Algorithmic strategies for GRB detection

HERMES Science Team

7 June 2021

Giuseppe Dilillo, Kim Ward

Goals and constraints

Goal:

- Characterize different trigger algorithms for GRB detection.
- Possibly identify good algorithms for operations onboard HERMES s/c.

Constraints:

- Algorithm should be fast (small OBC).
- Algorithm should be simple (many s/c).

Summary

1. Traditional approach to GRB detection.
2. Alternative approaches: FOCuS and the CUSUM framework.
3. New developments: Poisson FOCuS, background estimate through exponential smoothing techniques.

How to detect a GRB

4. BURST OBSERVATIONS

4.1. Triggers

A burst trigger occurs when the flight software detects an increase in the count rates of two or more NaI detectors above an adjustable threshold specified in units of the standard deviation of the background rate. The background rate is an average rate accumulated over the previous *T* seconds (nominally 17), excluding the most recent 4 s. Energy ranges are confined to combinations of the eight channels of the CTIME data. Trigger timescales may be defined as any multiple of 16 ms up to 8.192 s. Except for the 16 ms timescale, all triggers include two phases offset by half of the accumulation time. A total of 120 different triggers can be specified, each with a distinct threshold.

The trigger algorithms currently implemented include four energy ranges: the BATSE standard 50–300 keV range, 25–50 keV to increase sensitivity for SGRs and GRBs with soft spectra, >100 keV, and >300 keV to increase sensitivity for hard GRBs and TGFs. Ten timescales, from 16 ms to 8.192 s in steps of a factor 2, are implemented in the 50–300 keV range and the 25–50 keV range. The >100 keV trigger excludes the 8.192 s timescale, and the >300 keV trigger has only four timescales, from 16 ms to 128 ms.

The Fermi Gamma-Ray Burst Monitor, Meegan et al., Apj, 2009

SHORT RATE TRIGGERS

Running many short time scales through a triggering code can require most of the CPU time. Fortunately, the background counting rate of BAT is not expected to change on short time scales (i.e., less than a few seconds). Thus, for the short time scales we will use simple traditional triggers where there is a single background period of fixed duration before the foreground period. This is the type of trigger that was used on all GRB experiments from Vela to BATSE.

The short trigger looks for statistically significant increases in the count rate on five time scales: 4, 8, 16, 32, and 64 msec. This is done for nine different regions of the focal plane (four quadrants, the left half, right half, top half, bottom half, and the full focal plane) and for four energy ranges. Thus, there are 36 combinations of

The Trigger Algorithm for the Burst Alert Telescope on Swift, Fenimore et al., AIP Conf. Procs, 2003

Due to the nature of GRBs an onboard specific trigger criterion is needed. The GRBM trigger operates on the signals detected between the LLT and ULT. With a time resolution on 7.8125 ms a moving average is continuously computed on a Long Integration Time (LIT) that is adjustable between 8 and 128 s. The counts in a Short Integration Time (SIT, adjustable between 7.8125 ms and 4 s) are compared to the moving average, and if they exceed a $n\sigma$ level (where σ is the Poissonian standard deviation and n can be 4, 8 or 16) then the trigger condition is satisfied for that shield. If the same condition is simultaneously active for at least two shields, then the GRBM trigger condition is satisfied and the following time profiles (High Resolution Time Profiles) are stored, separately for each of the four shields:

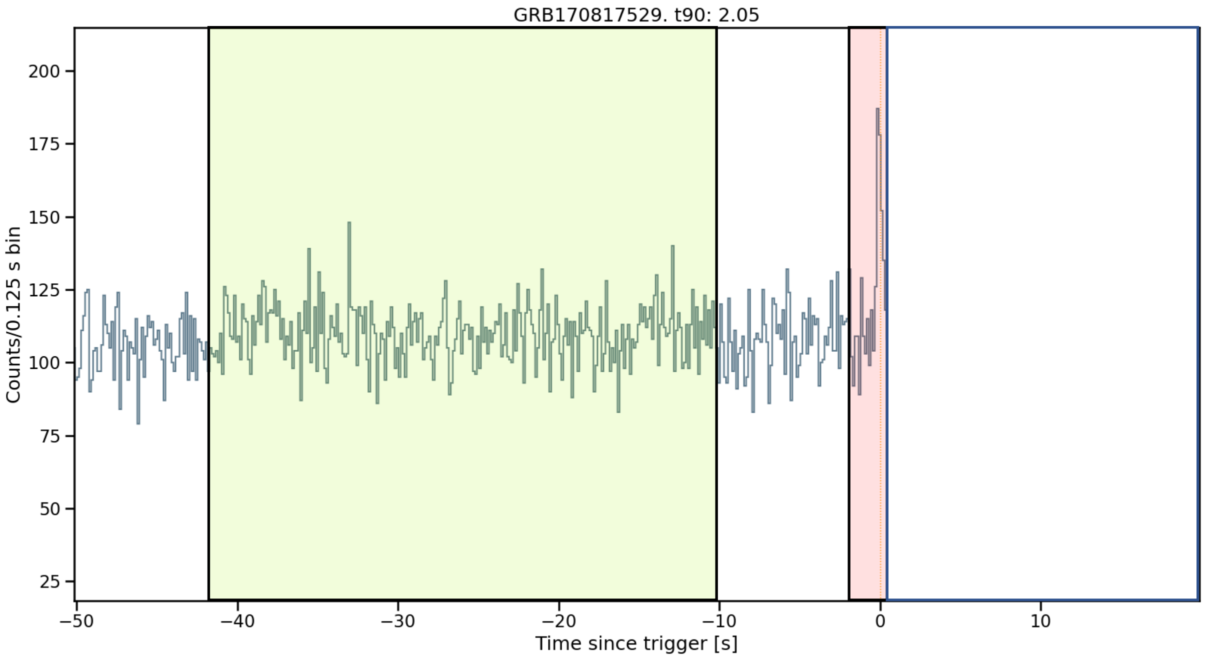
In-flight performances of the BeppoSAX Gamma-Ray Burst Monitor, Feroci et al., SPIE procs, 1997

The standard recipe

1. Count events in background window.
2. Count events in foreground window.
3. Somehow compute the significance of the foreground count excess.

$$S_t^h = S(x_{t-h+1}, \ldots, x_t | \mu_t)$$

4. If it exceeds a threshold value you issue a trigger, else wait for next count.



Count excess significance

The *true* approach to computing standard significance S:

$$p(\geqslant n) = \sum_{i=n}^{\infty} \frac{B^i\, e^{-B}}{i!} = 1 - \frac{\Gamma(n+1,\, B)}{\Gamma(n+1)}$$

$$\int_S^{\infty} N(x)\, dx = p(\geqslant n)$$

The background counts B is supposed to be the *true* background count.

Moreover, this computation is slow or requires tabulates of incomplete gamma and erf functions.

Useful with simulations and checks.

Count excess significance

$$S = \frac{(C_{i,k} - B_i 2^{k-10})^2}{C_{i,k} + \sigma^2_{\min}}\,. \qquad (1)$$

Here, $\sigma^2_{\min}$ is a commandable control variable to ensure that there is a minimum variance when the counts are small. A trigger is declared if S is greater than a threshold, $\sigma^2_{\text{threshold}}$. Each of the triggers for the 180 combinations are controlled by three commandable variables: an enable/disable, $\sigma^2_{\text{threshold}}$, and $\sigma^2_{\min}$.

The Trigger Algorithm for the Burst Alert Telescope on Swift,
Fenimore et al., 2003

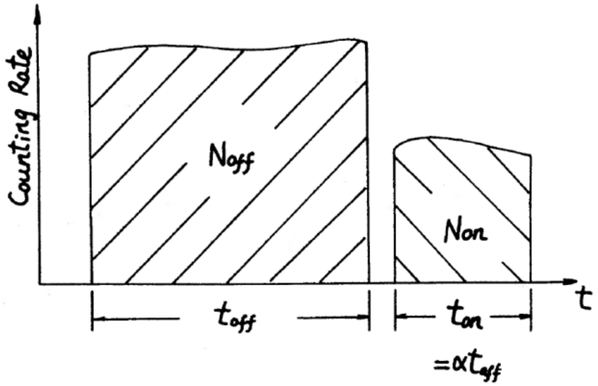
Common yet flawed approach:

$$S = \frac{n_i - b_i}{\sqrt{b_i}}$$

- It only holds its ground for large b_i i.e. large windows. **Very dangerously prone to false positives** otherwise.

Better approach (*Li-Ma, 1983*):

$$S = \sqrt{2}\left\{n \log\left[\frac{\alpha+1}{\alpha}\left(\frac{n}{n+b}\right)\right] + b \log\left[(\alpha+1)\frac{b}{n+b}\right]\right\}^{1/2}$$



from Lin-Ma, 1983

- Good on small windows, reasonable false positive rates.

But:

- Slow-ish (logs and sqrts).
- Does not account for trends.

Different triggers for different GRBs

Problem:

GRBs are a wild bunch.

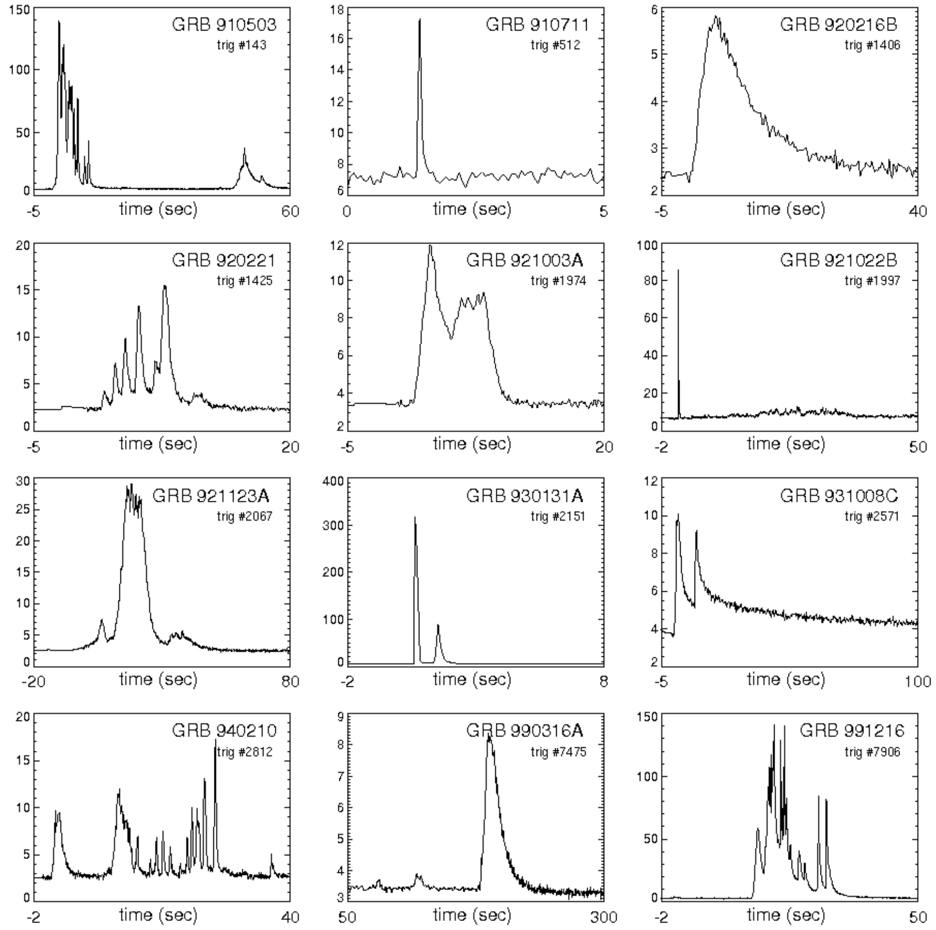
Solution :

Search foreground windows over **multiple timeframes** (and energy ranges).

combinations of the eight channels of the CTIME data. Trigger timescales may be defined as any multiple of 16 ms up to 8.192 s. Except for the 16 ms timescale, all triggers include two phases offset by half of the accumulation time. A total of 120 different triggers can be specified, each with a distinct threshold.

