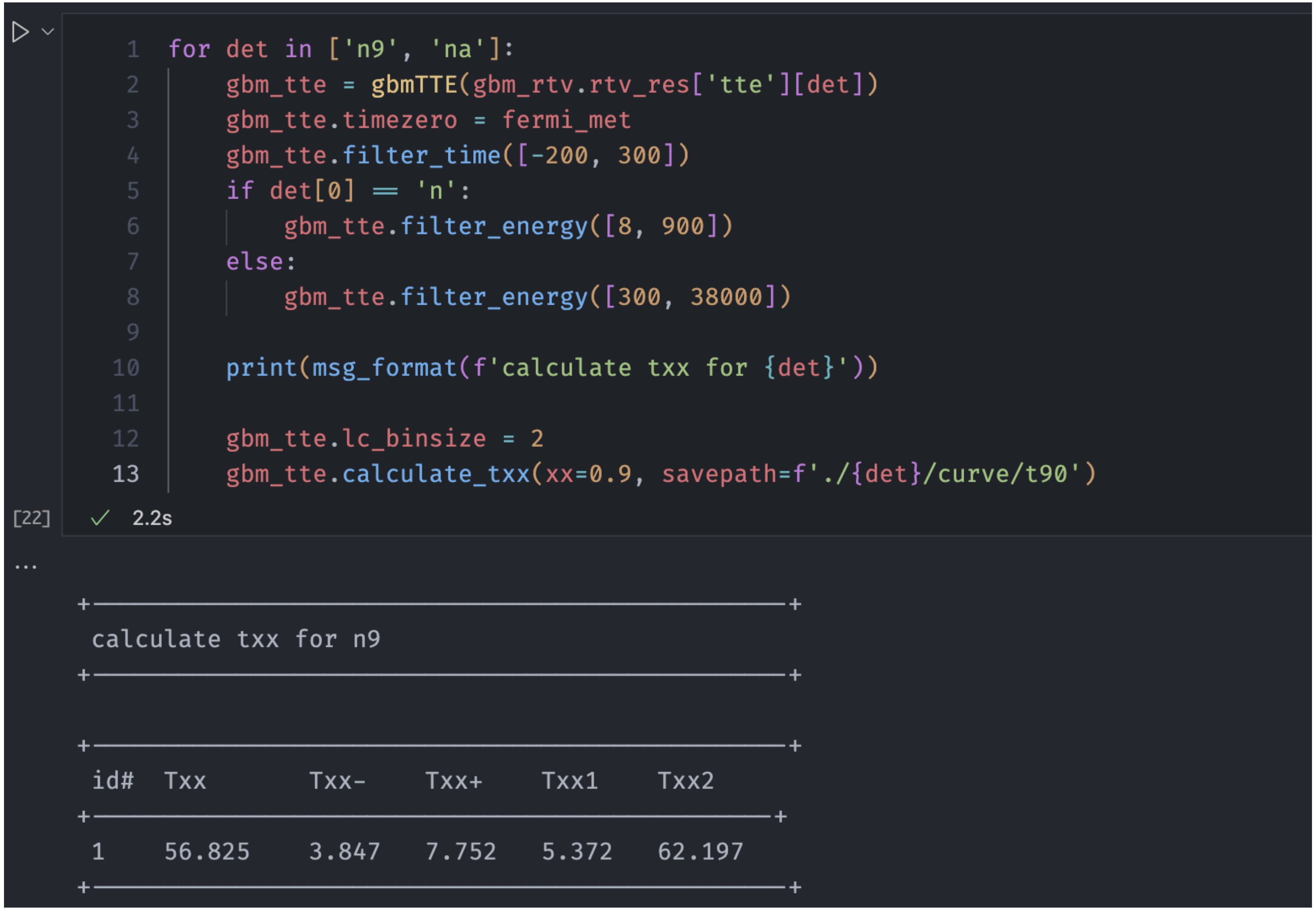


```python
for det in ['n9', 'na', 'b1']:
    gbm_tte = gbmTTE(gbm_rtv.rtv_res['tte'][det])
    gbm_tte.timezero = fermi_met
    gbm_tte.filter_time([-200, 300])
    if det[0] == 'n':
        gbm_tte.filter_energy([8, 900])
    else:
        gbm_tte.filter_energy([300, 38000])

    print(msg_format(f'extract light curve for {det}'))

    gbm_tte.lc_binsize = 2
    gbm_tte.extract_curve(savepath=f'./{det}/curve', show=True)
```

```
+------------------------------------------------+
 extract light curve for n9
+------------------------------------------------+
```



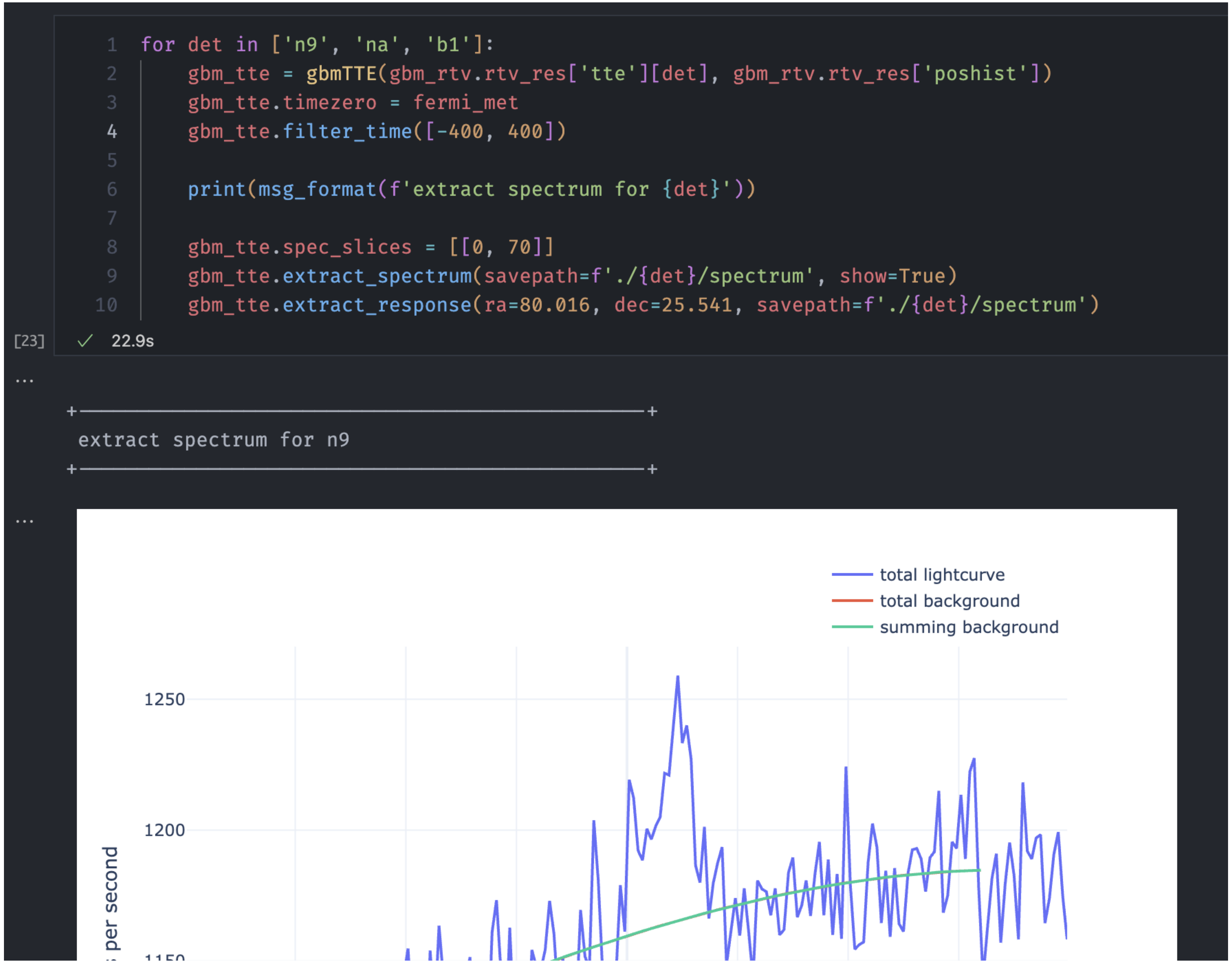
heapy example — Fermi/GBM data analysis



```python
for det in ['n9', 'na']:
    gbm_tte = gbmTTE(gbm_rtv.rtv_res['tte'][det])
    gbm_tte.timezero = fermi_met
    gbm_tte.filter_time([-200, 300])
    if det[0] == 'n':
        gbm_tte.filter_energy([8, 900])
    else:
        gbm_tte.filter_energy([300, 38000])

    print(msg_format(f'calculate txx for {det}'))

    gbm_tte.lc_binsize = 2
    gbm_tte.calculate_txx(xx=0.9, savepath=f'./{det}/curve/t90')
```

```
+----------------------------------------------------+
 calculate txx for n9
+----------------------------------------------------+

+----------------------------------------------------+
 id#  Txx       Txx-     Txx+     Txx1     Txx2
+---------------------------------------------------+
 1    56.825    3.847    7.752    5.372    62.197
+----------------------------------------------------+
```

