The *Bayesian Blocks* algorithm
from time series analysis to histogram representation

Luigi Pertoldi [`pertoldi@pd.infn.it`]

19 July 2018

Università degli Studi di Padova
INFN – Sezione di Padova

Outline

- The *bayesian block* representation is a non-parametric representation of data derived with a bayesian statistical procedure
- Invented by Jeffrey D. Scargle and applied in the context of astronomical time series analysis (GRBs hunting and characterization etc.)

This presentation is mainly based on the following works:

[1] J. D. Scargle *et al.*, Astrophys. J. 764 (2013) 167

[2] B. Pollack *et al.* (2017), `arXiv:1708.00810`

[3] J. D. Scargle, Astrophys. J. 504 (1998) 405

Features in the wish list

- *Non-parametric*: generic representation of data (not fitting!). Another famous technique on the market is e.g. kernel density estimation (KDE)
- Discover local structures in background data exploiting the full information brought by the data
- Impose few preconditions as possible on signal and background shapes
- Handle arbitrary sampling and dynamic ranges of data
- Operate in a bayesian framework and work with posterior probabilities:

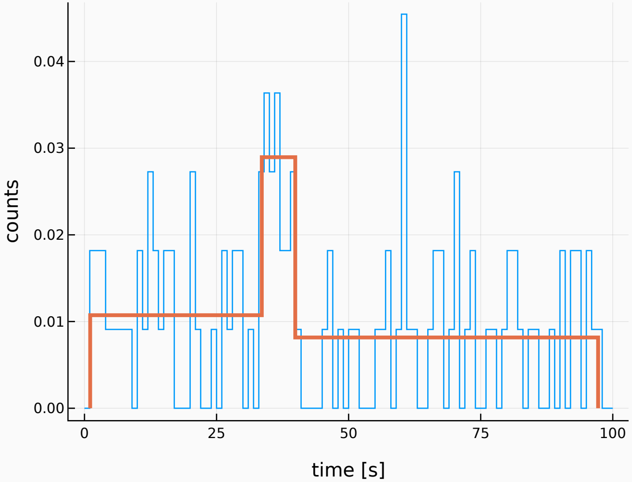
$$P(M|D) \propto P(D|M)P(M)$$

The algorithm

The idea

Segmentation of the data interval into variable-sized blocks, each block containing consecutive data elements satisfying some well-defined criterion.

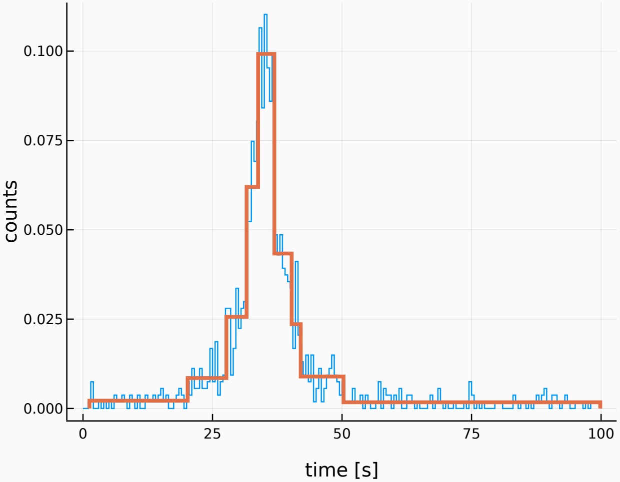
The *optimal segmentation* is the one that maximizes some quantification of this criterion.



The idea

Segmentation of the data interval into variable-sized blocks, each block containing consecutive data elements satisfying some well-defined criterion.

The *optimal segmentation* is the one that maximizes some quantification of this criterion.