The trigger algorithms currently implemented include four energy ranges: the BATSE standard 50–300 keV range, 25–

The short trigger looks for statistically significant increases in the count rate on five time scales: 4, 8, 16, 32, and 64 msec. This is done for nine different regions of the focal plane (four quadrants, the left half, right half, top half, bottom half, and the full focal plane) and for four energy ranges. Thus, there are 36 combinations of



"No two gamma-ray bursts are the same, as can be seen from this sample of a dozen light curves. Some are short, some are long, some are weak, some are strong, some have more spikes, some have none, each unlike the other one."

All is well that end well?

Of course not.

SHORT RATE TRIGGERS

Running many short time scales through a triggering code can require most of the CPU time.

Problems:

1. This approach is **slow**.
2. You end up with a **lot of parameters**.

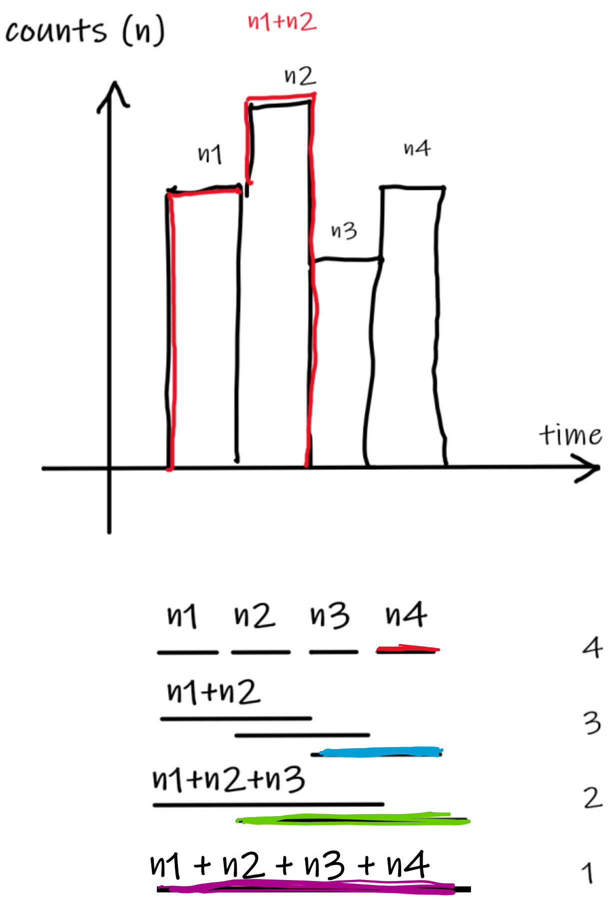
Abstract. The High Energy Transient Explorer uses a triggering algorithm for gamma-ray bursts that can achieve near the statistical limit by fitting to several background regions to remove trends. Dozens of trigger criteria run simultaneously covering time scales from 80 msec to 10.5 sec or longer. Each criteria is controlled by about 25 con-

BAT uses about 800 different criteria to detect GRBs, each defined by a large number of commandable parameters. Usually the critical parameter is the time scale of the sample being analyzed for a statistically significant increase. There are three triggering

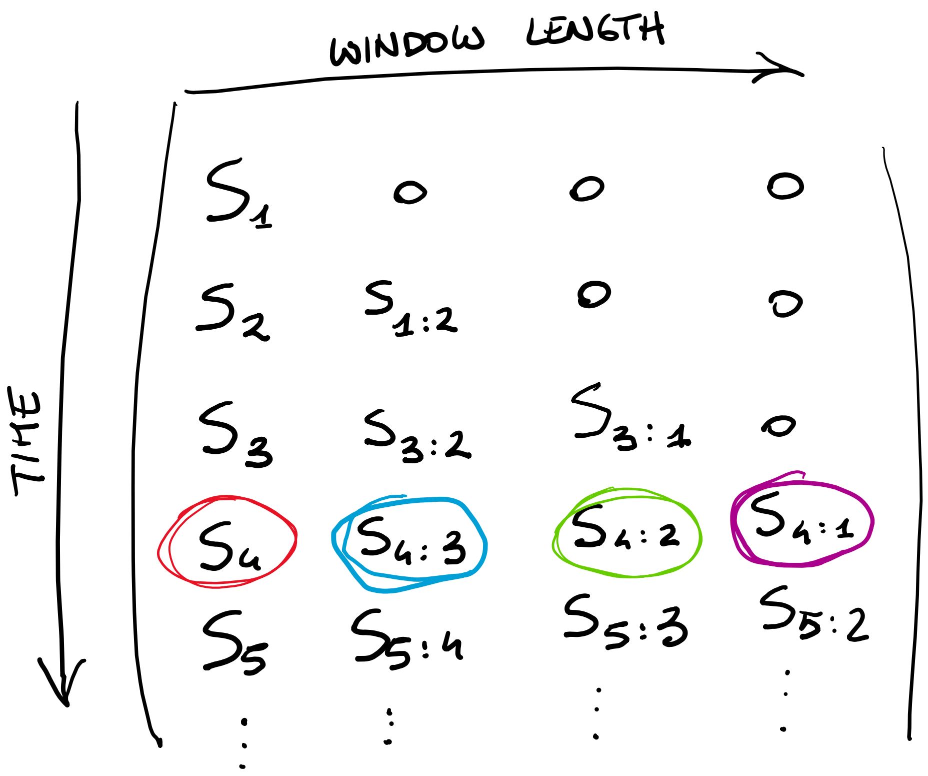
offset by half of the accumulation time. A total of 120 different triggers can be specified, each with a distinct threshold.

Both problems are aggravated in a nanosat context (small OBC, many s/c)

Representing trigger operations

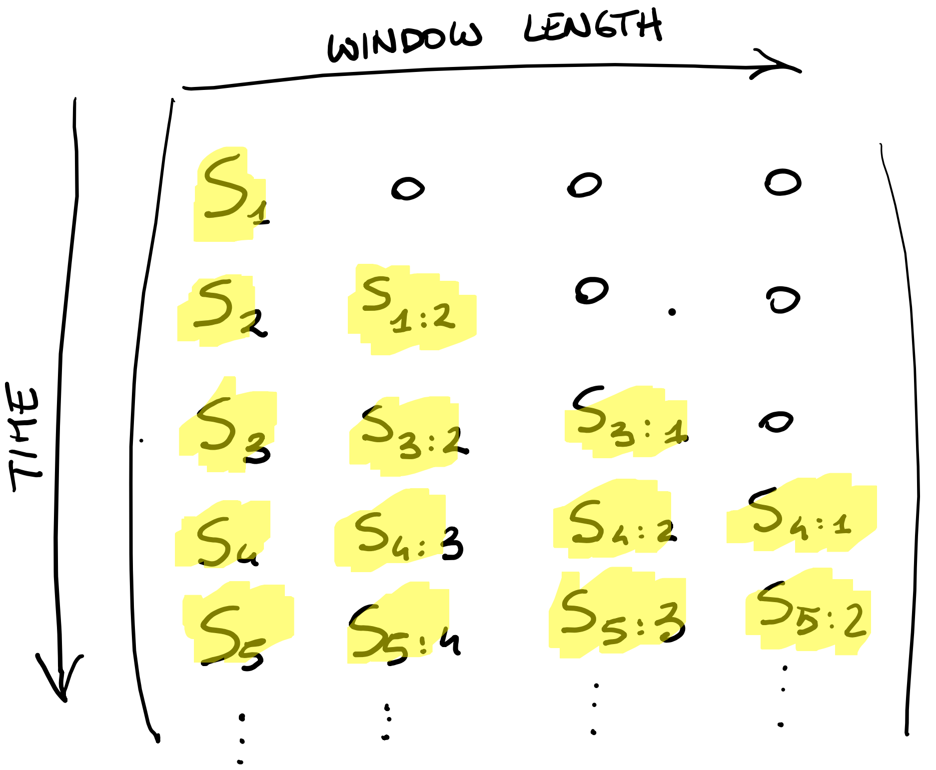


tot = N(N+1)/2 windows,
N new windows each time



FOR A FOREGROUND BUFFER WITH FOUR MEMORY UNITS

EXHAUSTIVE SEARCH :



FOR A FOREGROUND BUFFER WITH FOUR MEMORY UNITS
TRIGGER ALGORITHM COMPUTES ALL THE HIGHLIGHTED STATISTICAL SIGNIFICANCE

PARAMETRIC SEARCH:

$$
\begin{array}{c c}
 & \text{WINDOW LENGTH} \longrightarrow \\
\text{TIME} \big\downarrow &
\left(\begin{array}{cccc}
S_1 & 0 & 0 & 0 \\
S_2 & S_{1:2} & 0 & 0 \\
S_3 & S_{3:2} & S_{3:1} & 0 \\
S_4 & S_{4:3} & S_{4:2} & S_{4:1} \\
S_5 & S_{5:4} & S_{5:3} & S_{5:2} \\
\vdots & \vdots & \vdots & \vdots
\end{array}\right)
\end{array}
$$

FOR A FOREGROUND BUFFER WITH FOUR MEMORY UNITS

TRIGGER ALGORITHM COMPUTES ALL THE HIGHLIGHTED STATISTICAL SIGNIFICANCE

"INTERESTING SEARCH"