heapy example — Fermi/GBM data analysis



```python
for det in ['n9', 'na', 'b1']:
    gbm_tte = gbmTTE(gbm_rtv.rtv_res['tte'][det], gbm_rtv.rtv_res['poshist'])
    gbm_tte.timezero = fermi_met
    gbm_tte.filter_time([-400, 400])

    print(msg_format(f'extract spectrum for {det}'))

    gbm_tte.spec_slices = [[0, 70]]
    gbm_tte.extract_spectrum(savepath=f'./{det}/spectrum', show=True)
    gbm_tte.extract_response(ra=80.016, dec=25.541, savepath=f'./{det}/spectrum')
```

```
+──────────────────────────────────────────────+
extract spectrum for n9
+──────────────────────────────────────────────+
```

heapy example — joint light curve

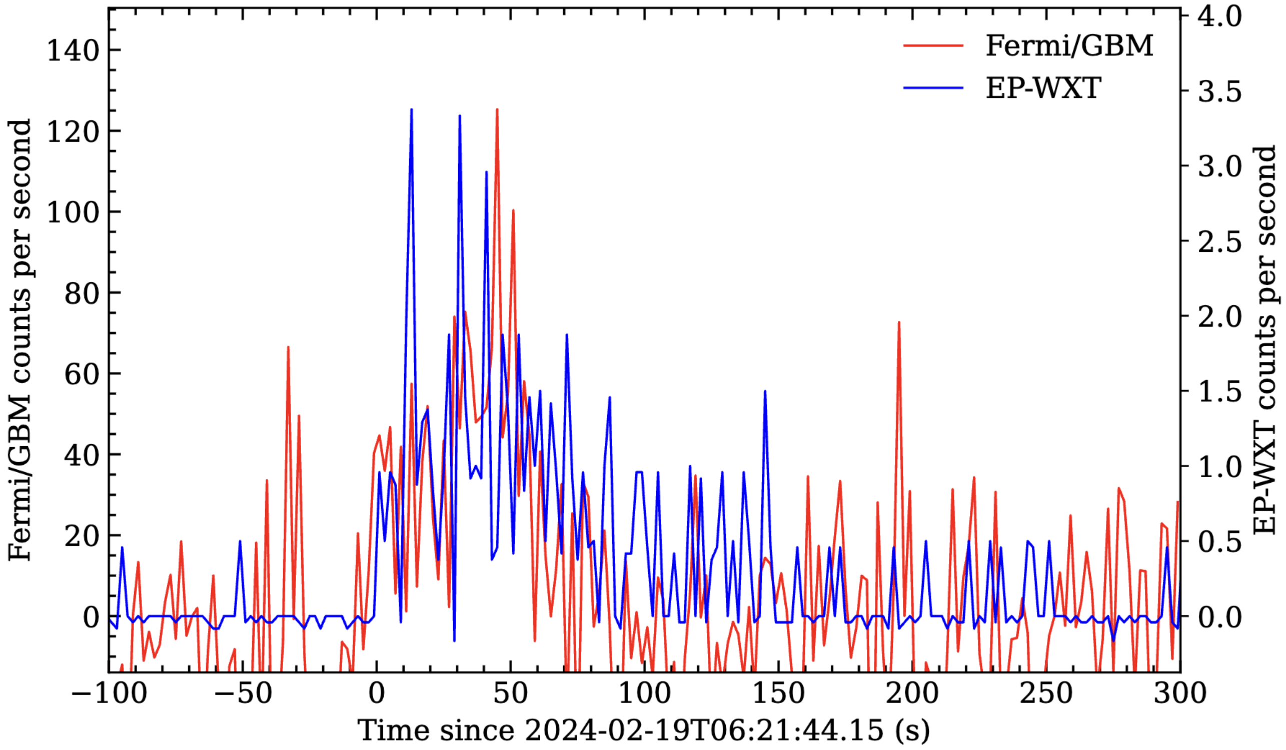
```python
import numpy as np
import matplotlib.pyplot as plt
from matplotlib import rcParams
from heapy.pipe.event import gbmTTE
from heapy.pipe.image import epImage
from heapy.util.time import fermi_utc_to_met, ep_utc_to_met
```

[2] ✓ 0.0s

```python
utc = '2024-02-19T06:21:44.15'
fermi_met = fermi_utc_to_met(utc)
ep_met = ep_utc_to_met(utc)

gbm_tte = gbmTTE.from_utc(utc=utc, det='n9')
gbm_tte.timezero = fermi_met
gbm_tte.filter_time([-200, 300])
gbm_tte.filter_energy([8, 900])
gbm_tte.lc_binsize = 2
gbm_lc_time = gbm_tte.lc_time
gbm_lc_net_rate = gbm_tte.lc_net_rate

wxt_event = epImage.from_wxtobs('ep13600003859wxtCMOS44l23v2', 's2')
wxt_event.timezero = ep_met
wxt_event.filter_time([-200, 465])
wxt_event.filter_pi([50, 400])
wxt_event.lc_binsize = 2
wxt_lc_time = wxt_event.lc_time
wxt_lc_net_rate = wxt_event.lc_net_rate
```

[3] ✓ 3.0s

```python
ax.plot(gbm_lc_time, gbm_lc_net_rate)
ax.plot(wxt_lc_time, wxt_lc_net_rate)
```



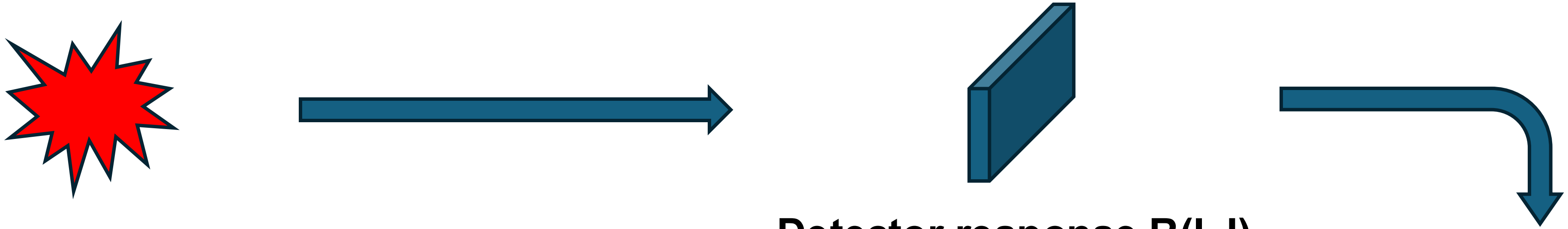
Part 4: bayspec

https://github.com/jyangch/bayspec

A Bayesian Inference-based Spectral Fitting Tool for High-energy Astrophysical Data

Other options: xspec, sherpa, threeml

bayspec: source -> observation



Source

Physical spectrum f(E,t)

Detector response R(I,J)
Scatter & Absorption

Observed spectrum D(I,t)

f(E,t): photons/s/cm^2/keV

R(I,J): cm^2

D(I,t): counts/s/channel

$$D(I,t) = \int_{E_J^1}^{E_J^2} f(E,t)dE \times R(I,J)$$

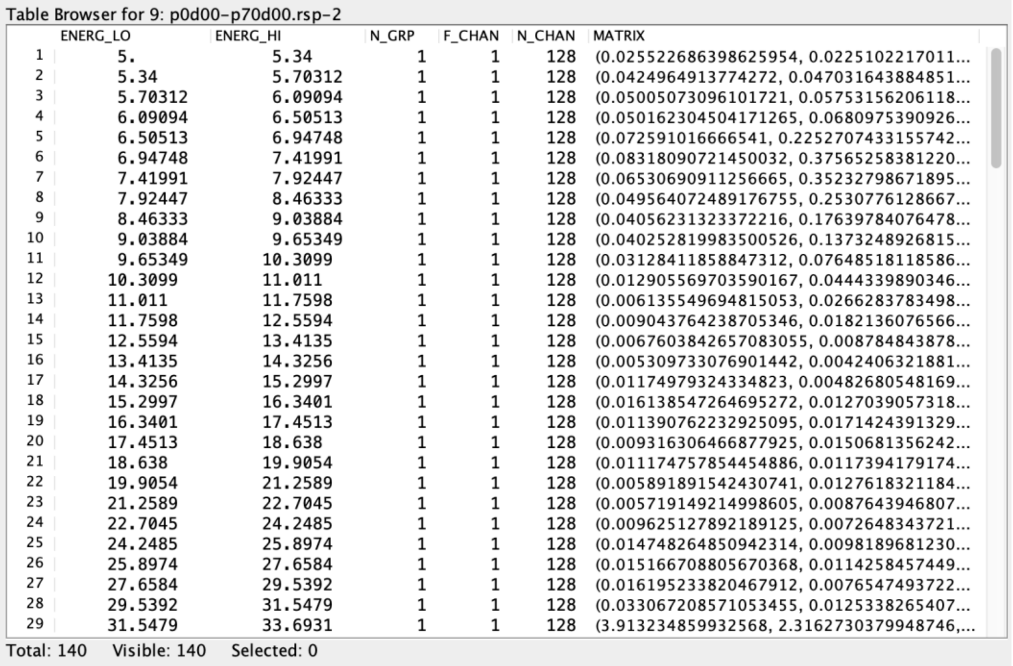
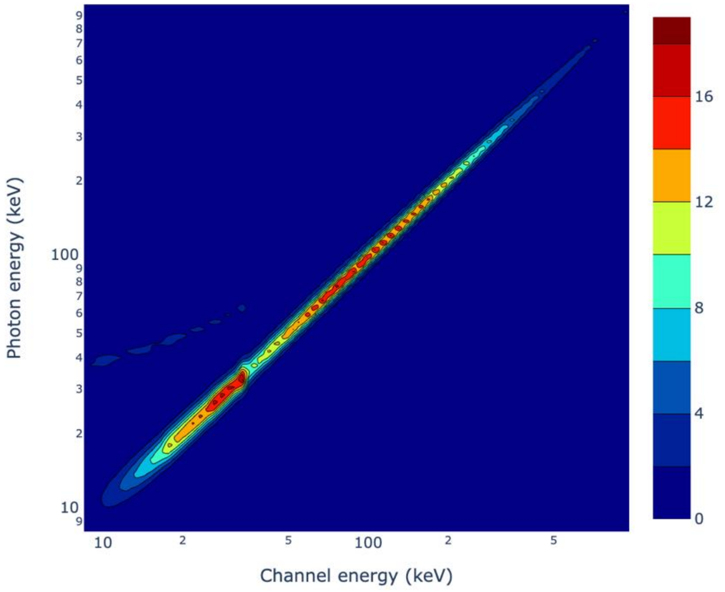


