Algorithms for GRB detection

Giuseppe Dilillo (postdoc, INAF-IAPS)
Kester Ward, Idris Eckley, Paul Fearnhead, Riccardo Crupi, Andrea Vacchi, Fabrizio Fiore

V Congresso Nazionale GRB,
15-09-2022

Summary

1. Conventional algorithms for detecting GRBs.
2. A new algorithm: FOCuS-Poisson
3. Testing different algorithms

How Gamma-Ray Bursts are detected

The logic of count rate trigger algorithms is:

1. High-energy photons are counted as they reach the detector.
2. An average rate for the background is determined.
3. Observed photon counts are integrated over a timescale and compared against the background guess.
4. A trigger is issued if the significance of the excess in observed counts overcome a threshold.

5. Repeat step 4. for different timescales.

pg. 667). Both the size and locations of the intervals over which the signal is averaged affect the result, and therefore one must consider many different values of the corresponding parameters. The idea is to minimize the chances of missing a signal because, for example, its duration is poorly matched to the interval size chosen. If the background is determined dynamically, by

[2]: Studies in astronomical time series analysis. VI. Bayesian Blocks representations – Scargle, Jackson et al.
http://doi.org/10.1088/0004-637X/764/2/167

6. Repeat steps 1-5 for different energy bands.

Problems of conventional trigger algorithms

Count-rate trigger algorithms are not evil.
They are robust and well understood.

Yet, they:

1. Are biased.
2. Have many parameters.
3. Are inefficient.

BAT uses about 800 different criteria to detect GRBs.

Setting the Triggering Thresholds on Swift, McLean et al..
https://doi.org/10.1063/1.1810931

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

[2]: THE FERMI GAMMA-RAY BURST MONITOR, Meegan et al.
https://doi.org/10.1088/0004-637X/702/1/791

One goal is to explore the widest possible parameter space. As such, as many triggers as possible will be run simultaneously until the flight computer is nearly saturated. Thus, special attention will be paid to the CPU usage.

[1]: The Trigger Algorithm for the Burst Alert Telescope on Swift,Fenimore et al.
https://doi.org/10.1063/1.1579409

A new algorithm for GRB detection

The idea: let's make an algorithm which does not test for *many* timescales.. Let's make an algorithm that tests over *all* timescales!

The biggest constraint: we want the algorithm to be fast.

How?

1. **Equipping the algorithm with a memory state and of the math necessary to evaluate evidences of a burst.**
2. **Having the algorithm work only when evidences are actually there.**

CUSUM, the math behind FOCuS

The math behind FOCuS is based over CUSUM, a well-established changepoint detection techniques.

If traditional algorithm looks for transients based on their duration, CUSUM-based algorithms looks for transients based on their intensity.

CUSUM search involves sequentially computing recursions like:

$$\zeta_{i,\mu} = \max\left(0, \zeta_{i-1,\mu} + x_i \log \mu - \lambda_i(\mu - 1)\right)$$

where μ is an intensity parameter

What does the recursion mean

$$\zeta_{i,\mu} = \max\left(0, \zeta_{i-1,\mu} + x_i \log \mu - \lambda_i(\mu - 1)\right)$$

Formally this is a sequentially performed likelihood test.
You can have three states:

1. $\mathrm{Z_t} = 0$. There are **no evidence** for significant excess in count.
2. $0 < \mathrm{Z_t} < \mathrm{T}^2$. There are **some evidences** for a change with post-change mean intensity μ i.e. keep acquiring.
3. $\mathrm{Z_t} > \mathrm{T}^2$. 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**.

FOCuS - Functional Online CUSUM

- Original implementation for normally distributed data by Romano G., Fearnhead P. et al. Collaborated with Ward K., and Fearnhead P. to develop the Poisson version of the algorithm.
- An improvement to CUSUM method which computes the CUSUM test statistic for all possible post-change mean values μ – **equivalent to testing over all possible durations.**
- The idea is to solve in the post-change mean μ the CUSUM recursion:

$$\zeta_i(\mu) = \max\left(0, \zeta_{i-1,\mu} + x_i \log \mu - \lambda_i(\mu - 1)\right)$$

  and trigger whenever $\zeta > \frac{T^2}{2}$.
- Crucially, this is possible since solutions at a given time are piece-wise functions which can be manipulated efficienty.
- The (2 or 3) parameters of piece-wise solutions are stored in the memory state and represent changepoints in the timeseries.