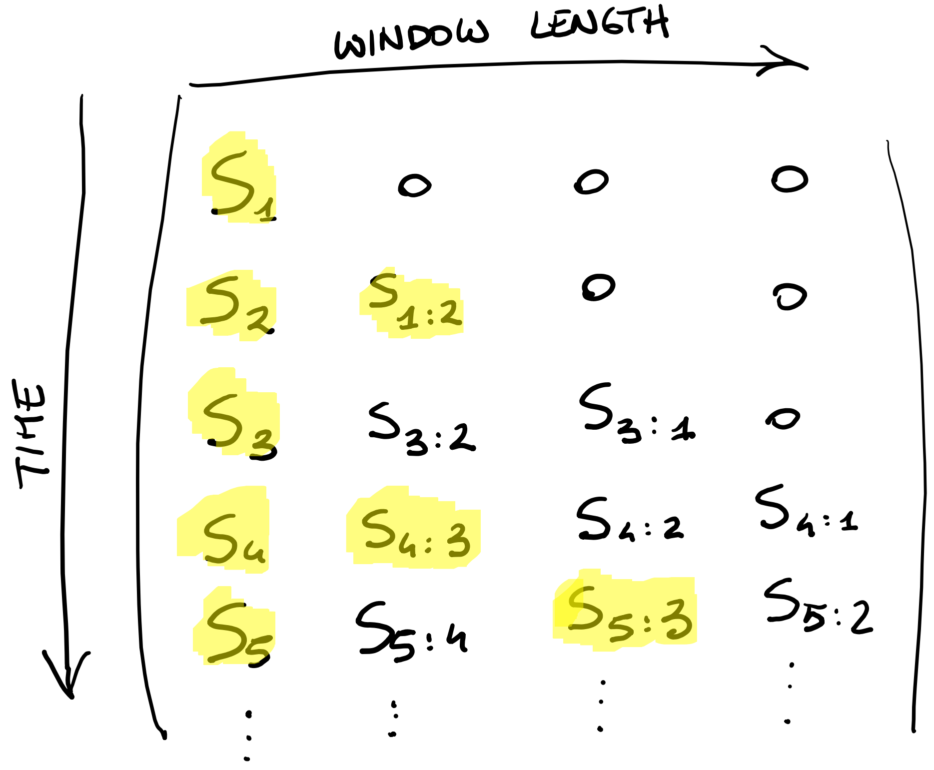
WINDOW LENGTH

TIME

$$
\begin{pmatrix}
S_1 & 0 & 0 & 0 \\
S_2 & S_{1:2} & 0 & 0 \\
S_3 & S_{3:2} & S_{3:1} & 0 \\
S_4 & S_{4:3} & S_{4:2} & S_{4:1} \\
S_5 & S_{5:4} & S_{5:3} & S_{5:2} \\
\vdots & \vdots & \vdots & \vdots
\end{pmatrix}
$$

FOR A FOREGROUND BUFFER WITH FOUR MEMORY UNITS TRIGGER ALGORITHM COMPUTES ALL THE HIGHLIGHTED STATISTICAL SIGNIFICANCE

"INTERESTING SEARCH"



FOR A FOREGROUND BUFFER WITH FOUR MEMORY UNITS

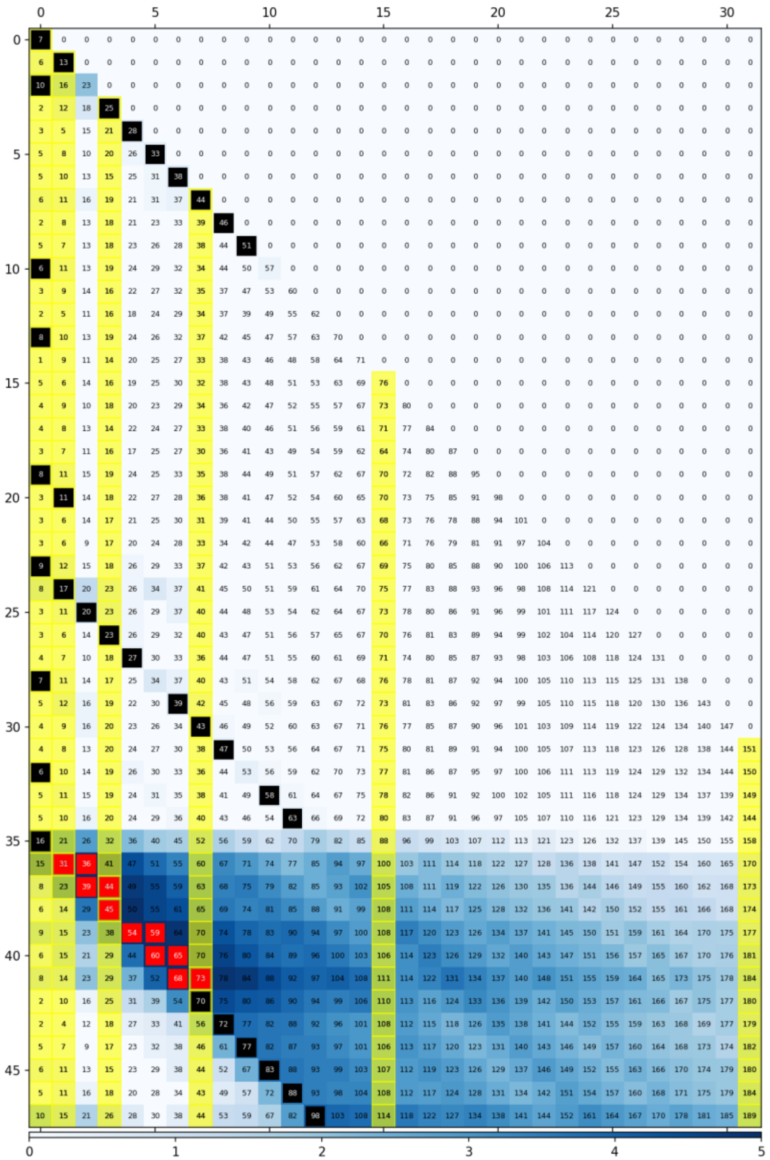
- EACH TIME A NEW COUNT IS AVAILABLE IT DECIDES IF THE DIAGONAL STEMMING FROM THAT COUNT IS A "GOOD ONE" AND SHOULD BE KEPT ACCORDING TO SOME CRITERIA.
- EACH TIME AN ELEMENT OVER A DIAGONAL IS COMPUTED IT DECIDES IF KEEP THAT DIAGONAL OR THROW IT AWAY
- LIKE EXHAUSTIVE SEARCH + FILTER.

TRIGGER ALGORITHM COMPUTES ALL THE HIGHLIGHTED STATISTICAL SIGNIFICANCE

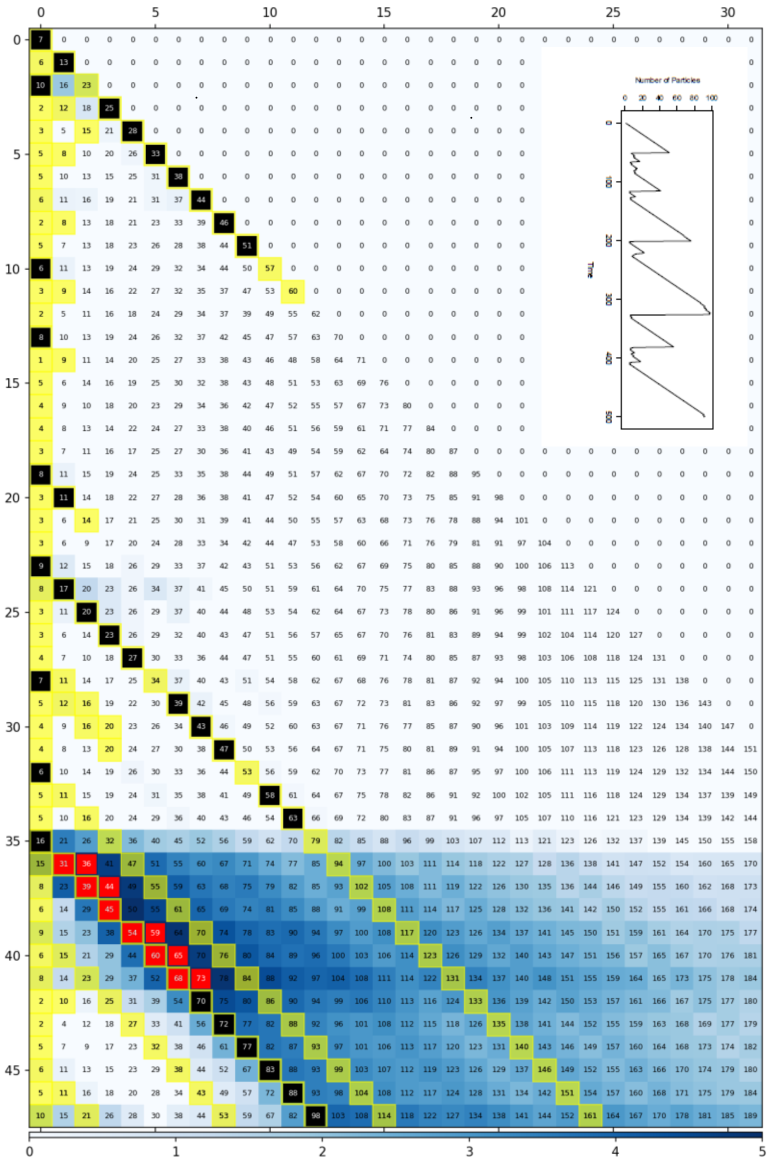
A RED SQUARE IS USED TO IDENTIFY SIGNAL WINDOWS WHERE S>5 ~ TRIGGER!

THE ALGORITHMS COMPUTE ONLY THE STATISTICAL SIGNIFICANCE OF SIGNAL WINDOWS HIGHLIGHTED IN YELLOW

AT EACH TIME A BLACK SQUARE IS PLOTTED IF A STATISTICAL SIGNIFICANT EXCESS WAS PRESENT AT SOME TIMESCALES, WHERE THE MAX EXCESS SIGNIFICANCE WAS REALIZED

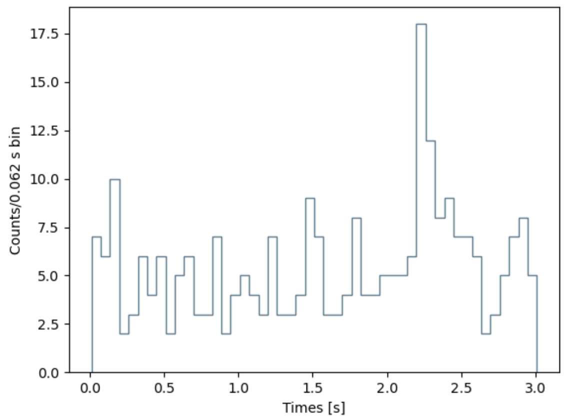


PARAMETRIC



INTERESTING

SHORT SIGNALS ARE EASIER TO REPRESENT



— NUMBERS REPRESENT COUNTS OVER A PARTICULAR SIGNAL WINDOW SO THEY INCREASE GOING LEFT TO RIGHT.

— BLUEISH COLORS REPRESENT SIGNIFICANCE.

— FOREGROUND OF 32 MEMORY UNITS.