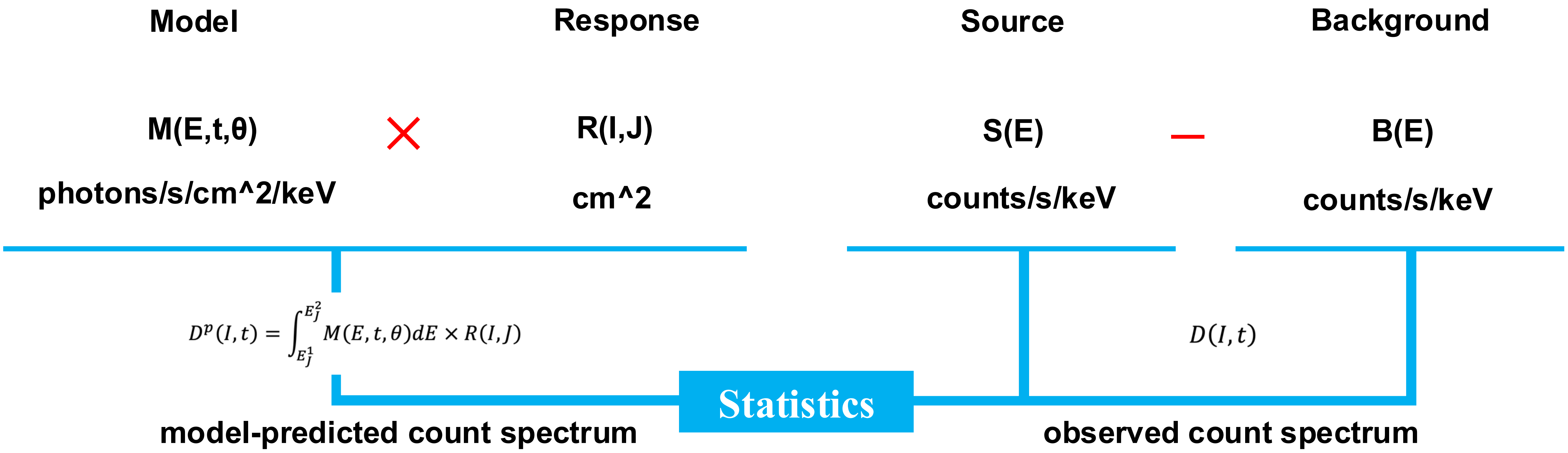
Table Browser for 9: p0d00-p70d00.rsp-2

	ENERG_LO	ENERG_HI	N_GRP	F_CHAN	N_CHAN	MATRIX
1	5.	5.34	1	1	128	(0.025522686398625954, 0.0225102217011...
2	5.34	5.70312	1	1	128	(0.0424964913774272, 0.047031643884851...
3	5.70312	6.09094	1	1	128	(0.05005073096101721, 0.05753156206118...
4	6.09094	6.50513	1	1	128	(0.050162304504171265, 0.0680975390926...
5	6.50513	6.94748	1	1	128	(0.07259101666541, 0.2252707433155742...
6	6.94748	7.41991	1	1	128	(0.08318090721450032, 0.37565258381220...
7	7.41991	7.92447	1	1	128	(0.06530690911256665, 0.35232798671895...
8	7.92447	8.46333	1	1	128	(0.049564072489176755, 0.2530776128667...
9	8.46333	9.03884	1	1	128	(0.04056231323372216, 0.17639784076478...
10	9.03884	9.65349	1	1	128	(0.040252819983500526, 0.1373248926815...
11	9.65349	10.3099	1	1	128	(0.03128411858847312, 0.07648518118586...
12	10.3099	11.011	1	1	128	(0.012905569703590167, 0.0444339890346...
13	11.011	11.7598	1	1	128	(0.006135549694815053, 0.0266283783498...
14	11.7598	12.5594	1	1	128	(0.009043764238705346, 0.0182136076566...
15	12.5594	13.4135	1	1	128	(0.006760384265708305 5, 0.00878484387 8...
16	13.4135	14.3256	1	1	128	(0.005309733076901442, 0.0042406321881...
17	14.3256	15.2997	1	1	128	(0.01174979324334823, 0.00482680548169...
18	15.2997	16.3401	1	1	128	(0.016138547264695272, 0.012703905731 8...
19	16.3401	17.4513	1	1	128	(0.011390762232925095, 0.0171424391329...
20	17.4513	18.638	1	1	128	(0.009316306466877925, 0.0150681356242...
21	18.638	19.9054	1	1	128	(0.011174757854454886, 0.0117394179174...
22	19.9054	21.2589	1	1	128	(0.005891891542430741, 0.0127618321184...
23	21.2589	22.7045	1	1	128	(0.005719149214998605, 0.0087643946807...
24	22.7045	24.2485	1	1	128	(0.009625127892189125, 0.0072648343721...
25	24.2485	25.8974	1	1	128	(0.014748264850942314, 0.0098189681230...
26	25.8974	27.6584	1	1	128	(0.015166708805670368, 0.0114258457449...
27	27.6584	29.5392	1	1	128	(0.016195233820467912, 0.0076547493722...
28	29.5392	31.5479	1	1	128	(0.033067208571053455, 0.0125338265407...
29	31.5479	33.6931	1	1	128	(3.913234859932568, 2.3162730379948746,...

Total: 140 Visible: 140 Selected: 0



bayspec: compare model and data



Model | **Response** | **Source** | **Background**

M(E,t,θ) × **R(I,J)**

photons/s/cm^2/keV | **cm^2**

S(E) — **B(E)**

counts/s/keV | **counts/s/keV**

$$D^p(I,t) = \int_{E_J^1}^{E_J^2} M(E,t,\theta)dE \times R(I,J)$$

$$D(I,t)$$

Statistics

model-predicted count spectrum

observed count spectrum

Poisson data with Poisson background (cstat):

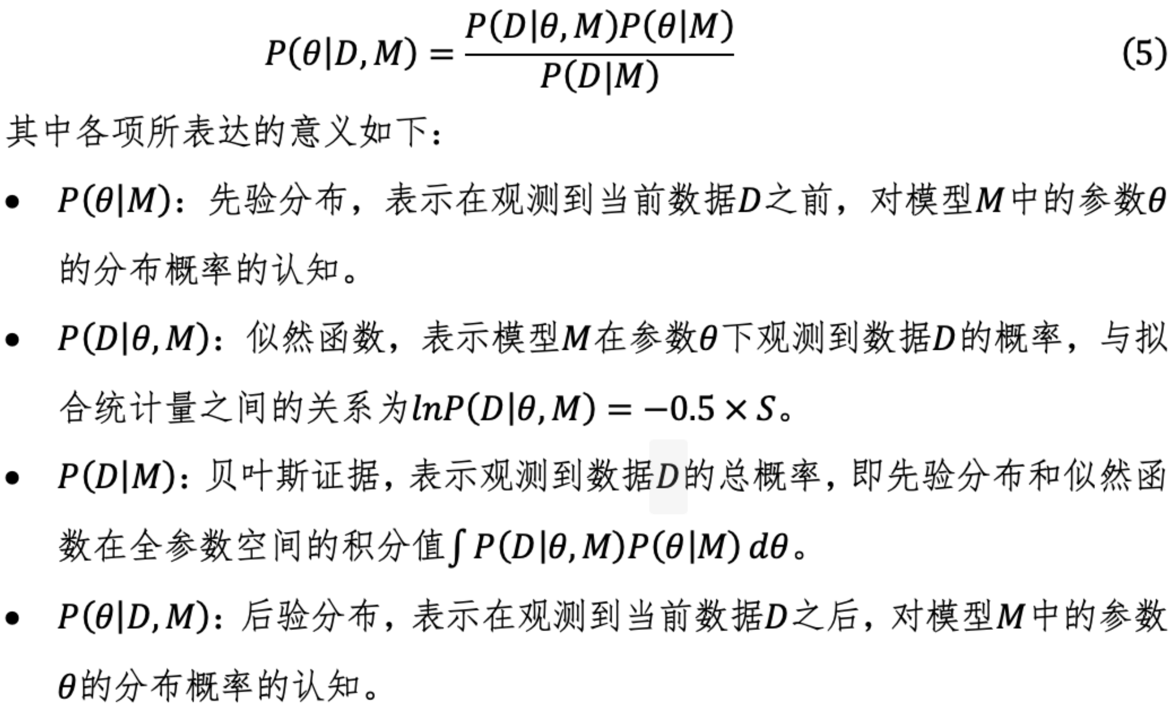
$$L = \prod_{i=1}^{N} \frac{(t_s(m_i + b_i))^{S_i} \mathrm{e}^{-t_s(m_i+b_i)}}{S_i!} \times \frac{(t_b b_i)^{B_i} \mathrm{e}^{-t_b b_i}}{B_i!}$$