How the algorithm works

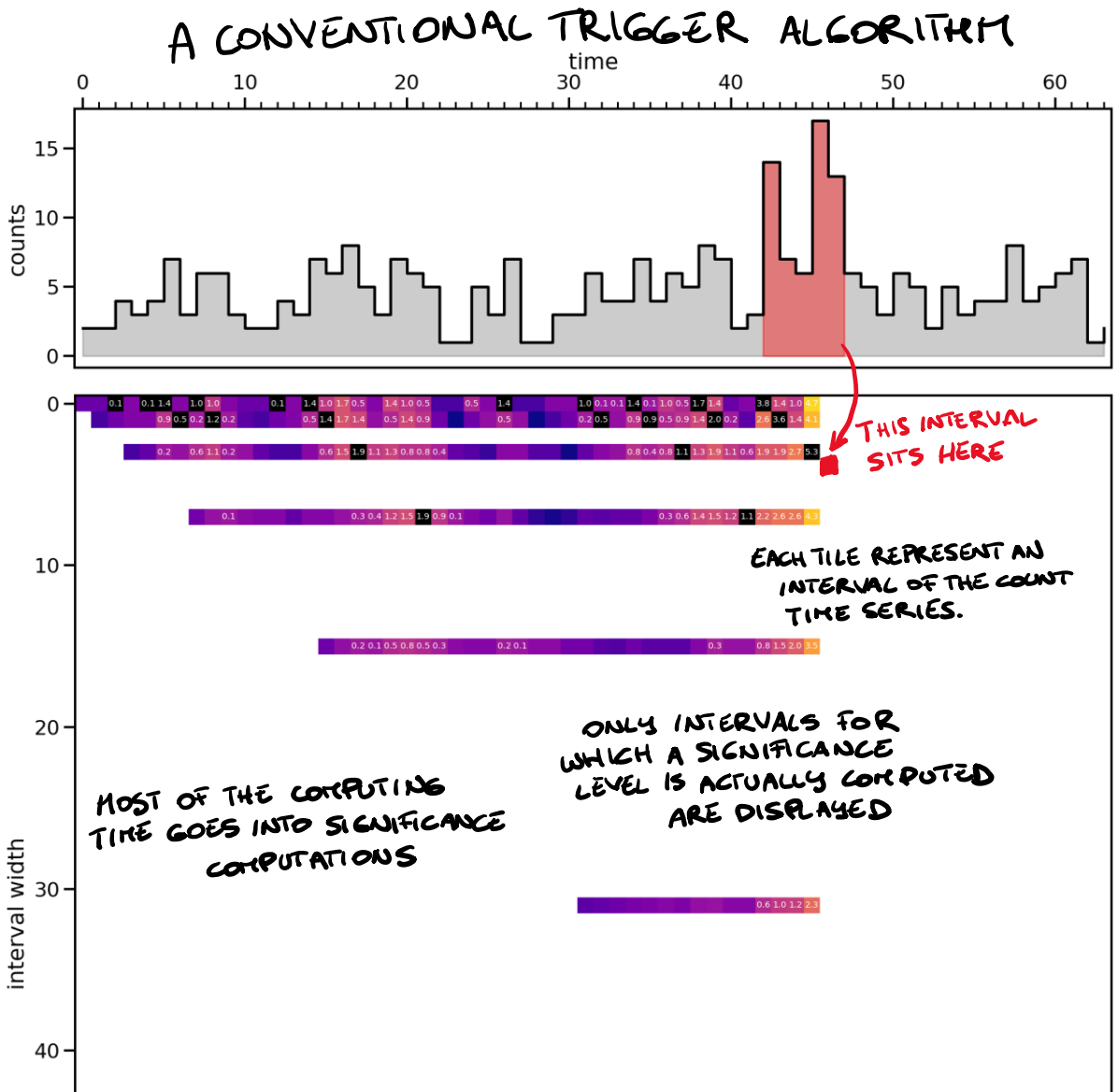
1. At each step, FOCuS takes as input the latest photon count observed and the photon count expected from background.
2. The algorithm mantains a list of “changepoints”.
3. Only changepoints which can possibly result in a trigger are retained between iterations.
4. Whenever the significance of the excess in counts since a changepoint relative to background exceeds a threshold, a trigger is issued.