The fitness function (a.k.a. the likelihood) — $P(D|M)P(M)$

We'll consider here, for simplicity, 1-dim data and the *Piecewise Constant Model* (a.k.a. constant representation of data within a block)

- **key property:** a block's fitness depends on its data only!
- We are interested only in the length spanned by the block, e.g. the height is just a nuisance parameter
- **Recipe**: build your fitness function and then marginalize (or maximize/minimize) wrt the nuisance parameters
- The fitness of the entire partition will be the product of each block's fitness

The fitness function – Cash statistics

There's a considerable freedom in choosing the fitness function (rely on *sufficient statistics*). Let's use here the Cash statistics (Cash 1979). With a model $M(t, \vartheta)$, the unbinned log-likelihood reads:

$$\log L(\vartheta) = \sum_n \log M(t_n, \vartheta) - \int M(t, \vartheta) dt$$

our model is constant: $M(t, \lambda) = \lambda$, so, for block k:

$$\log L^{(k)}(\lambda) = N^{(k)} \log \lambda - \lambda T^{(k)}$$

Now we maximize wrt the nuisance parameter λ (height of the block)

$$\log L^{(k)}_{\max} = N^{(k)}(\log N^{(k)} - \log T^{(k)}) + N^{(k)}$$

The fitness function – Cash statistics

The $N^{(k)}$ term sums up to N so it can be dropped because it's independent of the partition

$$\log L_{\max}^{(k)} = N^{(k)}(\log N^{(k)} - \log T^{(k)}) + \cancel{N^{(k)}}$$

Has nice features:

- it's simple...
- ...and scale invariant! (try to $T \to \alpha T$)

The fitness of the entire partition will be then:

$$\log L = \sum_k \log L_{\max}^{(k)}$$

The prior for the number of blocks — $P(D|M)P(M)$

A flat prior on the number of blocks is unreasonable, most of the times one looks for a representation where $N_{\text{blocks}} \ll N$ rather than $N_{\text{blocks}} \approx N$. For example the "geometric" prior:

$$P(N_{\text{blocks}}) = \begin{cases} P_0 \gamma^{N_{\text{blocks}}} & 0 \leq N_{\text{blocks}} \leq N \\ 0 & \text{else} \end{cases}$$

has well-understood properties ($\gamma < 1$) and is simply implemented in the algorithm.

The value of γ affects the representation (note that, however, sharply defined structures are retained). But wait, there's an objective procedure to select it...

The prior for N_{blocks} — fixing the parameter

It's a tradeoff between a conservative choice and a liberal choice — it's always a matter of fixing the rate of Type-I errors!

General rule

Running the algorithm with a few different values can be enough. In general, the number of change-points is insensitive to a large range of reasonable values of your "steepness" parameter

Rigorous approach: calibrate the prior as a function of the number of data points N and the *false-positive rate* p_0 on toy pure-noise experiments. A calibration of this type performed in [1] yields:

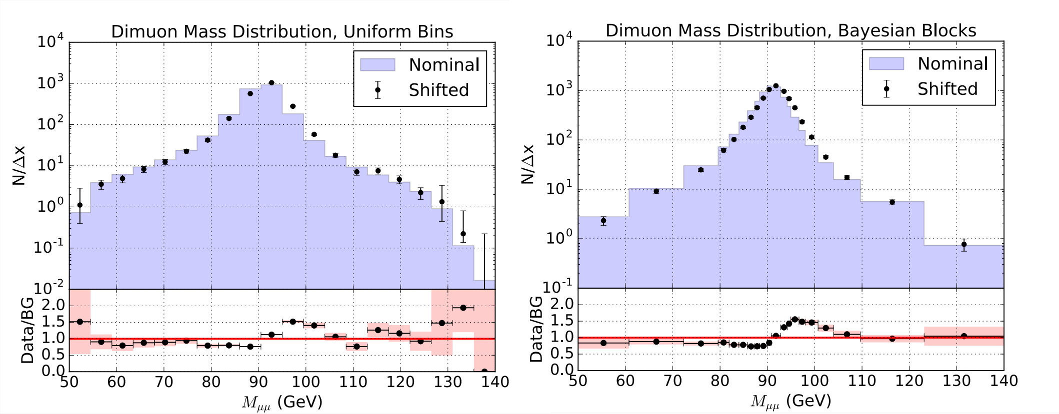
$$\log P(N, p_0) = \log(73.53 p_0 N^{-0.478}) - 4$$

Applications — binning histograms

The idea was developed to be mainly applied to time series analyses (e.g. to spot light flux changes from astrophysical objects), but has advantages also in binning histograms:

- More objective way to present data, avoid *ad-hoc* binning (my guess: forget about systematic uncertainties from binning choice?)
- binning-dependent features can be more objectively spotted
- differently from other rules (Knuth's, Scott's...) it doesn't use fixed bin width
- attractive when data spans different orders of magnitude
- effective hypothesis testing with the hybrid scheme proposed in [2]

Though can still seem crude and blocky to someone...