Poisson data with Gaussian background (pgstat):

$$L = \prod_{i=1}^{N} \frac{(t_s\,(b_i + m_i))^{S_i} e^{-t_s(b_i+m_i)}}{S_i\,!} \times e^{-\frac{(B_i - b_i t_b)^2}{2\sigma_i^2}}$$

bayspec: bayesian inference

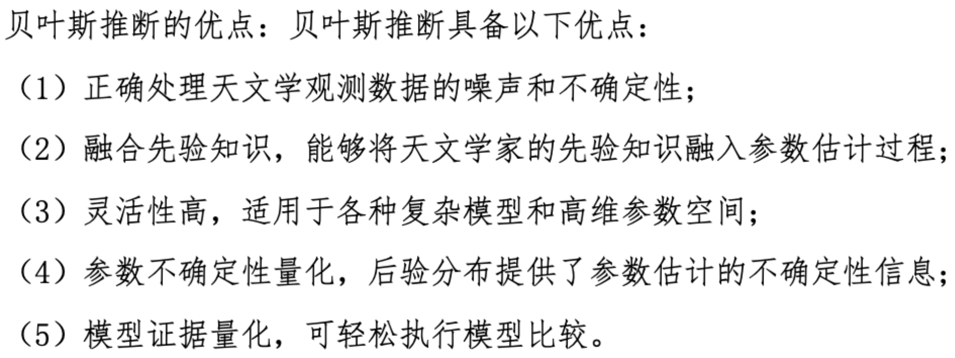
Bayes' theorem



$$P(\theta|D,M) = \frac{P(D|\theta,M)P(\theta|M)}{P(D|M)} \qquad (5)$$

其中各项所表达的意义如下：

- $P(\theta|M)$：先验分布，表示在观测到当前数据D之前，对模型M中的参数θ的分布概率的认知。
- $P(D|\theta,M)$：似然函数，表示模型M在参数θ下观测到数据D的概率，与拟合统计量之间的关系为$lnP(D|\theta,M) = -0.5 \times S$。
- $P(D|M)$：贝叶斯证据，表示观测到数据D的总概率，即先验分布和似然函数在全参数空间的积分值$\int P(D|\theta,M)P(\theta|M)\,d\theta$。
- $P(\theta|D,M)$：后验分布，表示在观测到当前数据D之后，对模型M中的参数θ的分布概率的认知。



贝叶斯推断的优点：贝叶斯推断具备以下优点：

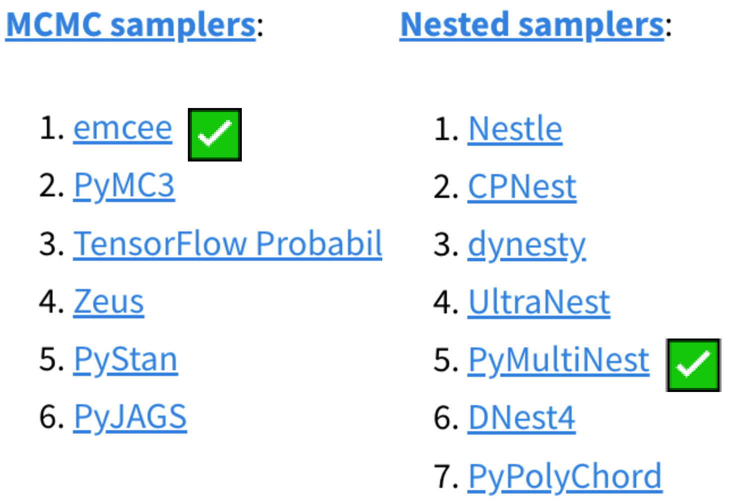
（1）正确处理天文学观测数据的噪声和不确定性；

（2）融合先验知识，能够将天文学家的先验知识融入参数估计过程；

（3）灵活性高，适用于各种复杂模型和高维参数空间；

（4）参数不确定性量化，后验分布提供了参数估计的不确定性信息；

（5）模型证据量化，可轻松执行模型比较。



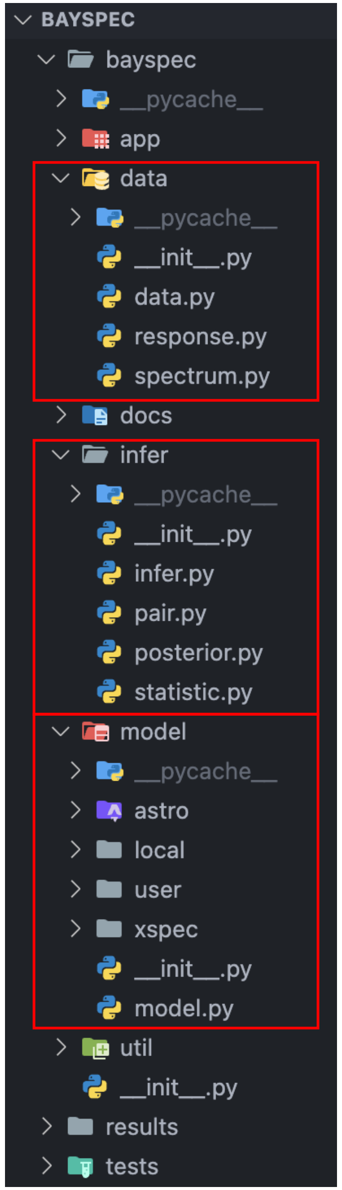
MCMC samplers:

1. emcee ✓
2. PyMC3
3. TensorFlow Probabil
4. Zeus
5. PyStan
6. PyJAGS

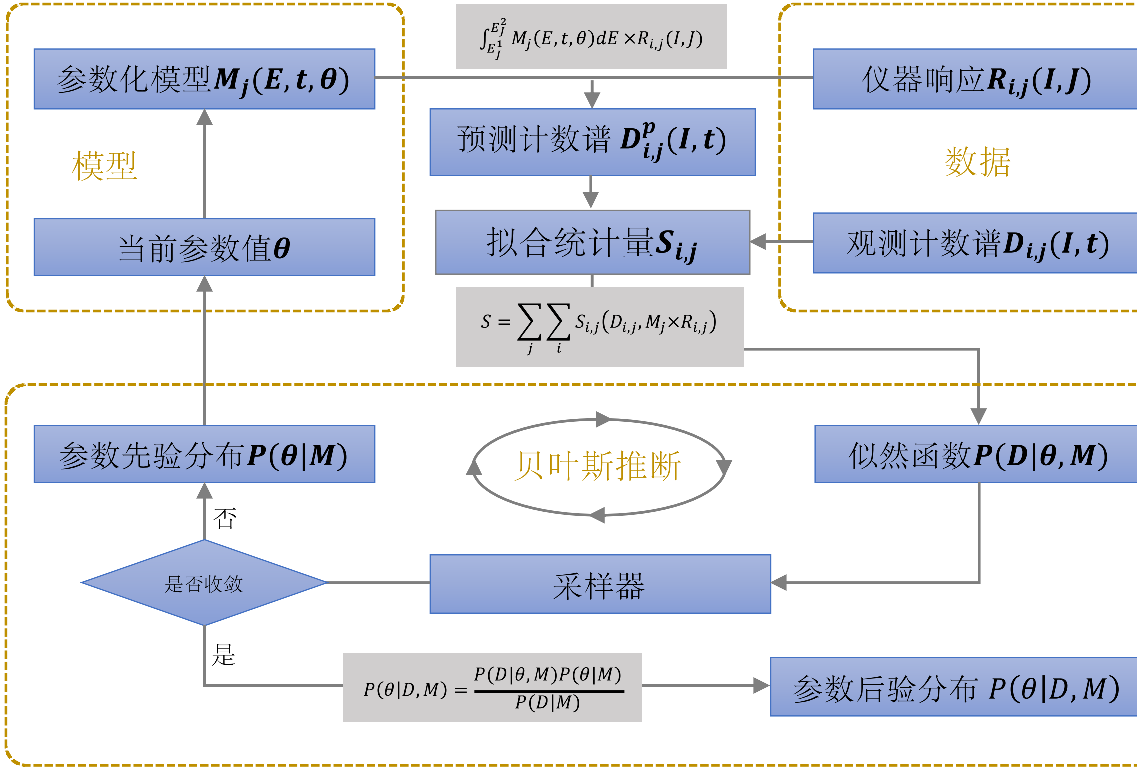
Nested samplers:

1. Nestle
2. CPNest
3. dynesty
4. UltraNest
5. PyMultiNest ✓
6. DNest4
7. PyPolyChord

bayspec package structure



```
BAYSPEC
  bayspec
    __pycache__
    app
    data
      __pycache__
      __init__.py
      data.py
      response.py
      spectrum.py
    docs
    infer
      __pycache__
      __init__.py
      infer.py
      pair.py
      posterior.py
      statistic.py
    model
      __pycache__
      astro
      local
      user
      xspec
      __init__.py
      model.py
    util
    __init__.py
  results
  tests
```



参数化模型$M_j(E,t,\theta)$

$\int_{E_j^1}^{E_j^2} M_j(E,t,\theta)dE \times R_{i,j}(I,J)$

仪器响应$R_{i,j}(I,J)$

模型

预测计数谱 $D_{i,j}^p(I,t)$

数据

当前参数值θ

拟合统计量$S_{i,j}$

观测计数谱$D_{i,j}(I,t)$

$S = \sum_j \sum_i S_{i,j}(D_{i,j}, M_j \times R_{i,j})$

参数先验分布$P(\theta|M)$

贝叶斯推断

似然函数$P(D|\theta,M)$

否

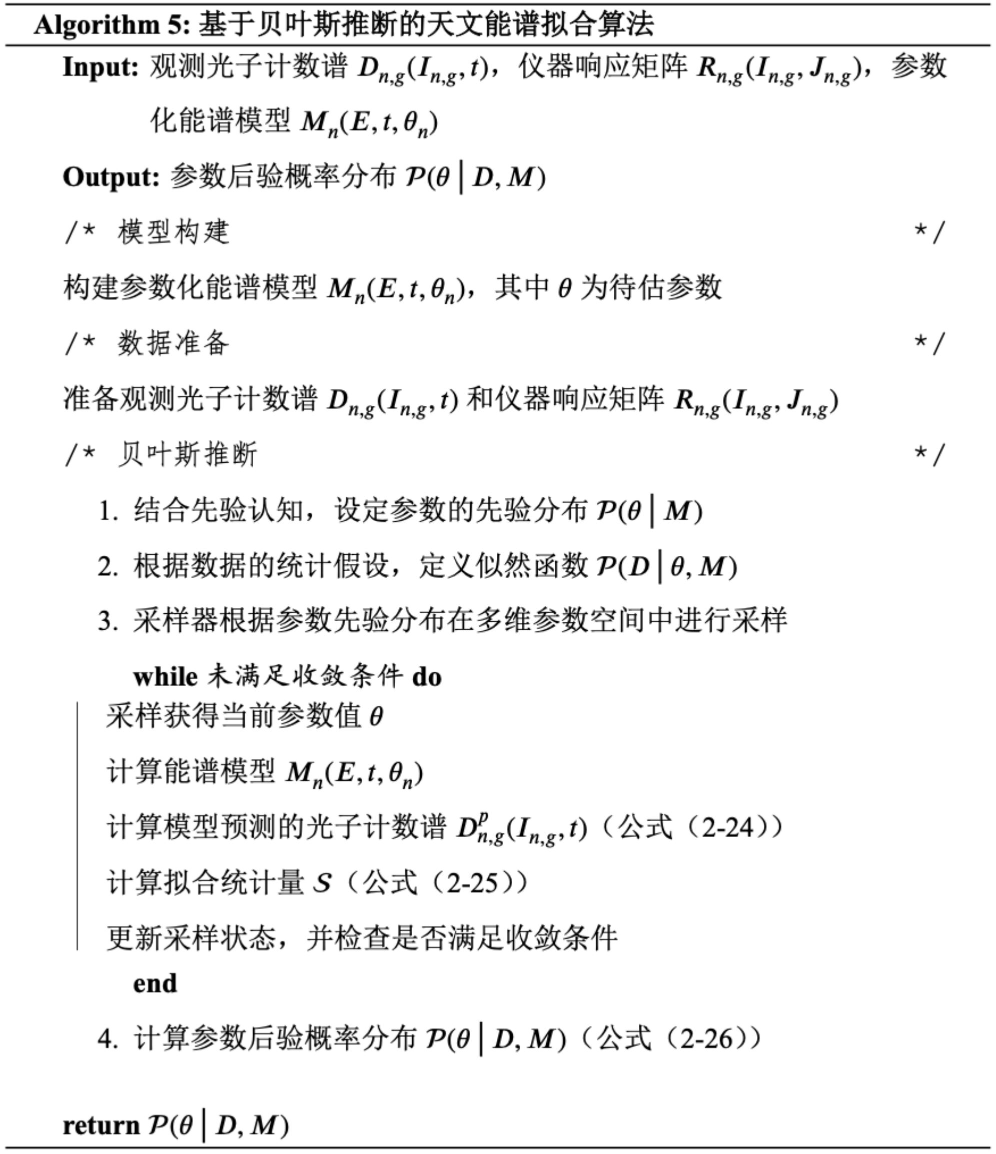
是否收敛

采样器

是

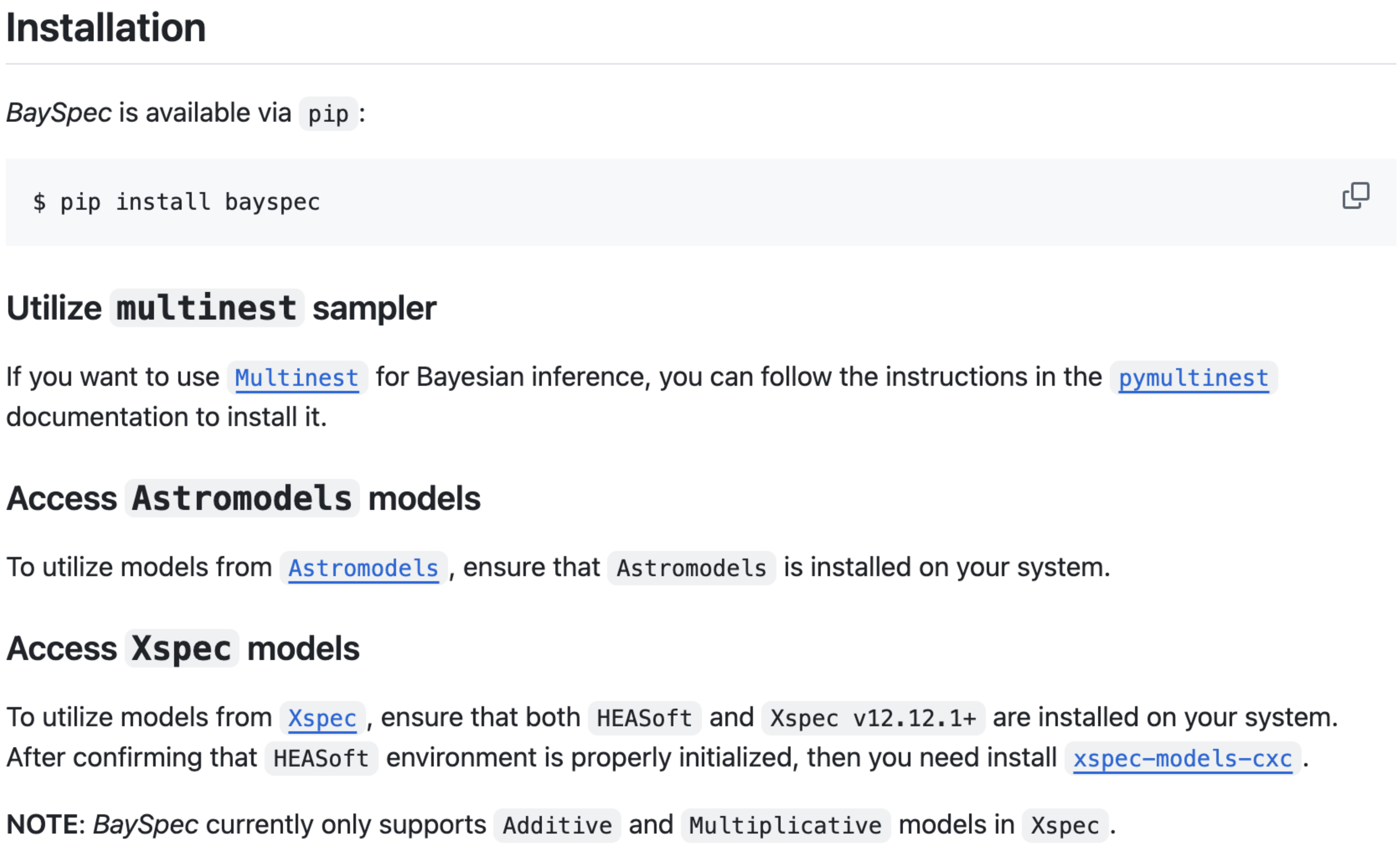
$P(\theta|D,M) = \frac{P(D|\theta,M)P(\theta|M)}{P(D|M)}$