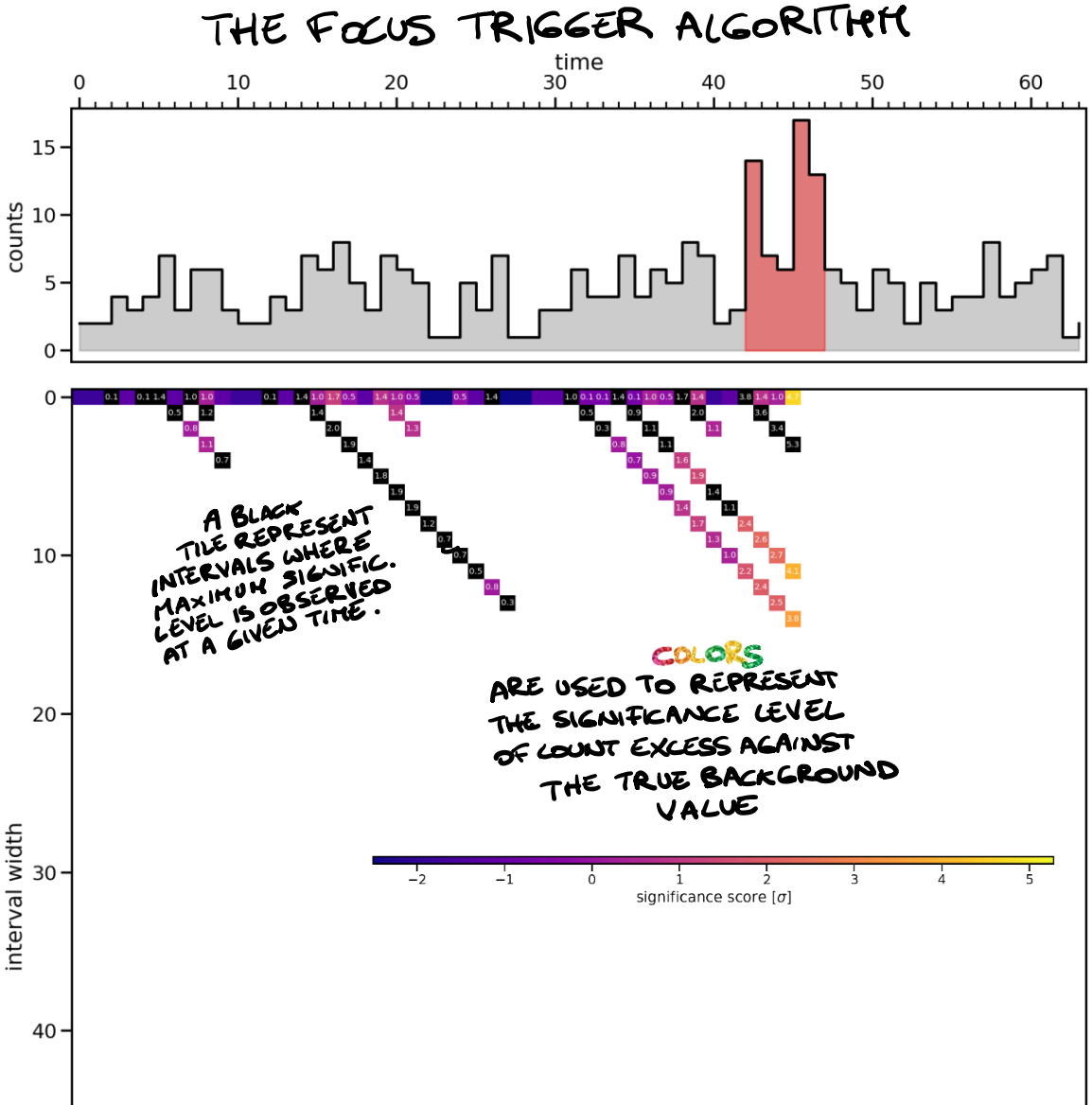
Per-iteration time and memory complexity after t observations is $O(\log(t))$ for exact implementations.

A visual comparison – algorithms traces

https://www.dropbox.com/s/kbes6o98b09sms0/BM_background_visualization_export.html?dl=0

https://www.dropbox.com/s/tsc8udpt3tzc5fu/focus_background_sworcw_export.html?dl=0





Assessing background

Whatever the detection algorithm, you will have to assess a background count-rate against which compare your observation.
In online applications the background level is guessed from the same data which are tested.
In many ways this is a problem of choosing the right filter.

The conventional way – SMA:

$$M_{t,n} = M_{t-1} + \frac{x_t - x_{t-n+1}}{n}$$

Other MA, interesting approach are worth investigation – e.g., EMA:

$$S_t = \alpha x_t + (1 - \alpha) S_{t-1}$$

How to assess significance

The right way to assess counts significance:

$$\alpha = \sum_{i=n}^{\infty} \frac{b^i \exp(-b)}{i!} \qquad \int_S^{\infty} n(x)dx = 1 - \phi(S) = 1 - \alpha$$