— IF AN ALGORITHM ALWAYS CHECKS THE BLACK SQUARES IT IS GRANTED TO FIND A TRIGGER, IF ANY. THIS 'OFTEN' HAPPENS FOR INTERESTING ALGORITHM. I HAVE NO CLEAR EXPLANATION WHY

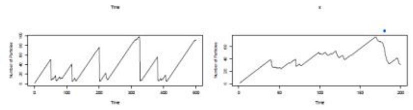


Figure 1: The Blocks data set (left-hand column) and Heaviside data set (right-hand column) together with results of analysis by the SRC algorithm with $\alpha = 10^{-6}$; data and inferred signal (top); marginal probability of changepoints (middle); and numbers of particles kept (bottom).

Where do we go from here?

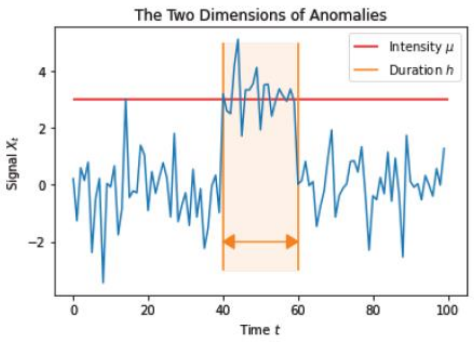
Winter 2019 – Initial discussion with Fiore and Riccardo.

Summer 2020 – Diagonal trigger algorithms work but no idea why.

September 2020 – Contact prof. Paul Fearnhead. Meet Paul, Idris and Kim.

Today – FOCuS.

The Lancaster group’s point of view

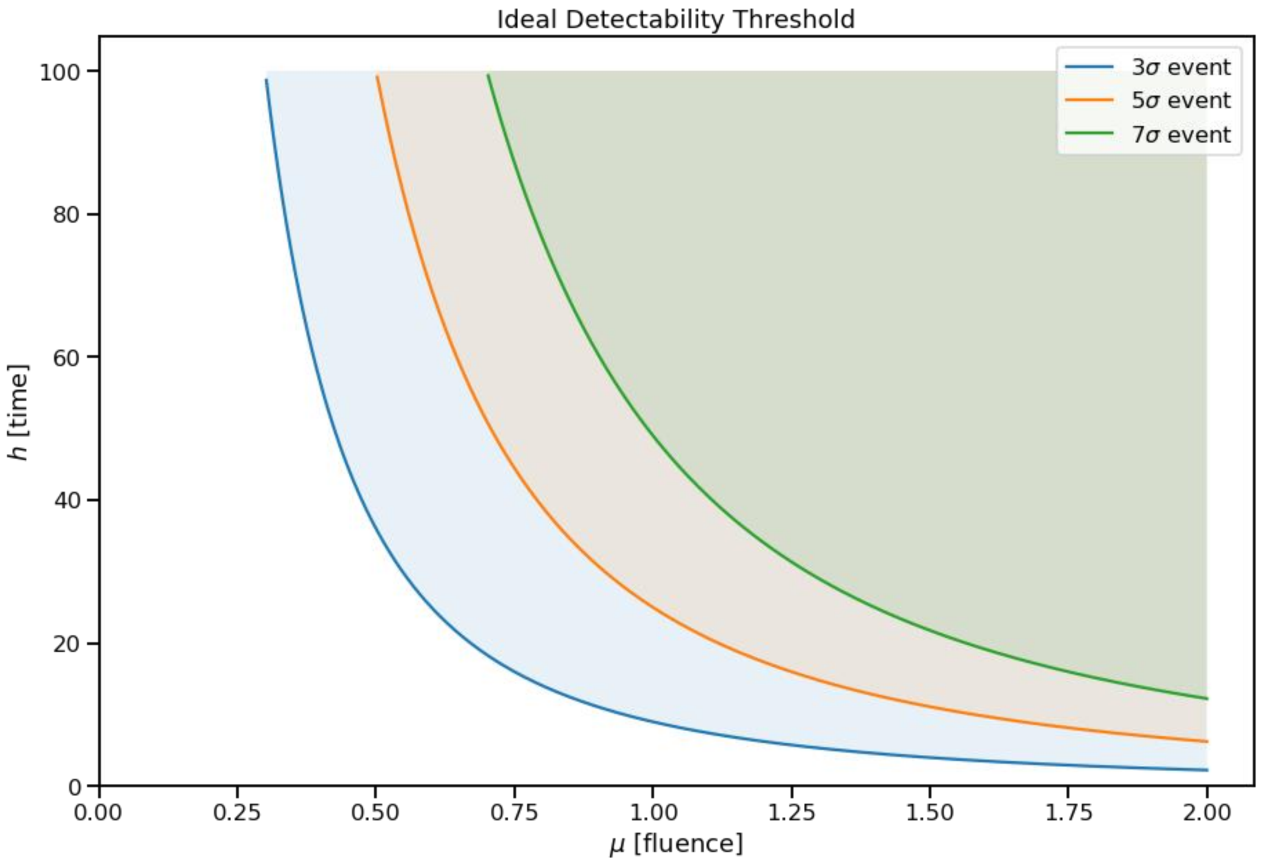


GRB observation domain.

Different GRBs occupy different places in this plot.

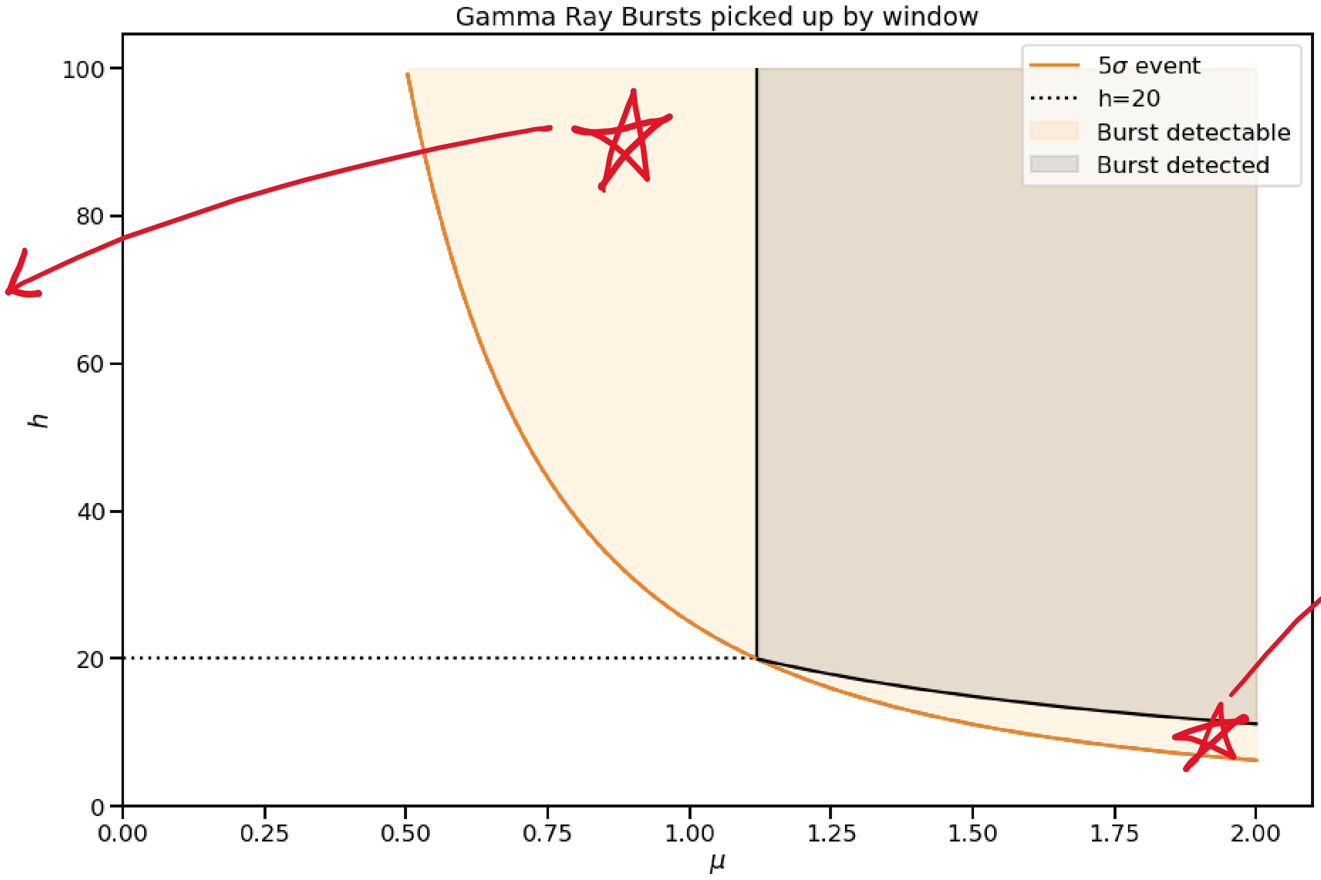
A GRB which is detectable above a certain significance threshold will overlap with shaded region.

The detection power of an algorithm can be quantified in terms of how good it **coverage** the shaded regions.



Single window strategy coverage

BURST IS LONG BUT NOT BRIGHT ENOUGH TO TRIGGER AT THIS TIMESCALE

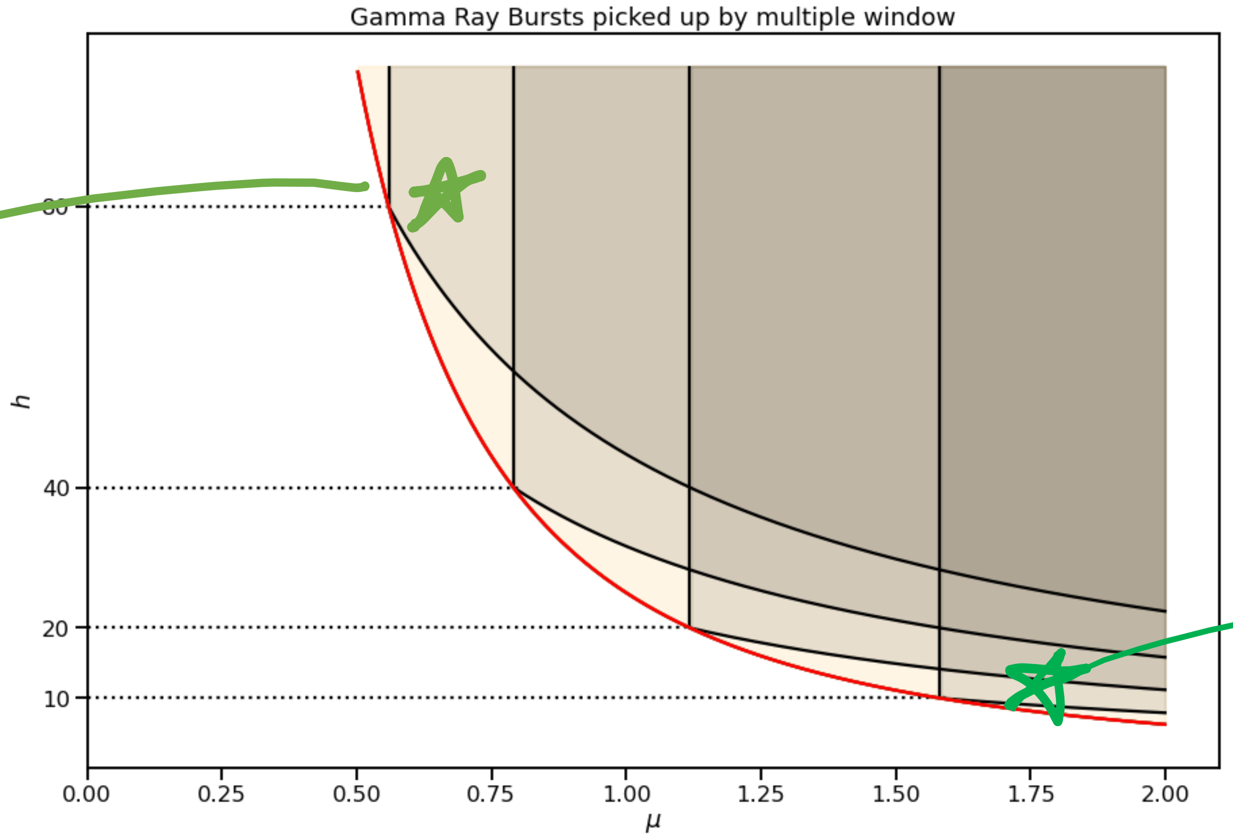


BURST IS BRIGHT BUT FOREGROUND WINDOW TOO LONG!