参数后验分布 $P(\theta|D,M)$



```
Algorithm 5: 基于贝叶斯推断的天文能谱拟合算法
  Input: 观测光子计数谱 $D_{n,g}(I_{n,g},t)$，仪器响应矩阵 $R_{n,g}(I_{n,g},J_{n,g})$，参数
      化能谱模型 $M_n(E,t,\theta_n)$
  Output: 参数后验概率分布 $\mathcal{P}(\theta \mid D,M)$
  /* 模型构建                                              */
  构建参数化能谱模型 $M_n(E,t,\theta_n)$，其中 $\theta$ 为待估参数
  /* 数据准备                                              */
  准备观测光子计数谱 $D_{n,g}(I_{n,g},t)$ 和仪器响应矩阵 $R_{n,g}(I_{n,g},J_{n,g})$
  /* 贝叶斯推断                                            */
    1. 结合先验认知，设定参数的先验分布 $\mathcal{P}(\theta \mid M)$
    2. 根据数据的统计假设，定义似然函数 $\mathcal{P}(D \mid \theta,M)$
    3. 采样器根据参数先验分布在多维参数空间中进行采样
      while 未满足收敛条件 do
        采样获得当前参数值 $\theta$
        计算能谱模型 $M_n(E,t,\theta_n)$
        计算模型预测的光子计数谱 $D_{n,g}^p(I_{n,g},t)$（公式（2-24））
        计算拟合统计量 $S$（公式（2-25））
        更新采样状态，并检查是否满足收敛条件
      end
    4. 计算参数后验概率分布 $\mathcal{P}(\theta \mid D,M)$（公式（2-26））

  return $\mathcal{P}(\theta \mid D,M)$
```

bayspec installation: pip3 install bayspec



Installation

BaySpec is available via `pip`:

```
$ pip install bayspec
```

Utilize `multinest` sampler

If you want to use `Multinest` for Bayesian inference, you can follow the instructions in the `pymultinest` documentation to install it.

Access `Astromodels` models

To utilize models from `Astromodels`, ensure that `Astromodels` is installed on your system.

Access `Xspec` models

To utilize models from `Xspec`, ensure that both `HEASoft` and `Xspec v12.12.1+` are installed on your system. After confirming that `HEASoft` environment is properly initialized, then you need install `xspec-models-cxc`.

NOTE: *BaySpec* currently only supports `Additive` and `Multiplicative` models in `Xspec`.

bayspec example: LE+ME (tbabs*cpl)

Data:

LE (0.5-4 keV, cstat)

ME (8-900 keV, pgstat)

Model:

tbabs (all free)

cpl (all free)

Sampler:

emcee

```
import numpy as np
from bayspec.model.local import *
from bayspec.model.astro import *
from bayspec import DataUnit, Data, Infer, Plot
```

[2] ✓ 0.0s

```
le = DataUnit(
    src='./LE/le2.src',
    bkg='./LE/le2.bkg',
    rmf='./LE/le2.rmf',
    arf='./LE/le2.arf',
    notc=[0.5, 4],
    stat='cstat',
    grpg={'min_evt': 5})

me = DataUnit(
    src='./ME/me2.src',
    bkg='./ME/me2.bkg',
    rsp='./ME/me2.rsp',
    notc=[8, 900],
    stat='pgstat',
    grpg={'min_evt': 50})

data = Data([('LE', le), ('ME', me)])

print(data)
```

[3] ✓ 0.1s

```
| Name | Noticing   | Statistic | Grouping         | Time |
| LE   | [[0.5, 4]] | cstat     | {'min_evt': 5}   | None |
| ME   | [[8, 900]] | pgstat    | {'min_evt': 50}  | None |
```

bayspec example: LE+ME (tbabs*cpl)

Data:

LE (0.5-4 keV, cstat)

ME (8-900 keV, pgstat)

Model:

tbabs (all free)

cpl (all free)

Sampler:

emcee

```python
tbabs_ins = AS_TbAbs()

cpl_ins = cpl()

model = tbabs_ins * cpl_ins

print(model)
```

```
(AS_TbAbs*cpl) [add]
AS_TbAbs: Photometric absorption (Tbabs implementation), f(E) = exp(- NH * sigma(E)) contributed by Dominique Eckert
cpl: cutoff power law model

 cfg#   Component   Parameter   Value
  1     AS_TbAbs    redshift      0
  2       cpl       redshift      0

 par#   Component   Parameter   Value       Prior        Frozen
  1     AS_TbAbs       NH         1    unif(0.0001, 20)   False
  2       cpl      $\alpha$      -1      unif(-8, 4)      False
  3       cpl      log$E_{c}$     2      unif(0, 4)       False
  4       cpl       log$A$       -1      unif(-6, 5)      False
```

bayspec example: LE+ME (tbabs*cpl)

Data:

LE (0.5-4 keV, cstat)

ME (8-900 keV, pgstat)

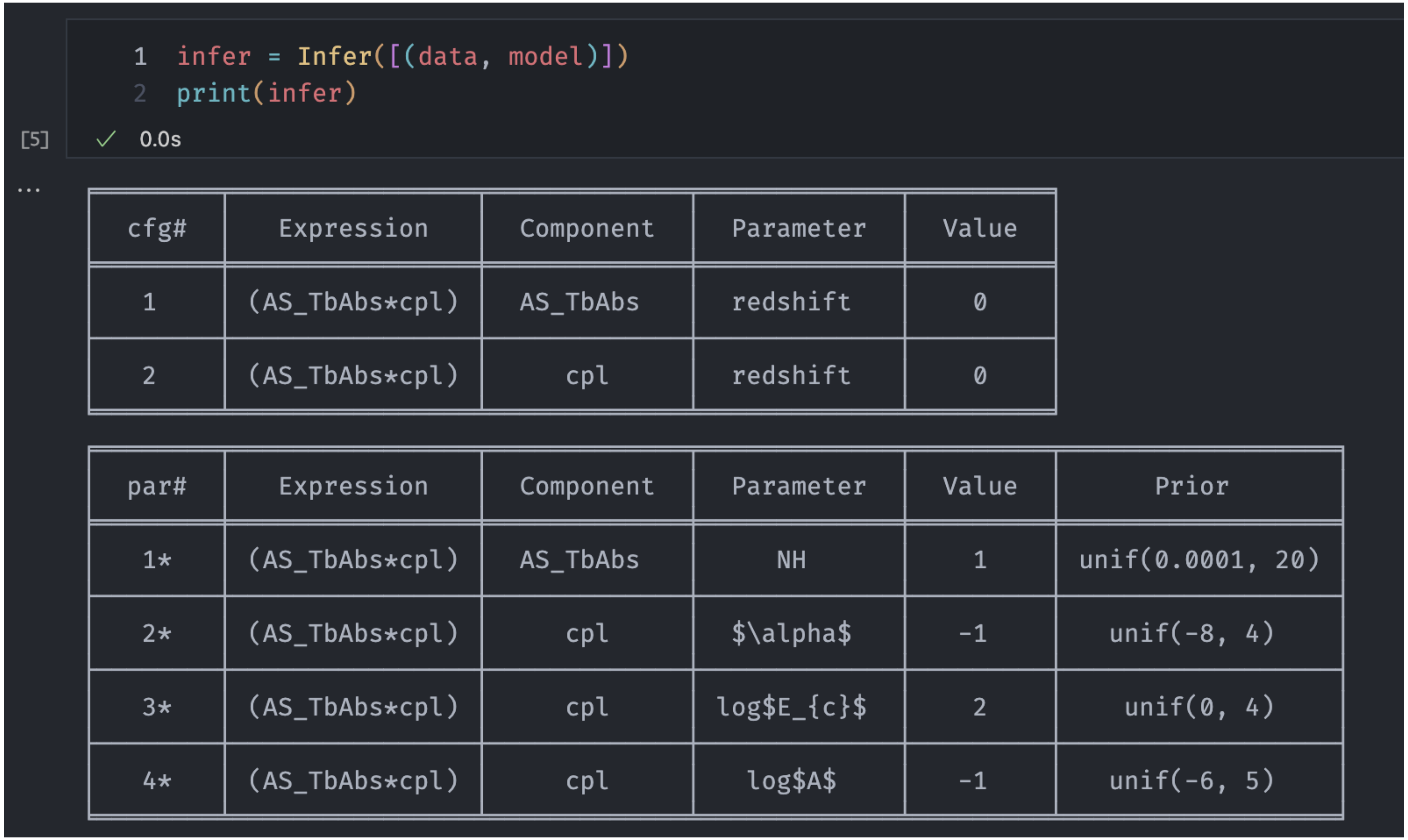
Model:

tbabs (all free)

cpl (all free)

Sampler:

emcee



```
infer = Infer([(data, model)])
print(infer)
```

[5] ✓ 0.0s

cfg#	Expression	Component	Parameter	Value
1	(AS_TbAbs*cpl)	AS_TbAbs	redshift	0
2	(AS_TbAbs*cpl)	cpl	redshift	0

par#	Expression	Component	Parameter	Value	Prior
1*	(AS_TbAbs*cpl)	AS_TbAbs	NH	1	unif(0.0001, 20)
2*	(AS_TbAbs*cpl)	cpl	α	-1	unif(-8, 4)
3*	(AS_TbAbs*cpl)	cpl	logE_{c}	2	unif(0, 4)
4*	(AS_TbAbs*cpl)	cpl	log$A$	-1	unif(-6, 5)

bayspec example: LE+ME (tbabs*cpl)

Data:

LE (0.5-4 keV, cstat)