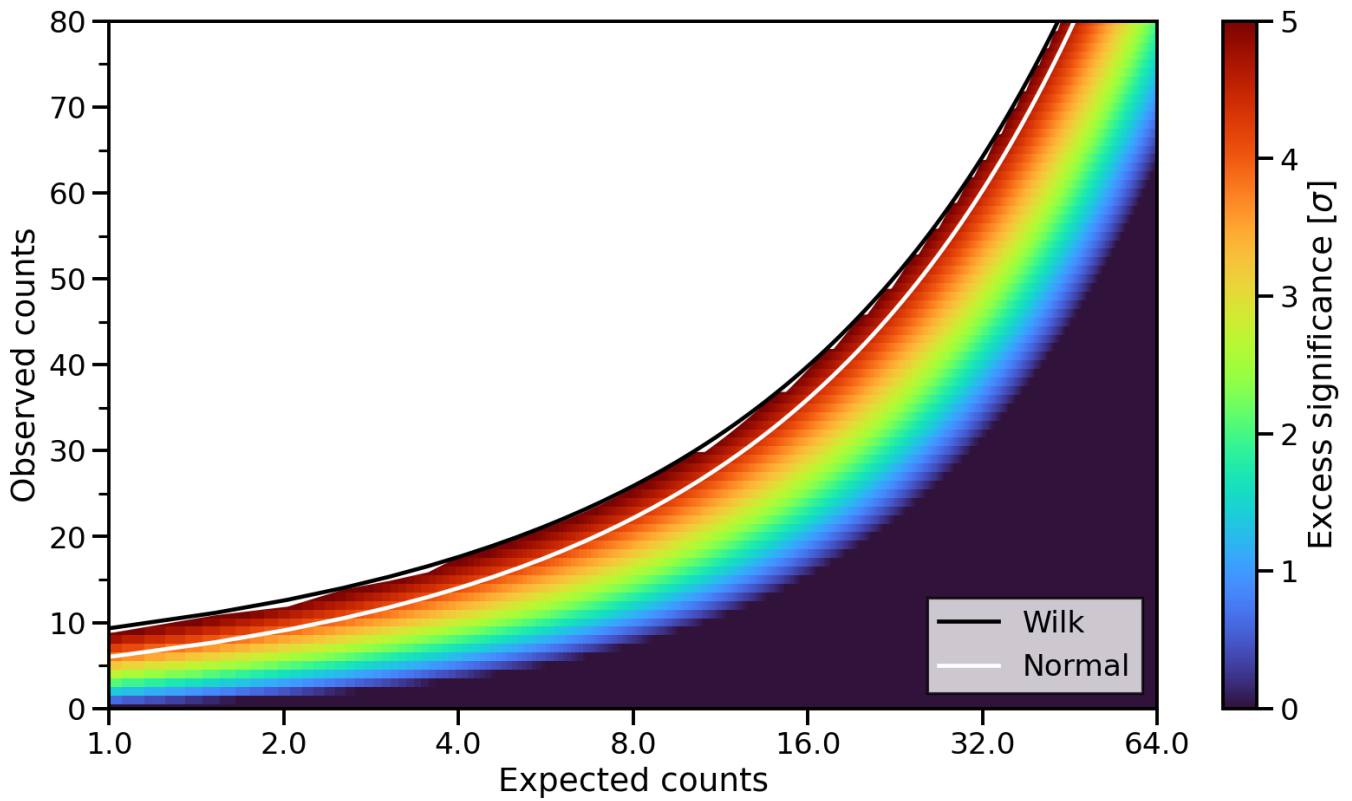
The dangerous way to assess counts significance:

$$S = \frac{n-b}{\sqrt{b}}$$

The safe way to assess significance:

$$S = \sqrt{2n\log(n/b) - 2(n-b)}$$

NO NEED TO COMPUTE THIS! (SQUARE YOUR THRESHOLD INSTEAD ONCE AND FOR ALL)



Testing different algorithms

Three main metrics:

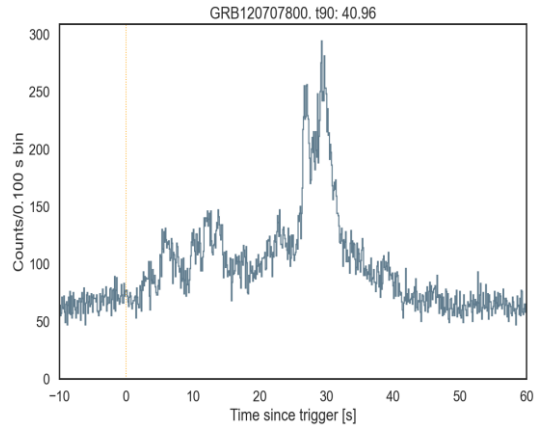
1. Detection power (or false negatives rate)
2. Computing times
3. False positives rates

On both synthethic and real data.

Presently we will cover only results obtained over synthetic data.

Synthetic data generated using a self-made SW tool called SynthBurst.