Baseline strategy coverage of detection domain is bad.

Traditional approach to detection

BURST DETECTED ON TIMESCALE $h = 80$



BURST DETECTED ON TIMESCALE $h = 10$

In order to achieve a better coverage one can pay computational resources in order to employ more foreground windows.

Alternative strategy: CUSUM

CUSUM (Cumulative Sum Control Chart) is a standard technique for anomaly detection. Initially developed for automating industrial quality control, L. Page in 1954.

Recipe: at each time a new count is available:

1. Estimate background rate.
2. Compute significance over background of the most recent count, y_t.
3. Recursively compute, for some μ:

$$S_t^{\mu} = \max\{0, S_{t-1} + 2\mu y_t - \mu^2\}$$

4. If $S_t^{\mu} > T^2$ issue a trigger else wait for next count.

What does the recursion mean

$$S_t^\mu = \max\{0, S_{t-1} + 2\mu y_t - \mu^2\}$$

This is a sequentially performed likelihood ratio (hypothesis) test. The alternative hypothesis being *is there any anomaly beginning at some time with mean intensity μ*?

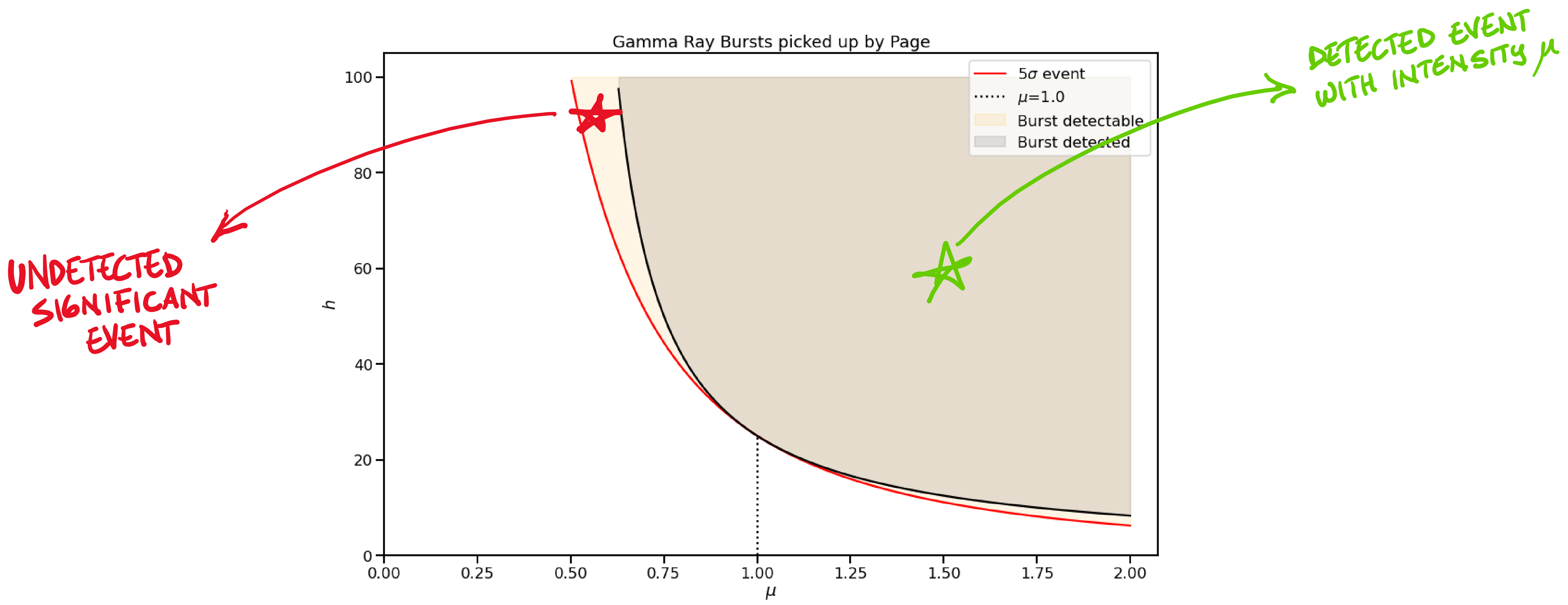
You can have three states:

1. $S_t = 0$. There are **no evidence** for anomaly.
2. $0 < S_t < T^2$. There are **some evidences** for a anomaly with intensity μ i.e. keep on acquiring.
3. $S_t > T^2$. There are **enough evidences** for a change with intensity μ within T sigma-significance level.

At times in which you go from 1 to 2 you have a new candidate **changepoint**.
At times in which you go from 2 to 3 you have a **trigger**.

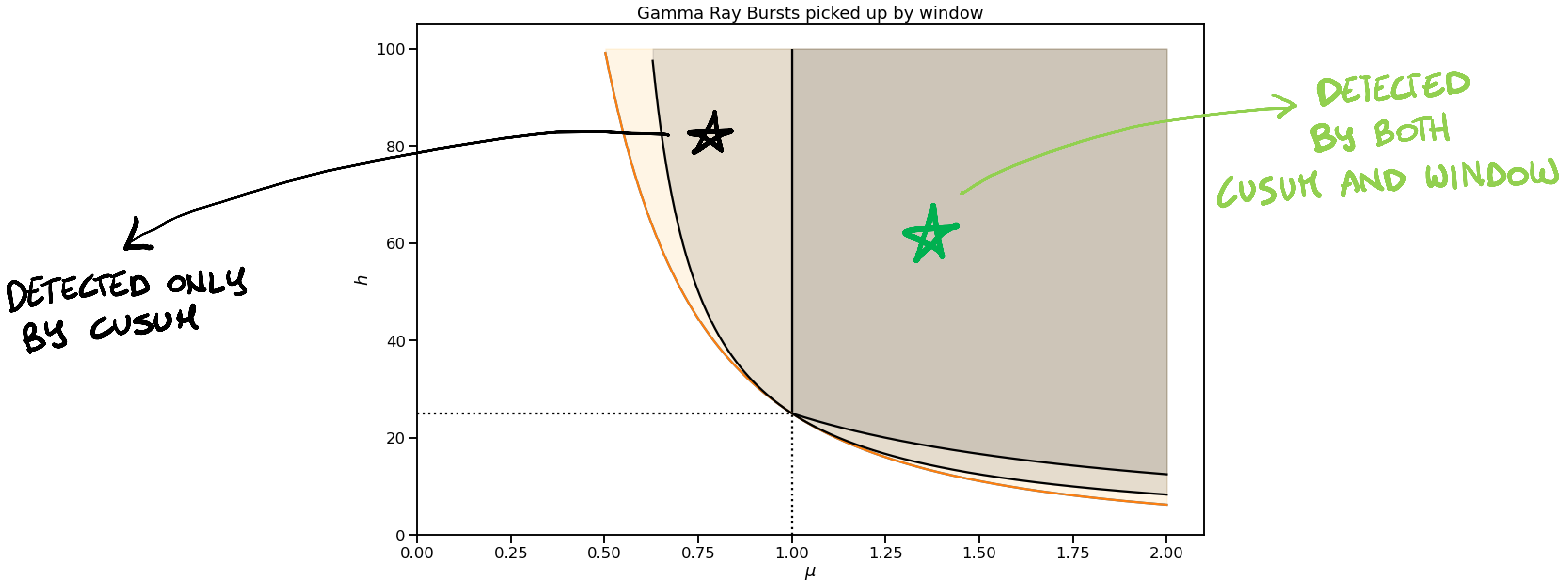
It's like diagonal method but with only one diagonal at time.

CUSUM coverage of detection domain



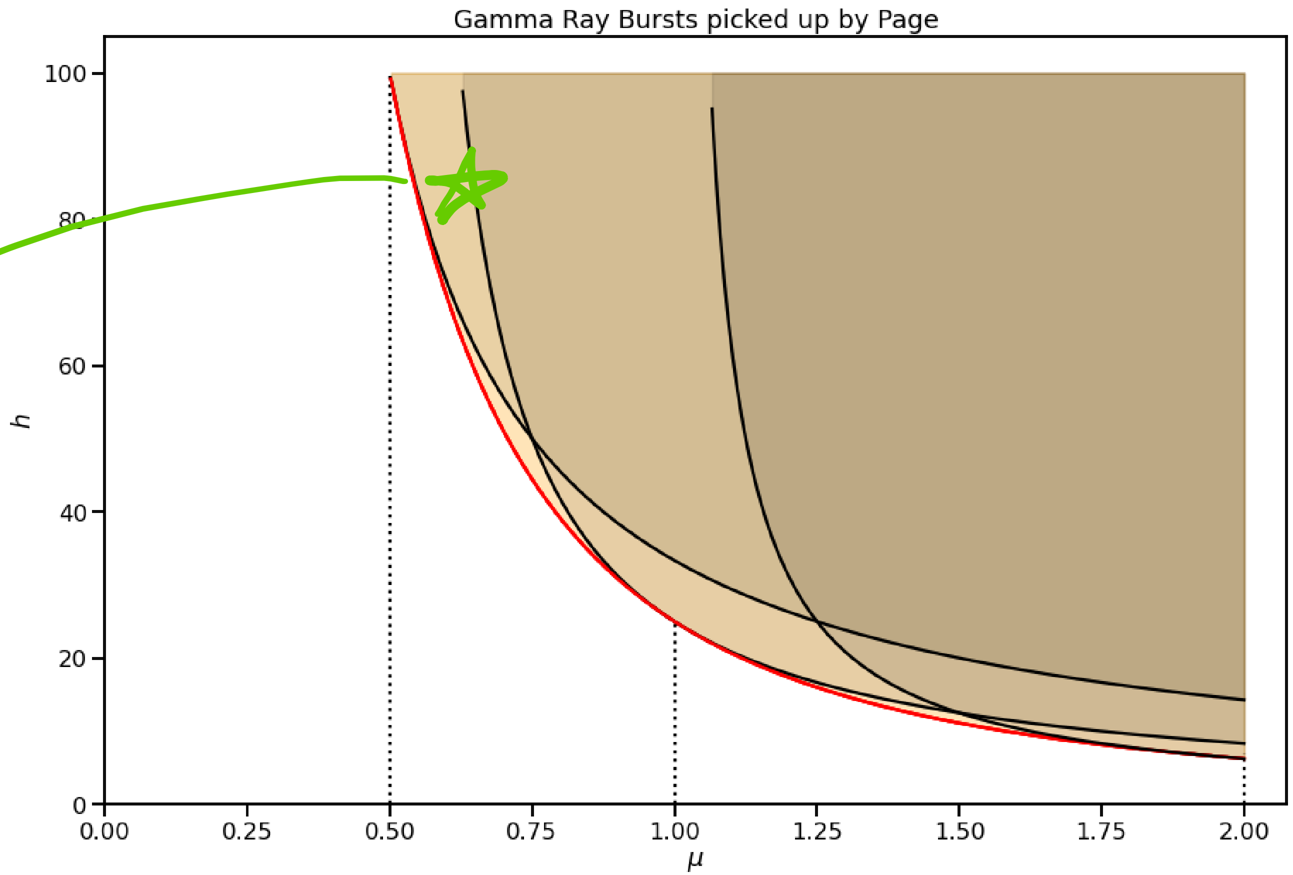
CUSUM strategies provide a better coverage of the observation domain