ME (8-900 keV, pgstat)

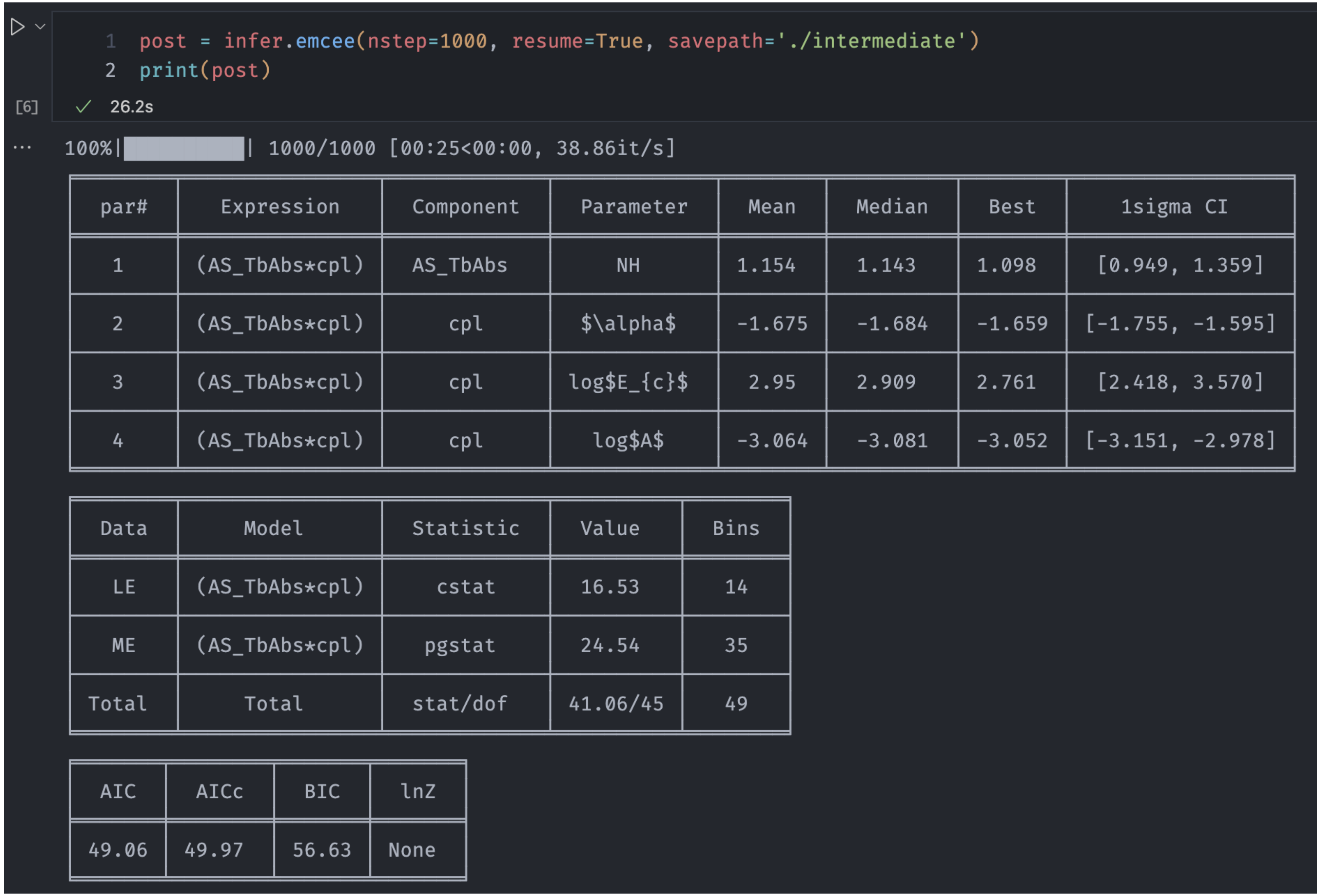
Model:

tbabs (all free)

cpl (all free)

Sampler:

emcee



```python
post = infer.emcee(nstep=1000, resume=True, savepath='./intermediate')
print(post)
```

[6] ✓ 26.2s

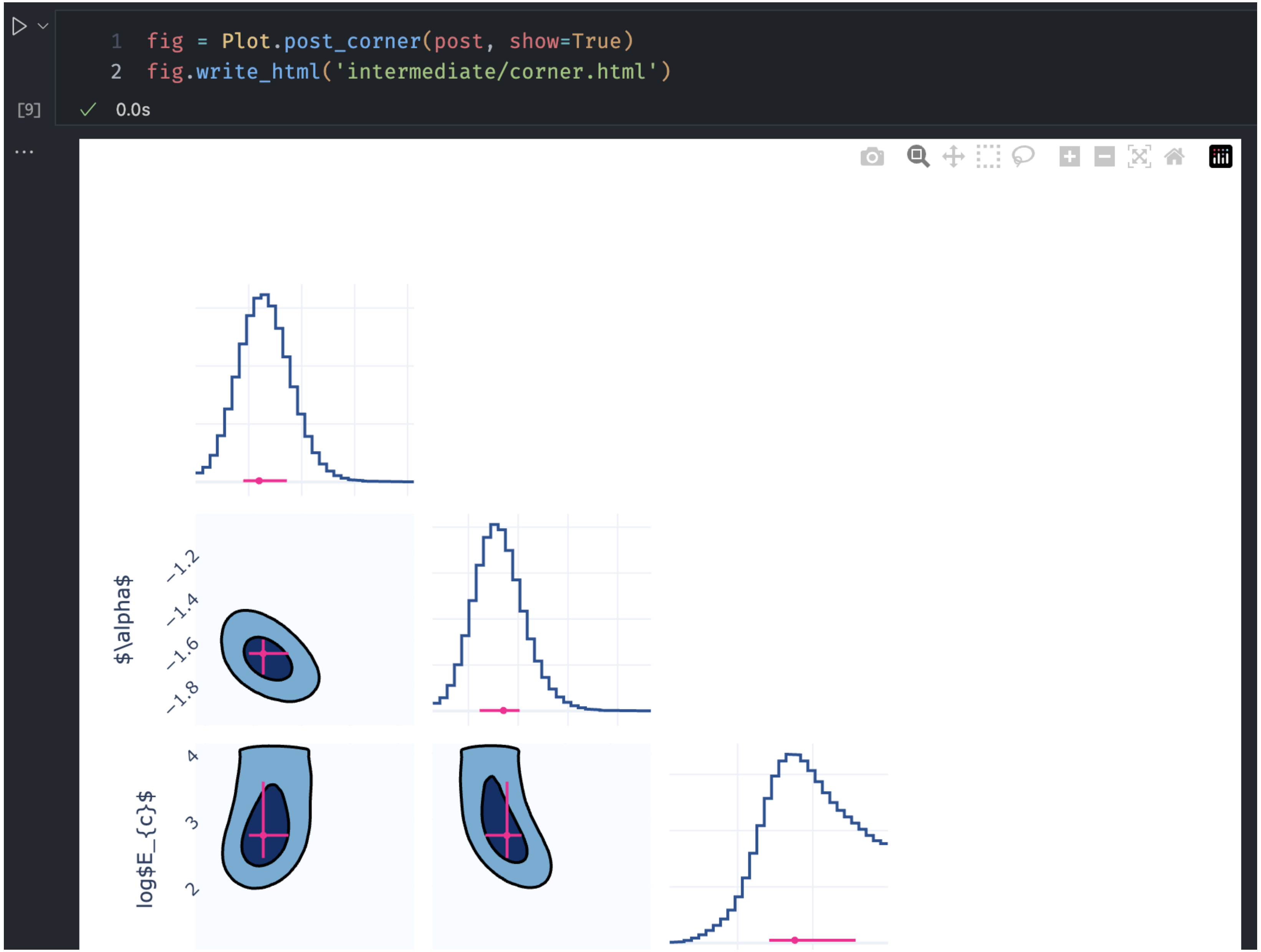
```
100%|██████████| 1000/1000 [00:25<00:00, 38.86it/s]
+------+--------------+-----------+------------+--------+--------+--------+------------------+
| par# | Expression   | Component | Parameter  | Mean   | Median | Best   | 1sigma CI        |
+------+--------------+-----------+------------+--------+--------+--------+------------------+
| 1    | (AS_TbAbs*cpl) | AS_TbAbs | NH         | 1.154  | 1.143  | 1.098  | [0.949, 1.359]   |
| 2    | (AS_TbAbs*cpl) | cpl      | $\alpha$   | -1.675 | -1.684 | -1.659 | [-1.755, -1.595] |
| 3    | (AS_TbAbs*cpl) | cpl      | log$E_{c}$ | 2.95   | 2.909  | 2.761  | [2.418, 3.570]   |
| 4    | (AS_TbAbs*cpl) | cpl      | log$A$     | -3.064 | -3.081 | -3.052 | [-3.151, -2.978] |
+------+--------------+-----------+------------+--------+--------+--------+------------------+
+-------+----------------+-----------+----------+------+
| Data  | Model          | Statistic | Value    | Bins |
+-------+----------------+-----------+----------+------+
| LE    | (AS_TbAbs*cpl) | cstat     | 16.53    | 14   |
| ME    | (AS_TbAbs*cpl) | pgstat    | 24.54    | 35   |
| Total | Total          | stat/dof  | 41.06/45 | 49   |
+-------+----------------+-----------+----------+------+
+-------+-------+-------+------+
| AIC   | AICc  | BIC   | lnZ  |
+-------+-------+-------+------+
| 49.06 | 49.97 | 56.63 | None |
+-------+-------+-------+------+
```

bayspec example: LE+ME (tbabs*cpl)

```
fig = Plot.infer(post, style='CE', show=True)
fig.write_html('intermediate/ctsspec.html')
```

```
fig = Plot.infer(post, style='NE', show=True)
fig.write_html('intermediate/phtspec.html')
```

bayspec example: LE+ME (tbabs*cpl)