Detection power – long bursts

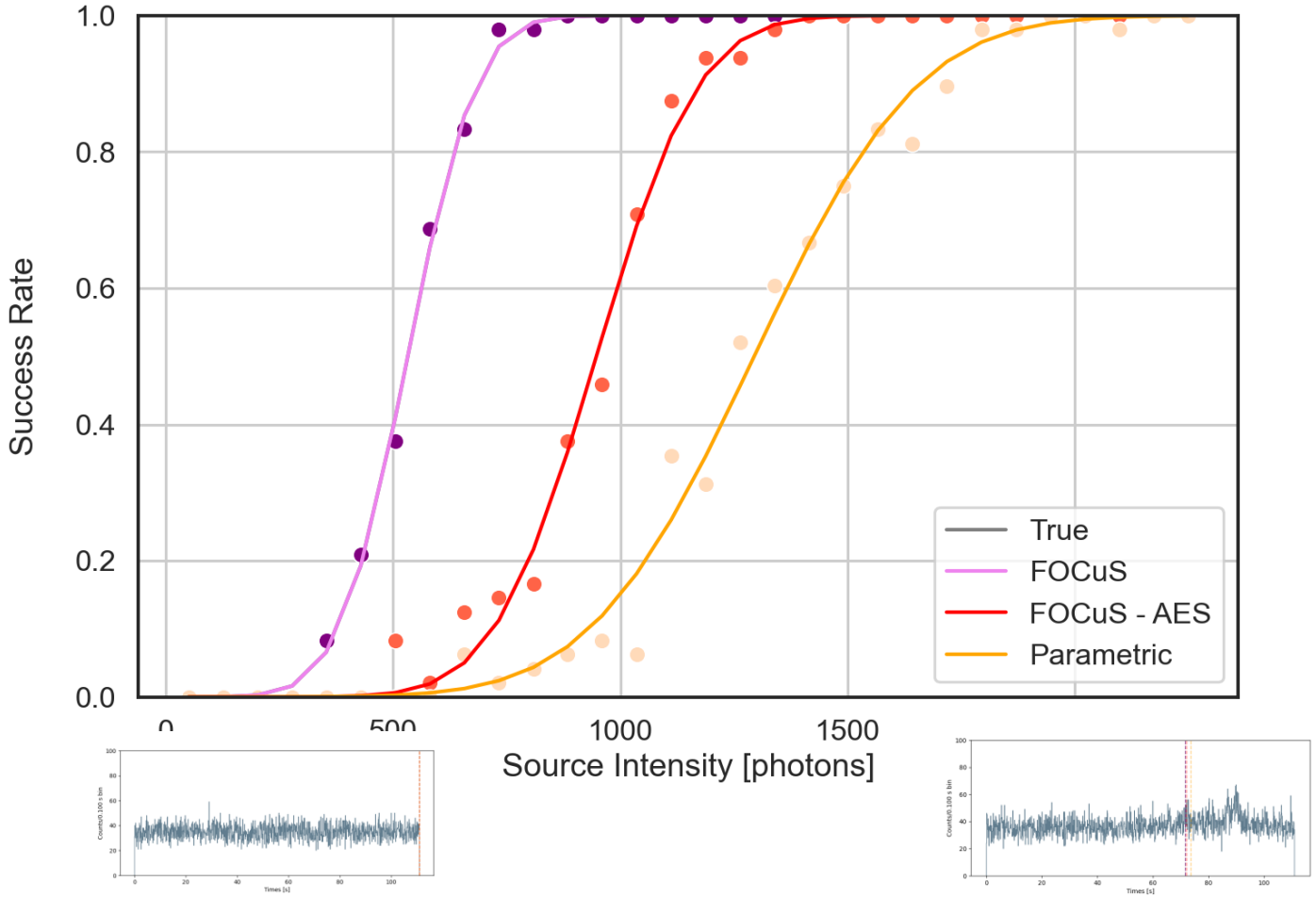


Note:

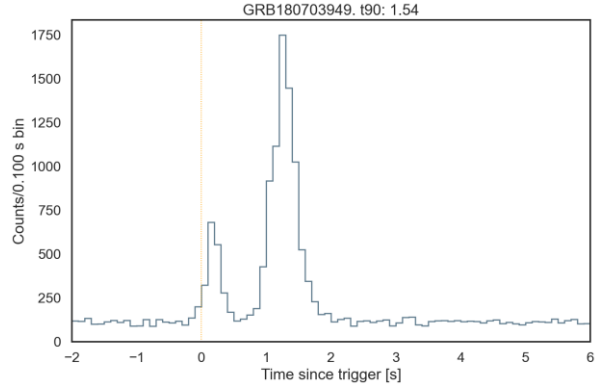
1. 30 brightness steps, 48 repetitions per step.
2. GRB time profile modelled after real observations of GRB120707.

Results:

1. Exact implementation of FOCuS operating with information on the true background rate had performances identical to those of an ideal algorithm.
2. FOCuS-AES detected 81% of the simulated burst which also triggered the ideal algorithm.
3. Parametric detected 63% of the simulated burst which also triggered the ideal algorithm.