CUSUM coverage of detection domain



For a certain h there is an associated $\mu = T/\sqrt{h}$ for which CUSUM coverage is strictly better.

CUSUM coverage of detection domain



THIS TIME WE GOT IT!

Multiple CUSUM provide good coverage but do not completely cover the detection domain.

FOCuS

- Stands for Functional Online CUSUM.
- Developed by Kim Ward and Gaetano Romano under supervision of Paul Fearnhead and Idris Eckley of statistics department of University of Lancaster. Very recent, unpublished results.
- An improvement to CUSUM method which **computes** S_t^μ **for all possible $\boldsymbol{\mu}$ values** –all equivalent h window lengths.
- For the normal implementation, the idea is to solve in μ recursion like:

$$S_t^\mu = \max\{0, S_{t-1} + 2\mu y_t - \mu^2\}$$

The key fact making this possible: **solutions at a given time are piece-wise quadratics** which can be calculated and maximized efficienty.

Quadratics and pruning

1. The algorithm carries around a list of **quadratics** over which a few, simple operations are performed at each time step.
2. Statistical evidences for a **changepoint** at time t_0 is stored in a quadratic created at time t_0 and updated over time.
3. The technique to determine which quadratic carry around is called **pruning.**

   An unpruned FOCuS algorithm would add one quadratic each time you get a count greater than that expected from bkg. It will keep updating all the quadratics created in the past > soon computationally unfeasible.

   However, most of the quadratics generated this way would share most of the information content or carry a number of evidences which is null, small or overwhelmed by other quadratics.

Just FOCuS things

1. Dealing with changepoint when you get a trigger you get also a meaningful estimate of the **GRB start time**.
2. **Small memory usage** – you need two number to completely characterize a quadratic and most of the time you will be dealing with less than 5 quadratics at time.
3. Basic python implementation fits in **100 lines of code**.
4. No windows set-up means **less parameters**.
   Possibly very cool dealing with a constellation of satellites.

Tests - so far..

Three main metrics:

1. Detection power
2. Computation time
3. False positives

The following tests were performed for the normal version of the algorithm.

Extensive on simulated and simplified (one energy range) HERMES-like data.

Few on real data.

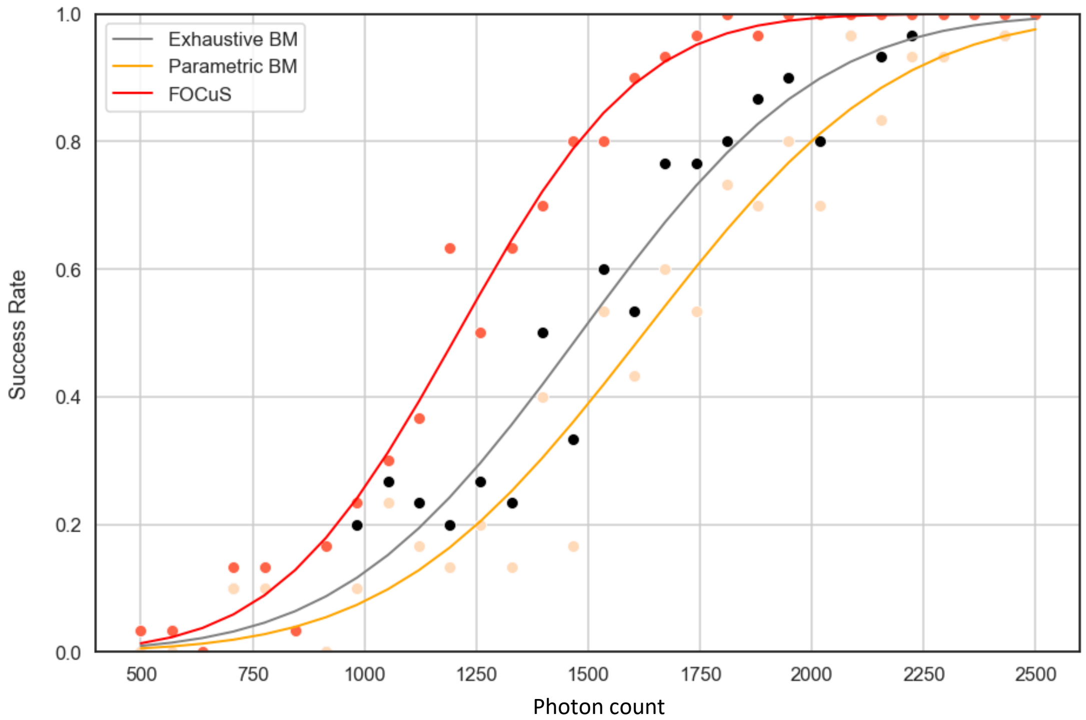
Detection power

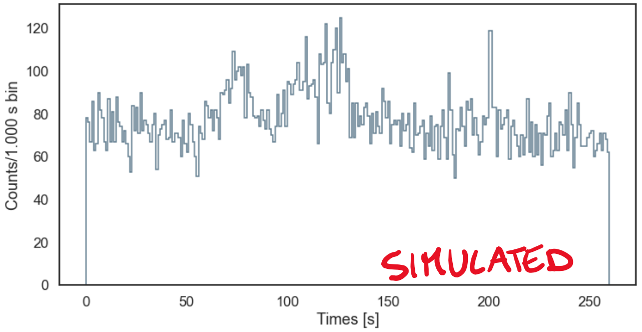
Long burst – GRB090323002

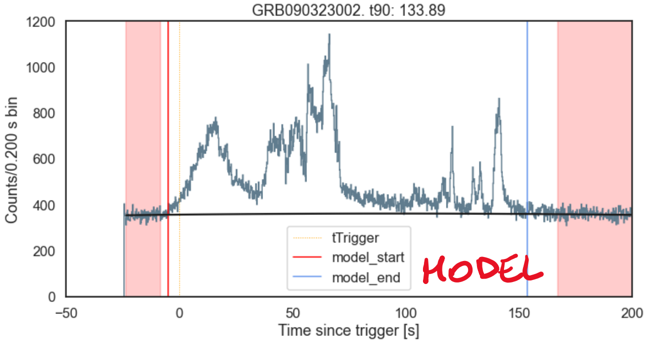
Notes:

1. Constant 70ph/s background, expected background for HERMES in band 50-300 keV.
2. Fundamental time bin 0.0625s.
3. Foreground duration 8 seconds.
4. Background window duration 32 seconds.
5. 30 simulations per transient fluence value.
6. Count significance as per Li-Ma, 1983.
7. Tests performed on 20 different bright, randomly selected GBM models.

In all tests FOCUS outperformed other algorithms.







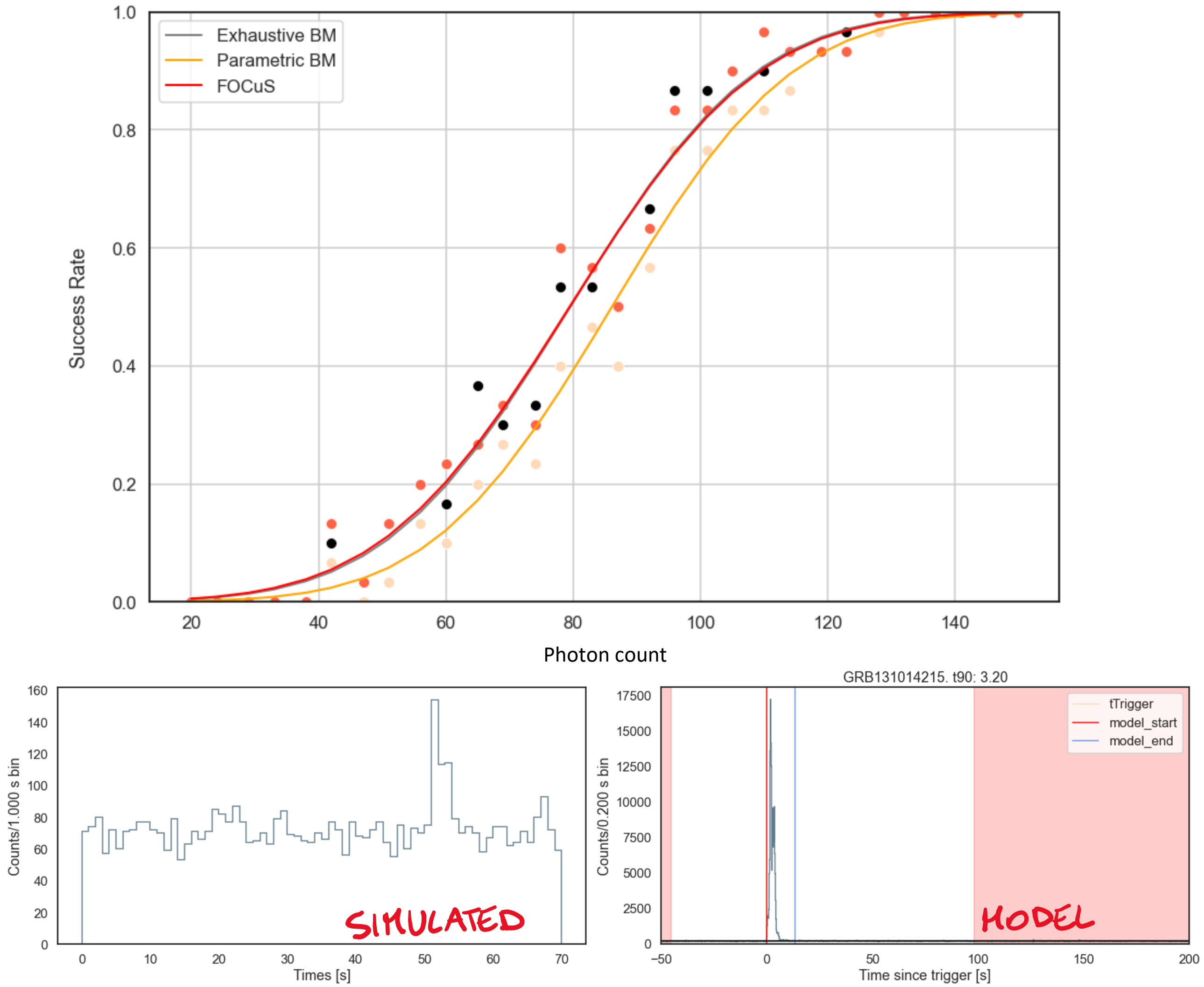
Detection power

Short burst – GRB131014215

Notes:

1. Constant 70ph/s background, expected background for HERMES in band 50-300 keV.
2. Fundamental time bin 0.0625s.
3. Foreground duration 8 seconds.
4. Background window duration 32 seconds.
5. 30 simulations per transient fluence value.
6. Count significance as per Lin-Ma, 1983.
7. Tests performed over 20 different bright, randomly selected GBM models.