Taken from [2], note the definiteness of the distortion pattern in the residuals with the bayesian block representation.

Implementation

Investigating the optimality of 2^N data partitions isn't a quick task for a computer $\rightarrow$ dynamic programming approach following the spirit of mathematical induction:

- Sort the data and start from the first one, the only possible partition is trivially optimal
- The optimal partition is updated at each step using the information from the previous ones $\rightarrow \mathcal{O}(N^2)$

```
for k in 1:N
# fitness function + prior
    F(r) = logfitness(N_k, T_k) + log_prior
    # compute all possible configurations
    A = [F(r) + (r == 1 ? 0 : best[r-1]) for r in 1:k]
    # save best configuration
    push!(last, indmax(A))
    push!(best, maximum(A))
end
```

Implementation

An implementation in Python exists, provided by the `AstroML` package[1], but it's targeted to time series analyses and it's slow when applied to large datasets.

I implemented a faster version in Julia[2] available here:

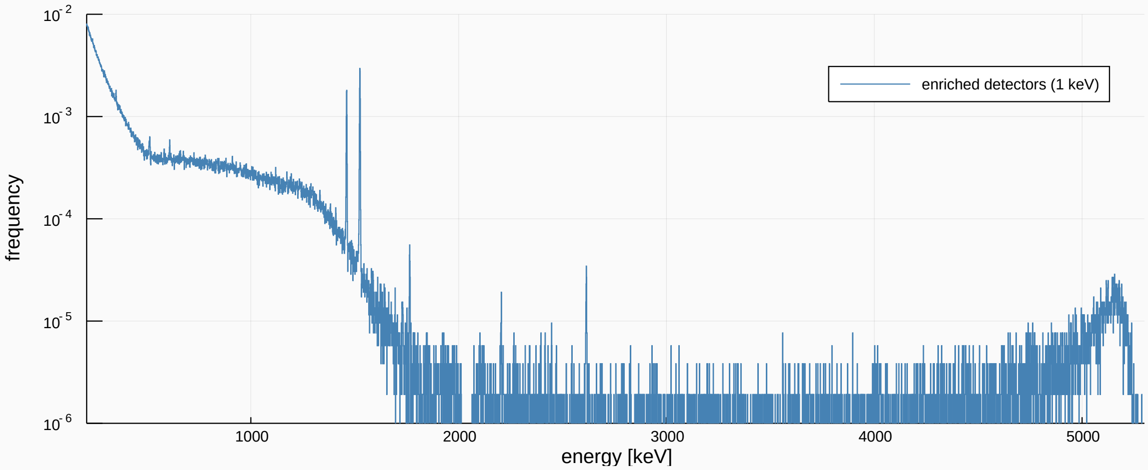
`https://tinyurl.com/bayesian-blocks-jl`

that can also rebin histograms.

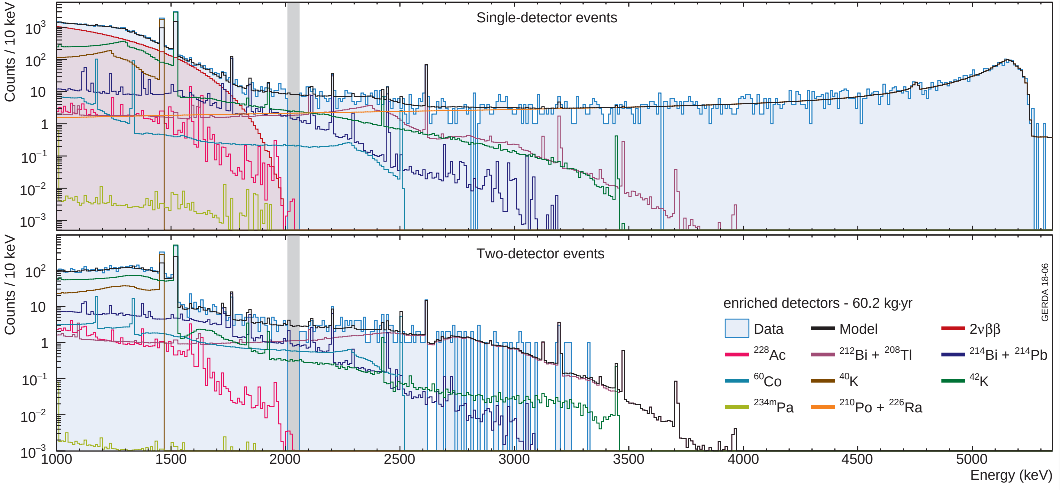
[1] `http://www.astroml.org`
[2] `https://julialang.org`