Detection power – short burst

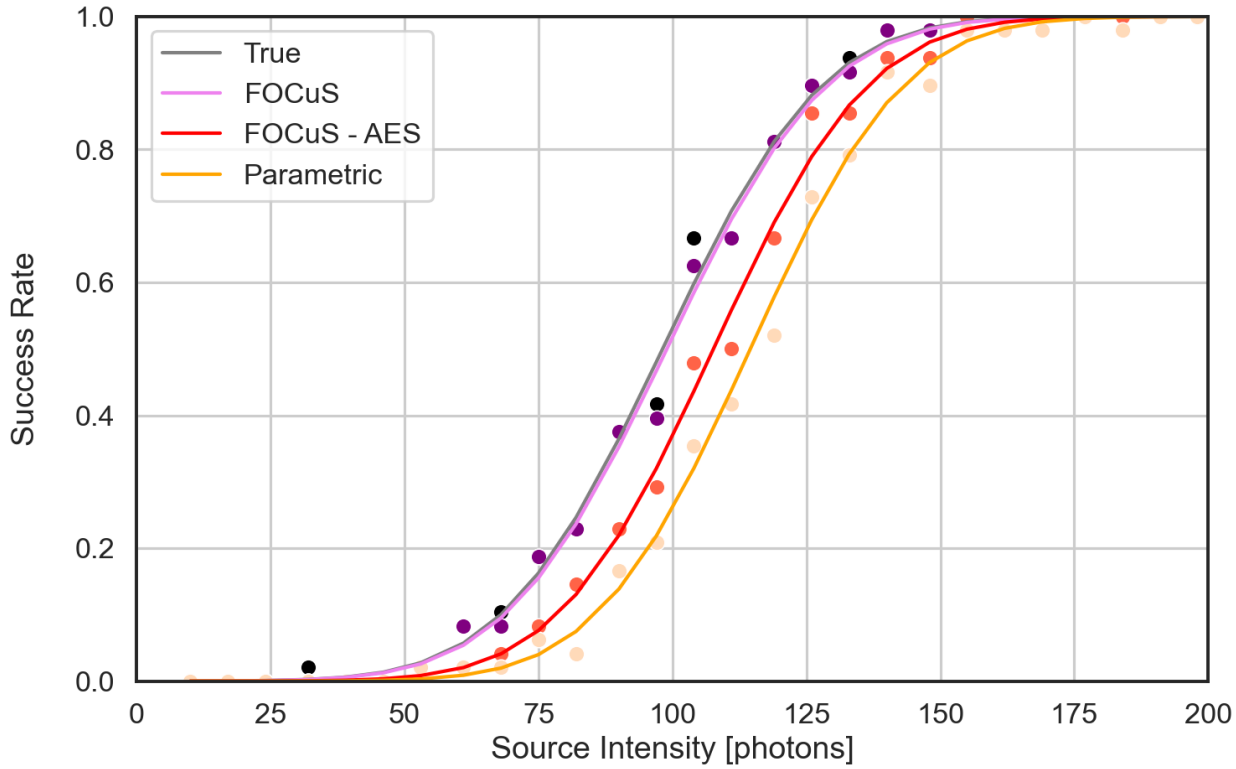


Note:

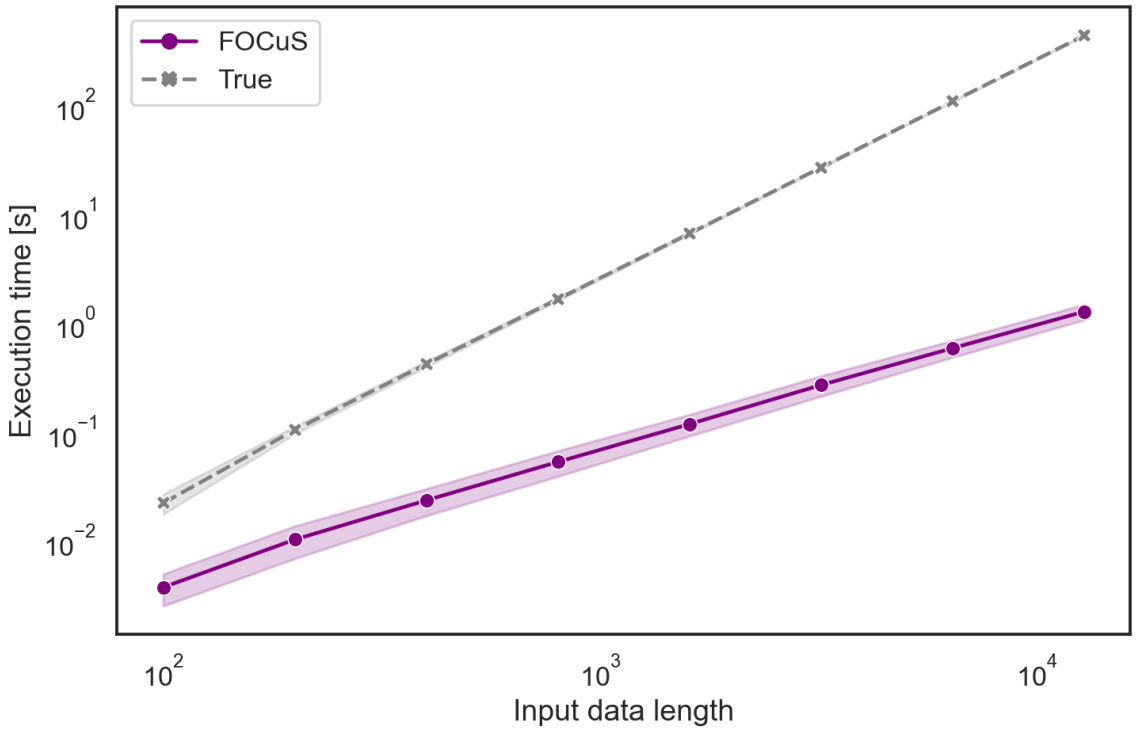
1. Constant, poissonian background.
2. 30 brightness steps, 48 repetitions per step.
3. GRB time profile modelled after real observations of GRB180703.