Similiar detection power for exhaustive search and FOCuS.



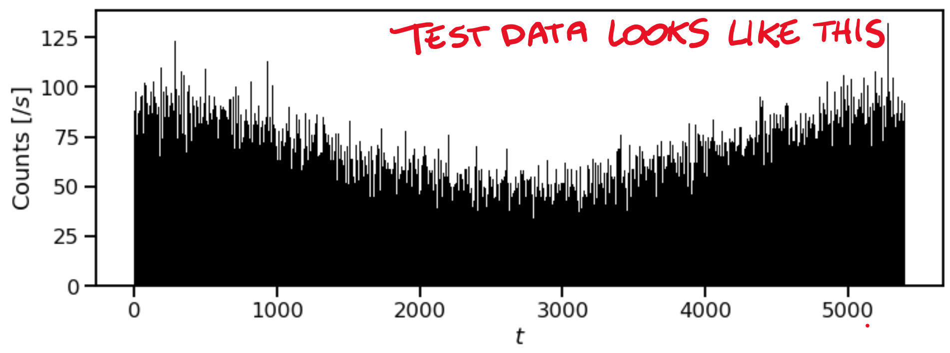
Computation times

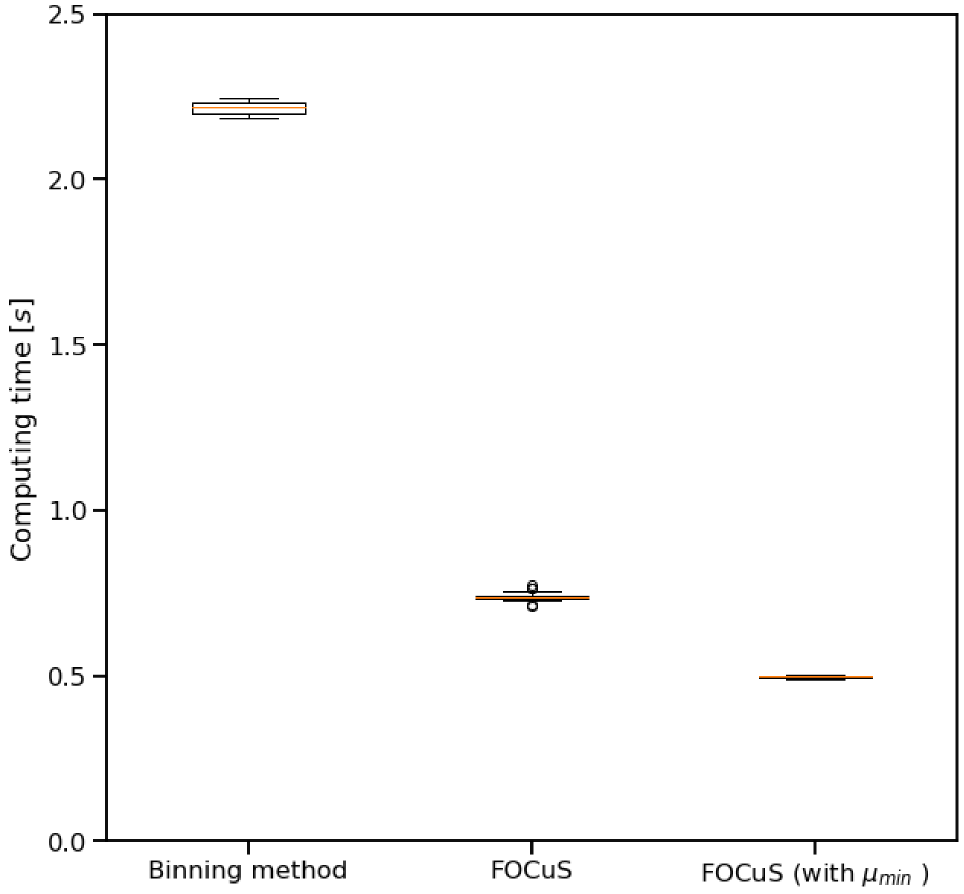
Different trigger algorithms are run over inhomogenous background simulations of fixed duration (5400 s). Duration of each trigger call is recorded.
The process is repeated for 100 background simulations.

Notes:

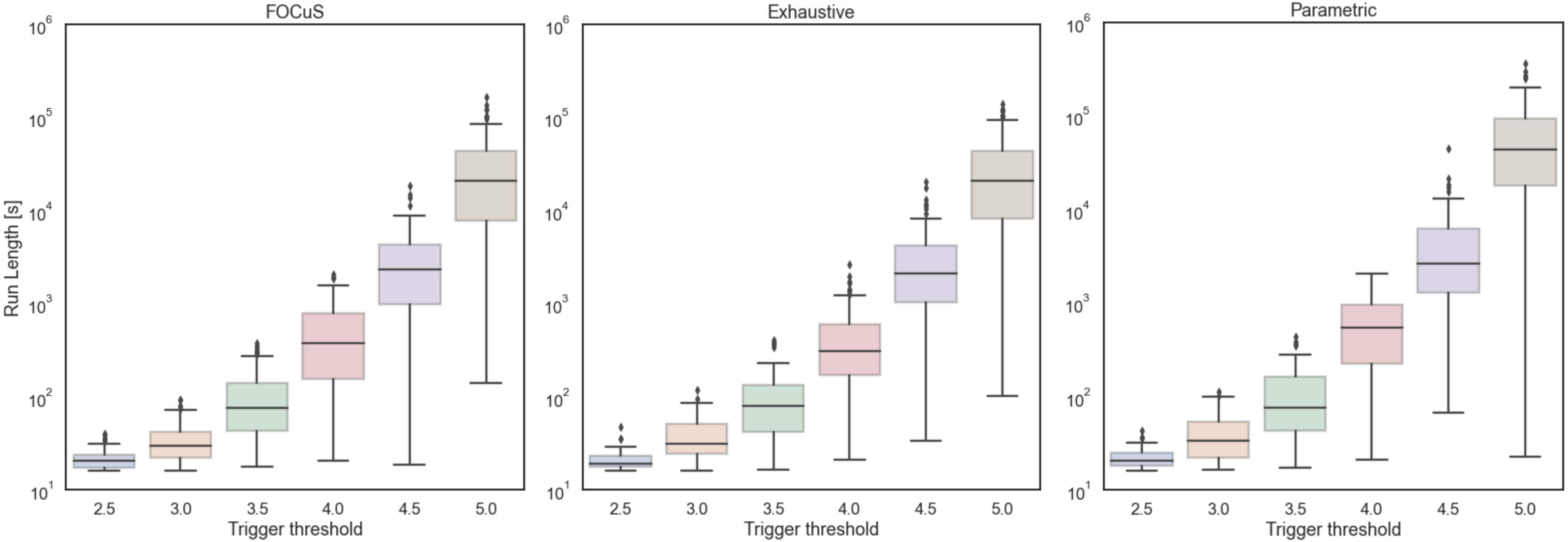
1. Background modulated by trigonometric function with period $5400\ s$.
2. Fundamental time bin 0.0625 s.
3. Foreground duration 8 seconds.
4. Background window duration 32 seconds.
5. Count significance as per Li-Ma 1983.

FOCUS +300 – 400% faster than log-spaced time lengths search.





False positives



Different trigger algorithms at different threshold are fed data from constant background until a positive is observed. The procedure is repeated 100 times. Count excess significance computed through Li-Ma, 1983. Fundamental time partitioning 0.0625s.

Parametric search is the algorithm less prone to false positives (less tests).

False positives
The impact of how you estimate significance

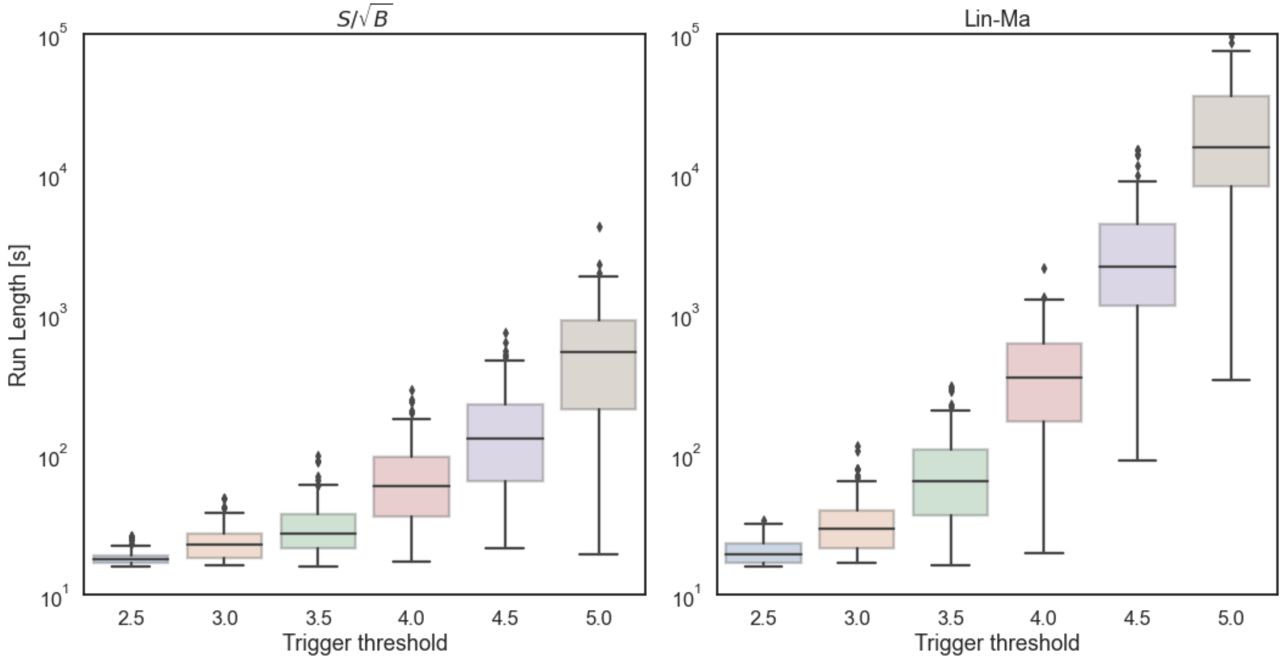
Because of the use of Poisson statistics in the definition of the trigger, it is a common misconception that trigger algorithms are guarding against statistical fluctuations. The σ level is usually never set below ~ 11 and, yet, there are still many false triggers. Obviously, the cause of false triggers is not statistical fluctuations. In most experiments (e.g., PVO, Ginga, ISEE-3), the σ level was 11 and 90% of the triggers were, in fact, not GRBs. BATSE also had a threshold equivalent to $\sim 11\sigma$; it used 5.5 σ in the second brightest illuminated detector which translates to $\sim 11\sigma$ in the brightest illuminated detector. BATSE achieved $\sim 50\%$ false trigger rate because many triggers could be rejected on-board by crude locating which was able to nullify many false triggers when the source appeared to be inside the satellite (i.e., particle events) or coming from the sun.