Application: GERDA's energy spectrum

Application — GERDA's energy spectrum



Application — GERDA's energy spectrum



Application — GERDA's energy spectrum

Problem

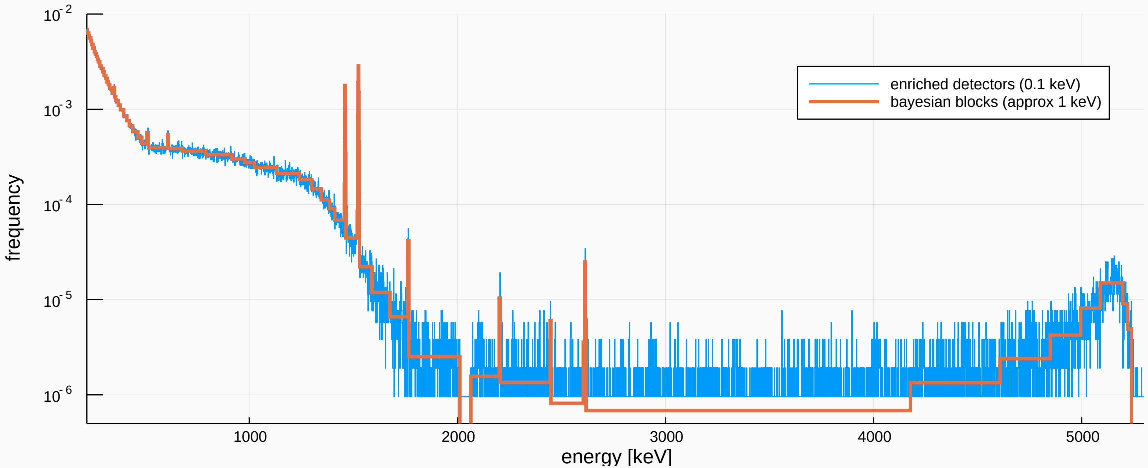
The spectrum consists of $\sim 10^6$ entries, even with $\mathcal{O}(N^2)$ instead of $\mathcal{O}(2^N)$ and fast code the computational effort is quite large!

Approximation

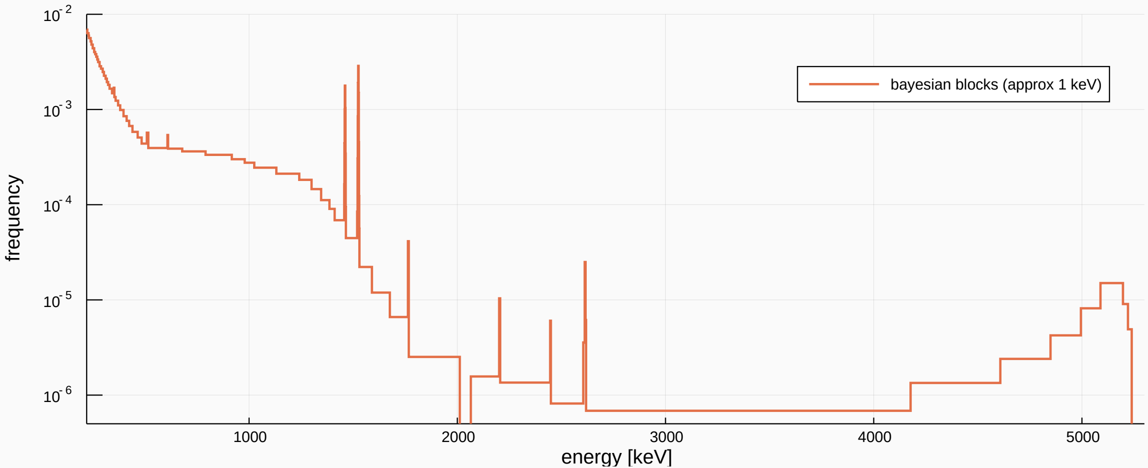
adopted for high statistics: bin the data and treat each bin as a data point of multiplicity = bincontent in the algorithm.