Results:

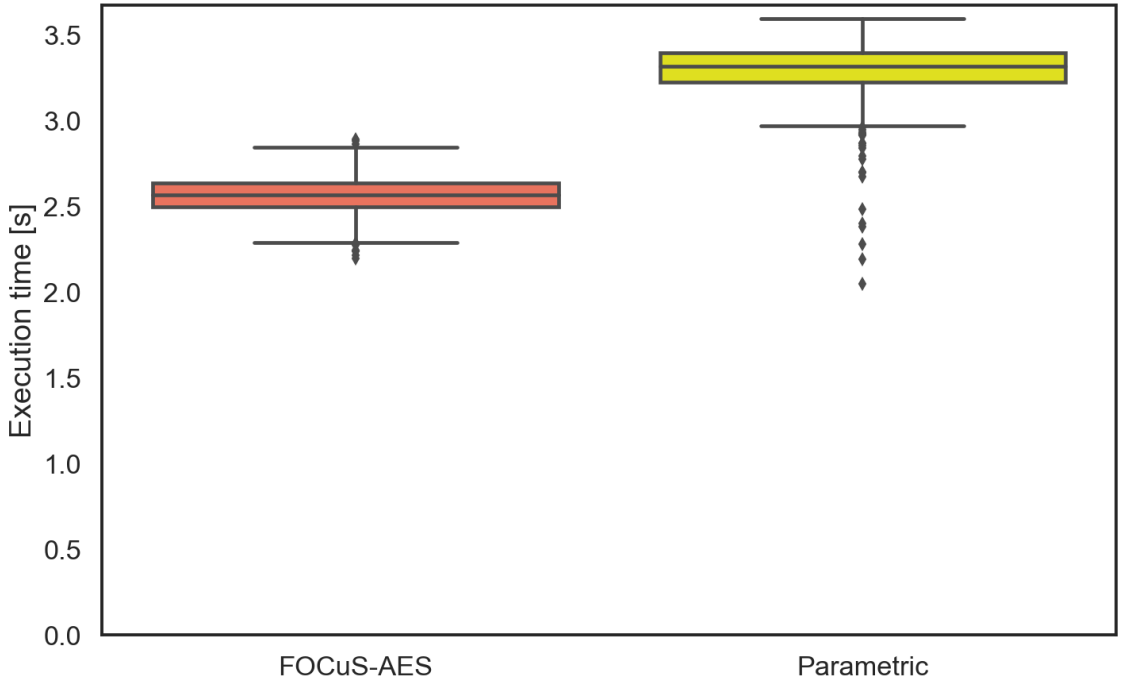
1. Exact implementation of FOCuS operating with information on the true background rate had performances identical to those of an ideal algorithm.
2. FOCuS-AES detected 92% of the simulated burst which also triggered the ideal algorithm.
3. Parametric detected 86% of the simulated burst which also triggered the ideal algorithm.