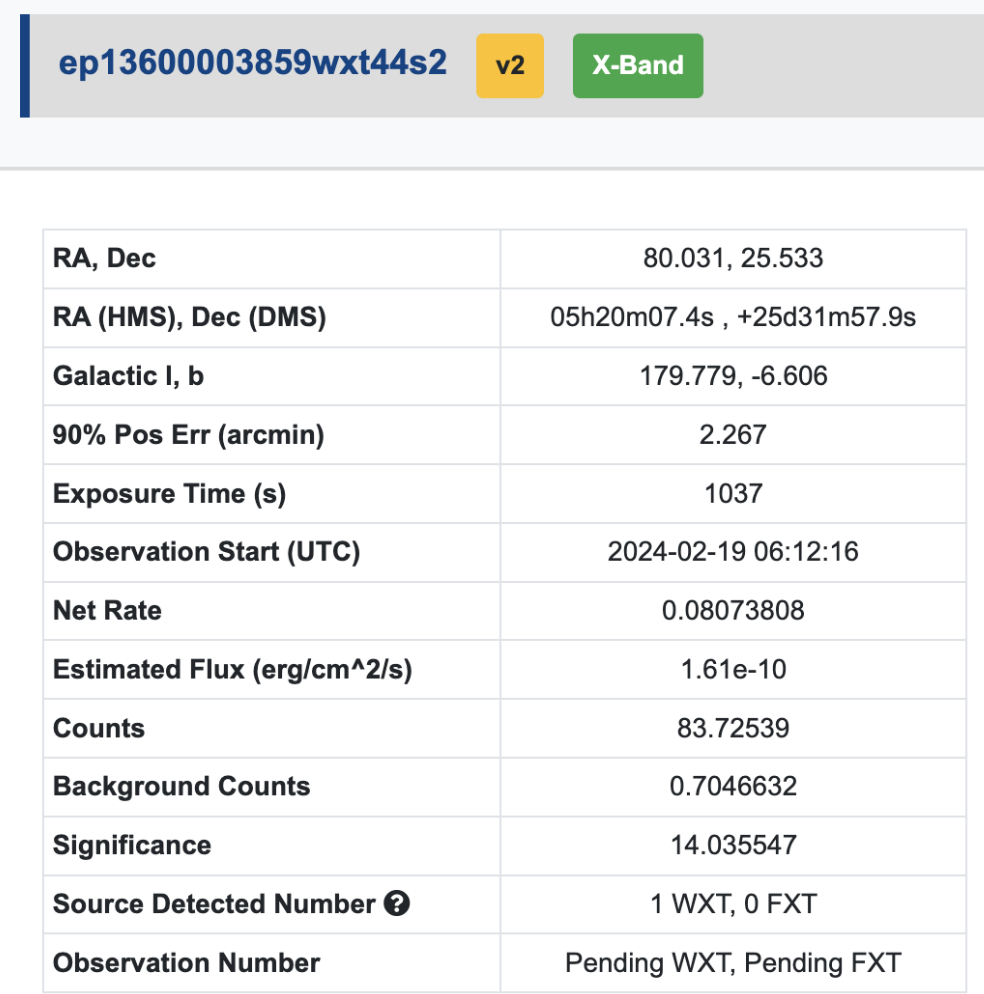
```
fig = Plot.post_corner(post, show=True)
fig.write_html('intermediate/corner.html')
```

bayspec example: LE+ME (tbabs*cpl)

```
earr = np.logspace(np.log10(0.5), 4, 100)

modelplot = Plot.model(ploter='plotly', style='vFv', CI=True)
fig = modelplot.add_model(model, E=earr, show=False)
fig = modelplot.add_model(cpl_ins, E=earr, show=True)
fig.write_html('intermediate/model.html')
```

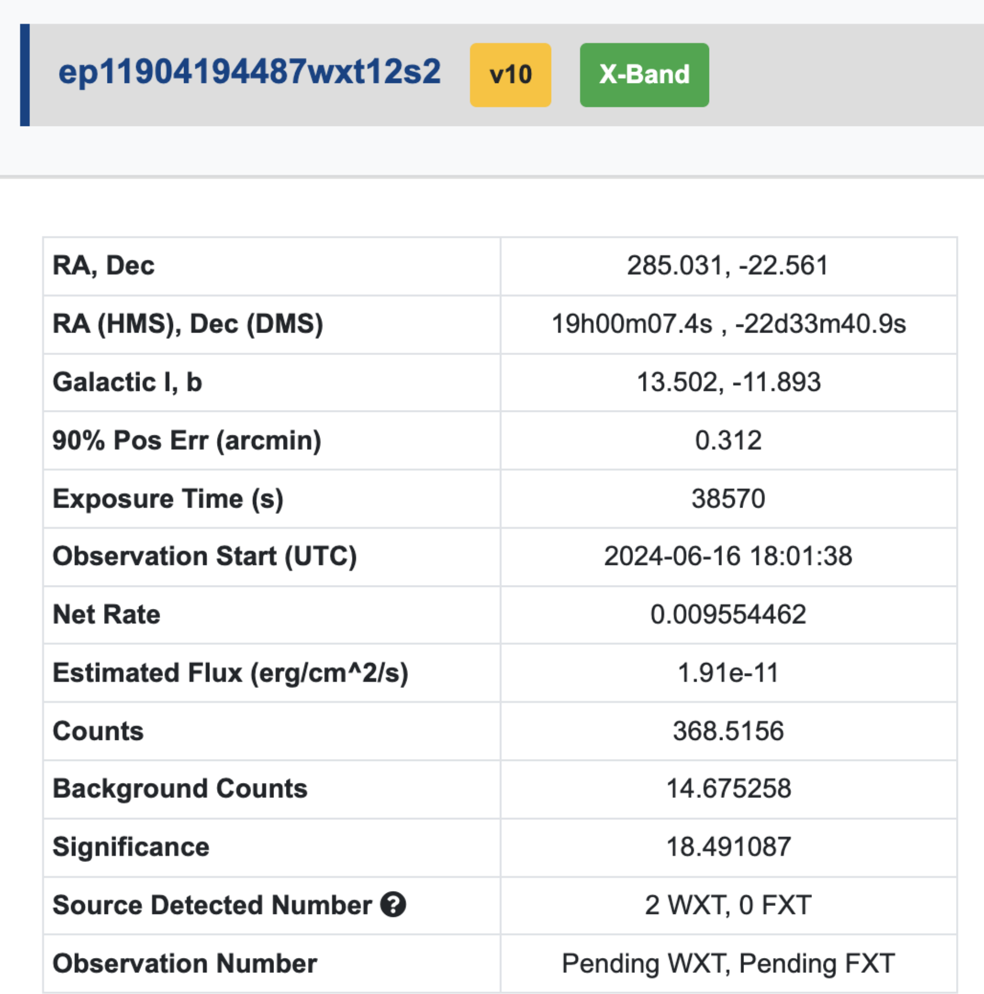
Part 5: exercises

针对以下两例EP探测到的快速X射线暂现源：
1，分析EP-WXT数据，生成X射线光变曲线和能谱
2，在GBM数据中检索其是否存在伽马射线对应体
3，若存在伽马射线对应体，则生成伽马射线光变曲线和能谱，并进行联合能谱拟合



ep13600003859wxt44s2 v2 X-Band

RA, Dec	80.031, 25.533
RA (HMS), Dec (DMS)	05h20m07.4s , +25d31m57.9s
Galactic l, b	179.779, -6.606
90% Pos Err (arcmin)	2.267
Exposure Time (s)	1037
Observation Start (UTC)	2024-02-19 06:12:16
Net Rate	0.08073808
Estimated Flux (erg/cm^2/s)	1.61e-10
Counts	83.72539
Background Counts	0.7046632
Significance	14.035547
Source Detected Number ?	1 WXT, 0 FXT
Observation Number	Pending WXT, Pending FXT



ep11904194487wxt12s2 v10 X-Band

RA, Dec	285.031, -22.561
RA (HMS), Dec (DMS)	19h00m07.4s , -22d33m40.9s
Galactic l, b	13.502, -11.893
90% Pos Err (arcmin)	0.312
Exposure Time (s)	38570
Observation Start (UTC)	2024-06-16 18:01:38
Net Rate	0.009554462
Estimated Flux (erg/cm^2/s)	1.91e-11
Counts	368.5156
Background Counts	14.675258
Significance	18.491087
Source Detected Number ?	2 WXT, 0 FXT
Observation Number	Pending WXT, Pending FXT

EP数据集下载链接：
https://pan.cstcloud.cn/s/bwSsnHrSTNY