The HETE Triggering Algorithm – Fenimore et al.

Same trigger algorithm at different threshold, employing different strategies for estimating significance in counts.
Algorithms are fed data from a constant poisson background until a positive is detected. Operation is repeated 100 times per threshold value.

Notes:

1. We used exhaustive (all window) logic.
2. Fundamental partitioning 0.0625 s.
3. Foreground duration 8 seconds.
4. Background window duration 32 seconds.



Run lengths for $\frac{N-B}{\sqrt{N}}$ are dangerously small. Li-Ma is good.

The latests: FOCuS Poisson

The FOCuS implementation we have seen so far was built for normally distributed data.

As such, it required some technique to standardize data before input.

We do have a new FOCuS implementation natively built for Poisson distributed data (counts)

NORMAL

$$\lambda + \mu$$

$$S_t^{\mu} = \max\{0, S_{t-1} + 2\mu y_t - \mu^2\}$$

$$q_\tau(\mu) = a_\tau \mu^2 + b_\tau \mu$$

```
focus_step(last_count_significance)
```

POISSON

$$\lambda\mu$$

$$S_t^{\mu} = \max\left\{0, S_{t-1}^{\mu} + X_t \log\mu - \lambda(\mu - 1)\right\}$$

$$C_\tau(\mu) = a_\tau \log\mu + b_\tau(\mu - 1)$$

```
focus_step(last_count, expected_background_count)
```

Exponential smoothing

Different strategies for computing background e.g. model query, moving average..

We are proposing using exponential smoothing techniques which are **faster**, i.e.:

Simple exponential smoothing:

Simplest (one parameter).

$$s_t = \alpha x_t + (1 - \alpha) s_{t-1}$$

or

Double exponential smoothing:

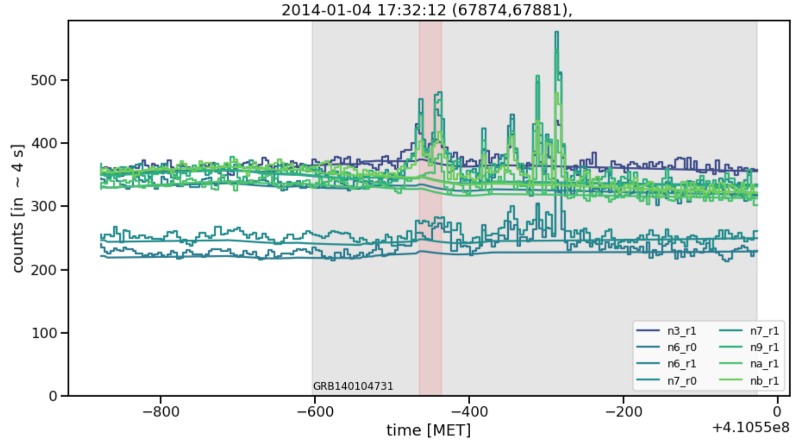
Two parameters, good with trend.

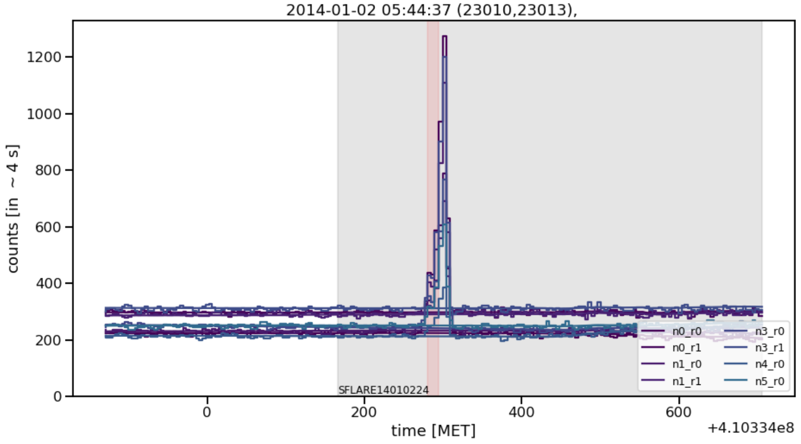
$$s_t = \alpha x_t + (1 - \alpha)(s_{t-1} + b_{t-1})$$
$$b_t = \beta(s_t - s_{t-1}) + (1 - \beta) b_{t-1}$$

FOCuS x NN

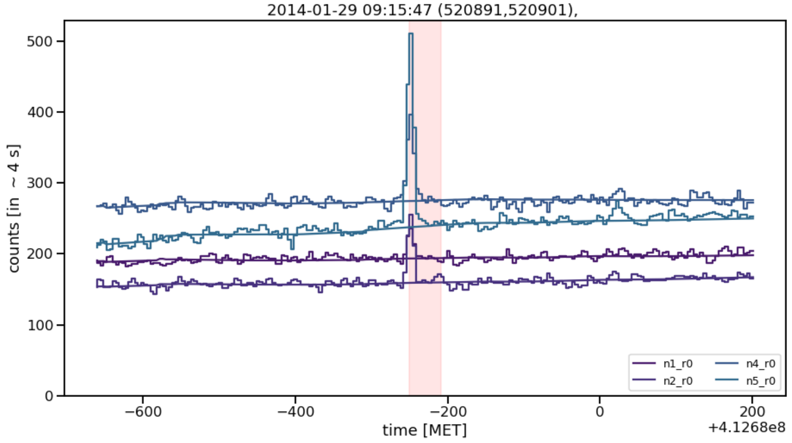
We ran FOCuS on 2 months of GBM data with bkg estimated through a neural network developed by Riccardo Crupi.

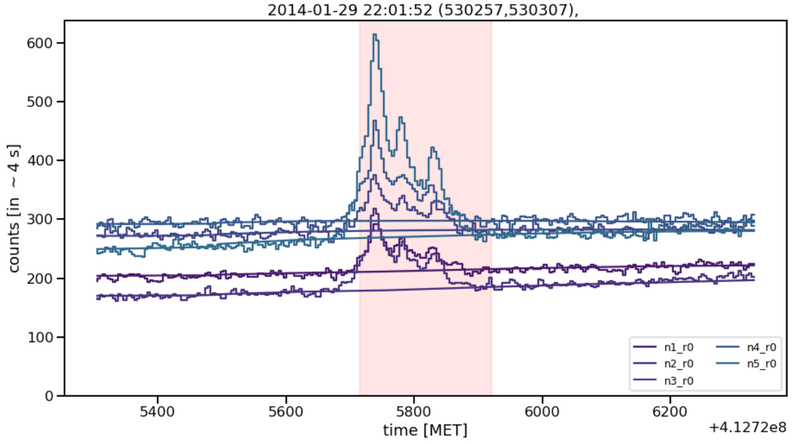
IN CATALOG

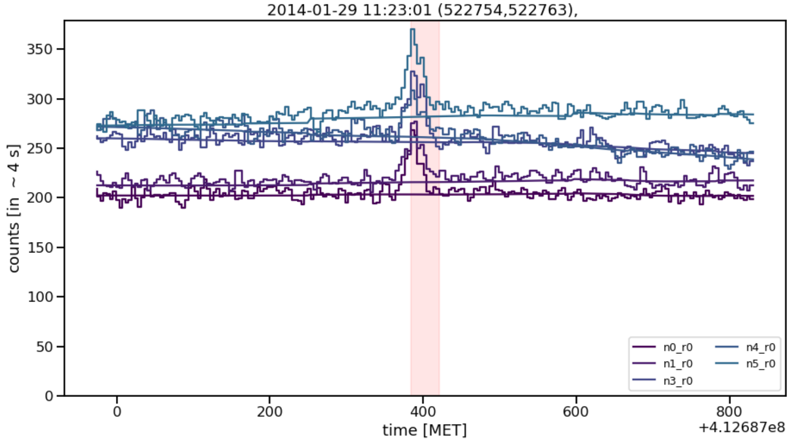


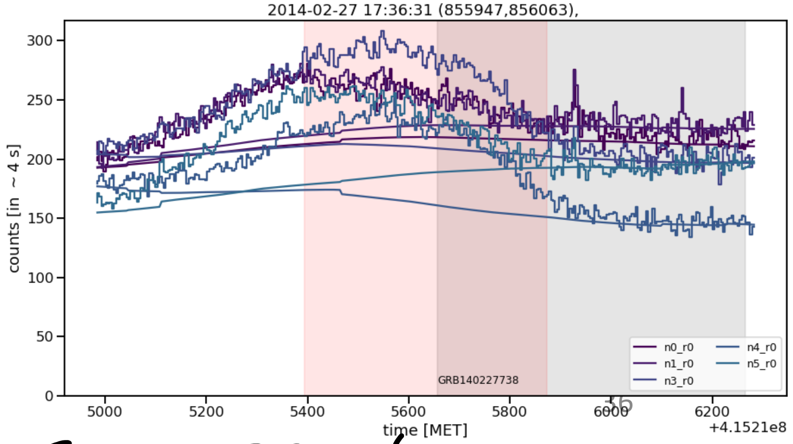


NOT IN CATALOG?









A LOT OF THESE!

Conclusions

- With traditional algorithms there are 1. a trade-off between computational and detection efficiency; 2. possibly messy interfaces. These problems worsen in nanosats context.
- How to estimate background rate and significance is critical.
- FOCuS is a much performant alternative to traditional trigger algorithms.
- More realistic data are needed for further testing and optimization.