Computational performances

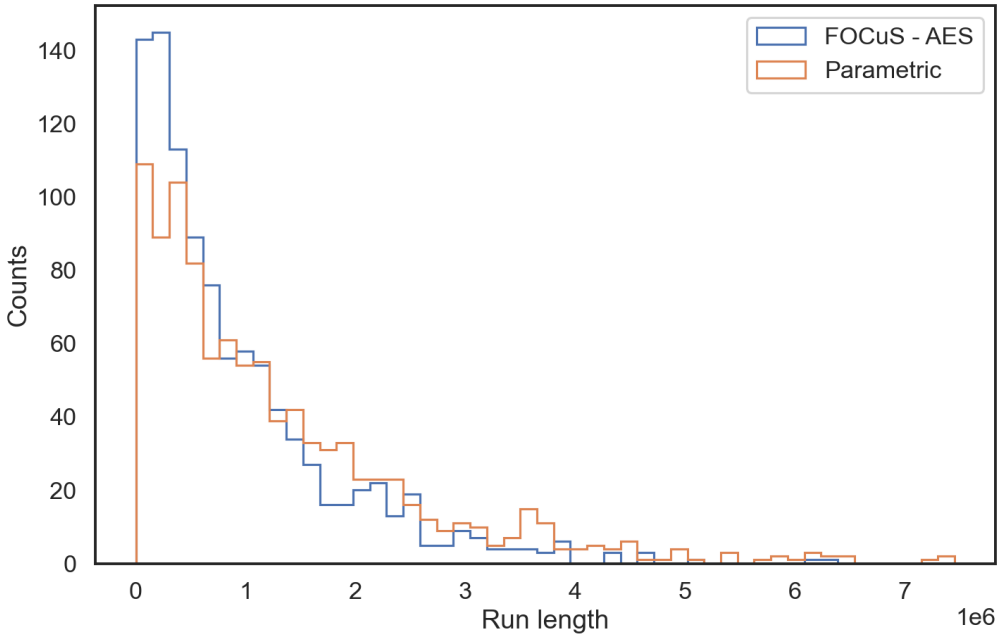


Average time (dots,crosses) and standard deviations running FOCuS and True TAs with infinite threshold over 100 randomly generated time series of counts with different length.

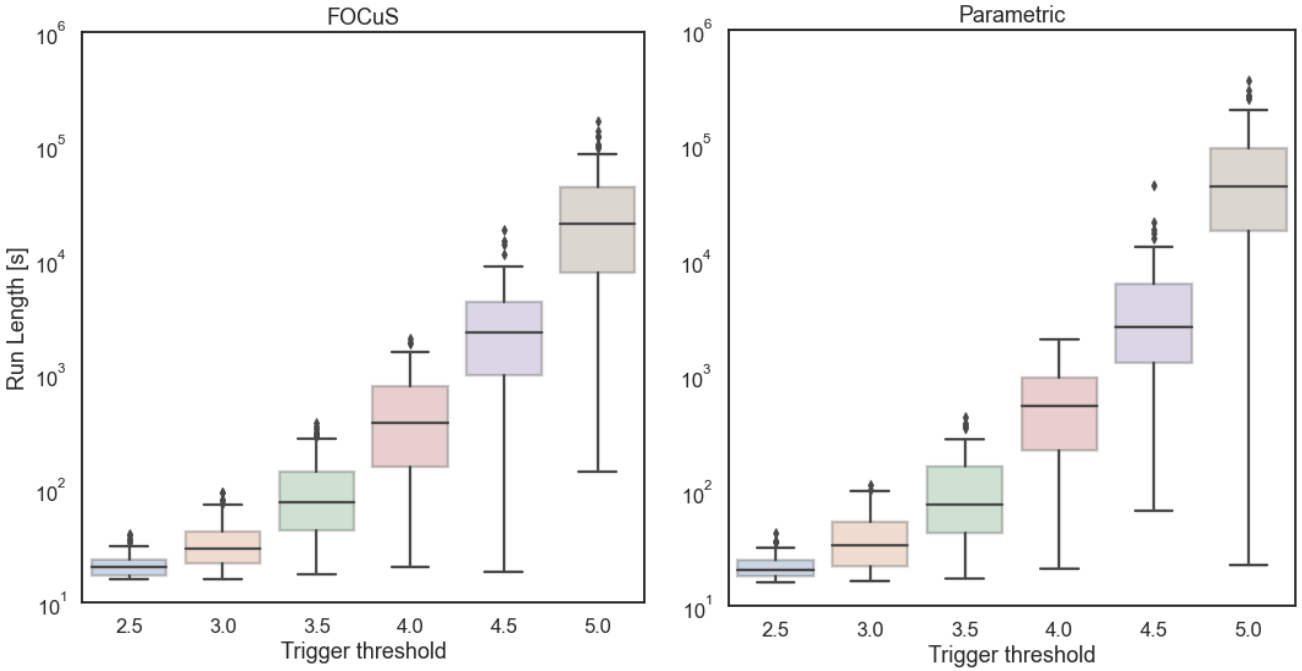


Box plot of the time needed to run over a time series of 10^5 random generated counts sampled from a Poisson distribution with rate parameter λ = 5.6.
FOCuS-AES 22% faster than Parametric.

False positive rate



Run lengths of FOCuS-AES and Parametric with threshold 5.0σ over 1000 Poisson count data streams with constant mean rate. The run length of FOCuS-AES is on average 38% smaller than parametric.



Average Run length as a function of the threshold.

Conclusions

Algorithms for detecting gamma-ray bursts stayed the same during the last 50 years.

More efficient and accurate alternatives exists, one of which is FOCuS.

In a nutshell FOCuS is an algorithm which efficiently tests for the presence of a GRB over *all timescales,* instead of many.

References

- K. Ward, G. Dilillo, I. Eckley, P. Fearnhead; 2022. Submitted. https://arxiv.org/pdf/2208.01494.pdf
- G. Dilillo, K. Ward, R. Crupi, A. Vacchi, F. Fiore. In preparation.

For further inquiries:

- Giuseppe Dilillo, astrophysicist, INAF-IAPS (Rome), giuseppe.dilillo@inaf.it.
- Kester Ward, mathematician, University of Lancaster, k.ward4@lancaster.ac.uk.