NB: I adopted the calibrated prior computed in [1] with $p_0 = 0.1$ in all the following plots.

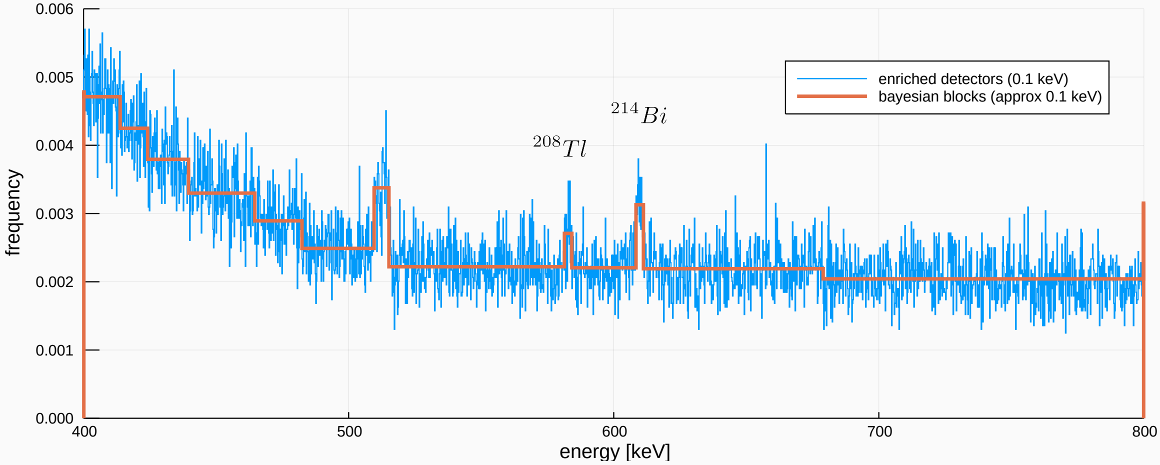
Application — GERDA's energy spectrum



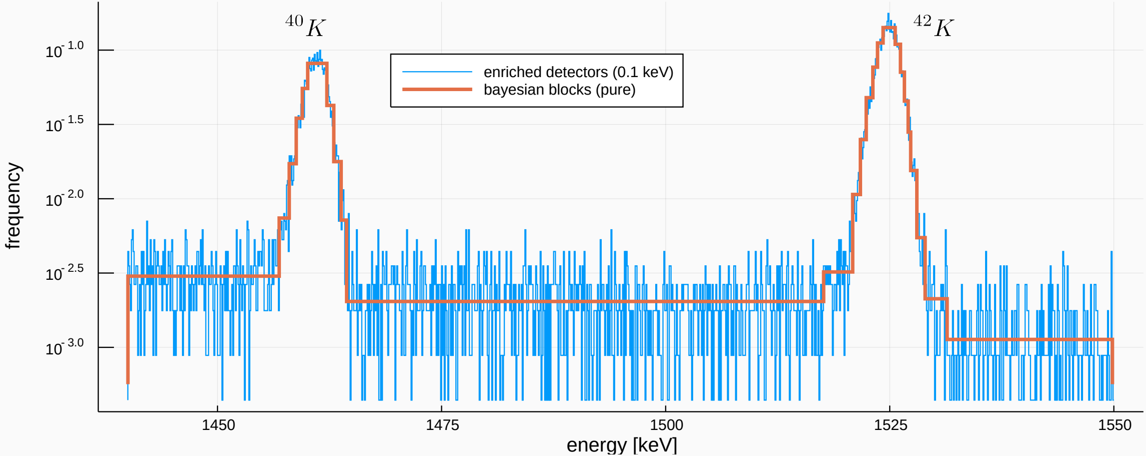
Application — GERDA's energy spectrum



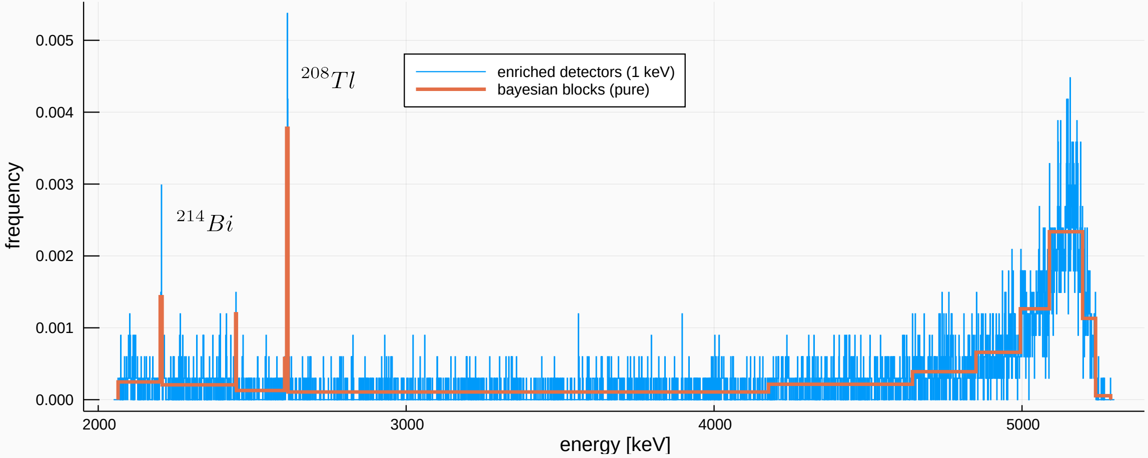
Application — GERDA's energy spectrum



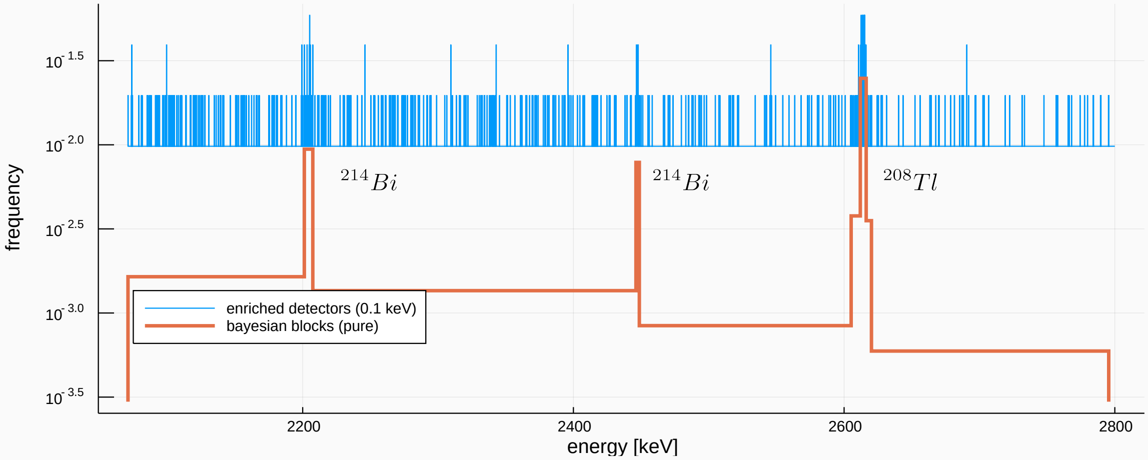
Application — GERDA's energy spectrum



Application — GERDA's energy spectrum



Application — GERDA's energy spectrum



Conclusions

The bayesian block representation provides an objective way to enlighten the key features of a data set by imposing few preconditions as possible.

Tips:

- choose a fitness function that's suitable for your data
- experiment different priors or...
- ...run the algorithm on toy data to calibrate it

Future:

- How is it to fit data with this binning? Is this data representation as informative as a fine, 1 keV, fixed-width binning? $\rightarrow$ verify statement in [2]

References

[1] *Studies in Astronomical Time Series Analysis. VI. Bayesian Block Representations*, J. D. Scargle *et al.*, Astrophys. J. 764 (2013) 167

[2] *Bayesian Blocks in High Energy Physics: Better Binning made easy!* B. Pollack *et al.* (2017), `arXiv:1708.00810`

[3] *Studies in astronomical time series analysis: 5. Bayesian blocks, a new method to analyze structure in photon counting data*, J. D. Scargle, Astrophys. J. 504 (1998) 405