The Astrophysical Journal, 764:167 (26pp), 2013 February 20 doi:10.1088/0004-637X/764/2/167

STUDIES IN ASTRONOMICAL TIME SERIES ANALYSIS. VI. BAYESIAN BLOCK REPRESENTATIONS

Jeffrey D. Scargle[1], Jay P. Norris[2], Brad Jackson[3], and James Chiang[4]
[1] Space Science and Astrobiology Division, MS 245-3, NASA Ames Research Center, Moffett Field, CA 94035-1000, USA; jeffrey.d.scargle@nasa.gov
[2] Physics Department, Boise State University, 2110 University Drive, Boise, ID 83725-1570, USA
[3] The Center for Applied Mathematics and Computer Science, Department of Mathematics, San José State University, One Washington Square, MH 308, San José, CA 95192-0103, USA
[4] W. W. Hansen Experimental Physics Laboratory, Kavli Institute for Particle Astrophysics and Cosmology, Department of Physics and SLAC National Accelerator Laboratory, Stanford University, Stanford, CA 94305, USA
Received 2012 August 8; accepted 2012 December 14; published 2013 February 4

ABSTRACT

This paper addresses the problem of detecting and characterizing local variability in time series and other forms of sequential data. The goal is to identify and characterize statistically significant variations, at the same time suppressing the inevitable corrupting observational errors. We present a simple nonparametric modeling technique and an algorithm implementing it—an improved and generalized version of *Bayesian Blocks*—that finds the optimal segmentation of the data in the observation interval. The structure of the algorithm allows it to be used in either a real-time *trigger* mode, or a *retrospective* mode. Maximum likelihood or marginal posterior functions to measure model fitness are presented for events, binned counts, and measurements at arbitrary times with known error distributions. Problems addressed include those connected with data gaps, variable exposure, extension to piecewise linear and piecewise exponential representations, multivariate time series data, analysis of variance, data on the circle, other data modes, and dispersed data. Simulations provide evidence that the detection efficiency for weak signals is close to a theoretical asymptotic limit derived by Arias-Castro et al. In the spirit of Reproducible Research all of the code and data necessary to reproduce all of the figures in this paper are included as supplementary material.

Key words: methods: data analysis – methods: statistical

Online-only material: Supplemental data file (tar.gz)

"The line is similar to a length of time, and as the points are the beginning and end of the line, so the instants are the endpoints of any given extension of time." *Leonardo da Vinci, Codex Arundel, folio 190v.*, c. 1500 A.D. (Capra 2007).

1. THE DATA ANALYSIS SETTING

This paper describes a method for nonparametric analysis of time series data to detect and characterize structure localized in time. *Nonparametric* methods seek generic representations, in contrast to fitting of models to the data. By *local* structure we mean light-curve features occupying sub-ranges of the total observation interval, in contrast to global signals present all or most of the time (e.g., periodicities) for which Fourier, wavelet, or other transform methods are more appropriate. Wavelets enjoy the best of both worlds, being effective for global and local, short duration, signals. Our adaptive segmentation approach is much in the spirit of wavelet analysis freed of its restriction to basis functions supported on fixed dyadic intervals.

The goal is to separate statistically significant features from the ever-present random observational errors. Although phrased in the time domain the discussion throughout is applicable to measurements sequential in wavelength, spatial quantities, or any other independent variable. This setting leads to the following desiderata: The ideal algorithm would impose as few preconditions as possible, avoiding assumptions about smoothness or shape of the signal that place a priori limitations on scales and resolution. The algorithm should handle arbitrary sampling (i.e., not be limited to gapless, evenly spaced data) and large dynamic ranges in amplitude, timescale, and signal to noise. For scientific data mining applications and for objectivity, the method should be largely automatic. To the extent possible it should suppress observational errors while preserving whatever valid information lies in the data. It should be applicable to multivariate problems. It should incorporate variation of the exposure or instrumental efficiency during the measurement, as well auxiliary, extrinsic information, e.g., spectral or color information. It should be able to operate both retrospectively (analyze all the data after they are collected) and in a real-time fashion that triggers on the first significant variation of the signal from its background level.

The algorithm described here has considerable success in achieving each of these features. With a simple and easy-to-use computational framework it represents the structure of the signal in a form handy for further analysis and the estimation of physically meaningful quantities. It includes an automatic penalty for model complexity, thus solving the vexing problems associated with model comparison in general and *determining the order of the model* in particular. It is exact, not a *greedy* approximation as in Scargle (1998). This term refers to algorithms that make improvements at each iteration but in the long run are short-sighted and not guaranteed to converge to a global optimum.

In a similar context the reference (Gregory & Loredo 1992), especially Appendix C, discusses evenly spaced block representations of time series for the detection of periodicities and other features in event data. Versions of our algorithm have been used in various applications, such as Horvath et al. (2005), Norris et al. (2010), Norris et al. (2011), Way et al. (2011), and Qin et al. (2013).

The following sections discuss, in turn, the basis of segmentation analysis (Section 1.1), the piecewise constant model adopted in this work (Section 1.2), extensions to piecewise linear and piecewise exponential models (Section 1.3), the types of data that the algorithm can accept (Sections 1.5 and 1.6),

data gaps (Section 1.7), exposure variations (Section 1.8), a parameter from the prior on the number of blocks (Section 1.9), generalities of optimal segmentation of data into blocks (Section 2), some error analysis (Section 2.8), a variety of block fitness functions (Section 3), and sample applications (Section 4). Appendices present some MatLab™ (The Mathworks, Inc.) and IDL TM (Exelis Visual Information Solutions™) code, some miscellaneous results, and details of other data modes, including dispersed data (Appendix C.8). Ancillary files are available providing scripts and data in order to reproduce all of the figures in this paper.

1.1. Optimal Segmentation Analysis

The above considerations point toward the most generic possible nonparametric data model, and have motivated the development of data segmentation and change-point methods—see, e.g., Ò Ruanaidh & Fitzgerald (1996) and Scargle (1998). These methods represent the signal structure in terms of a segmentation of the time interval into blocks, with each block containing consecutive data elements satisfying some well-defined criterion. The *optimal segmentation* is that which maximizes some quantitative expression of the criterion—for example, the sum over blocks of a goodness-of-fit measure of a simple model of the data lying in each block.

These concepts and methods can be applied in surprisingly general, higher dimensional contexts. Here, however, we concentrate on one-dimensional data ordered sequentially with respect to time or some other independent variable. In this setting segmentation analysis is often called *change-point detection*, since it implements models in which a signal's statistical properties change discontinuously at discrete times but are constant in the segments between these change points (see Section 2.5).

1.2. The Piecewise Constant Model

It is remarkable that all of the desiderata outlined in the previous section can be achieved in large degree by optimal fitting of a piecewise constant model to the data. The range of the independent variable (e.g., time) is divided into subintervals (here called *blocks*) generally unequal in size, in which the dependent variable (e.g., intensity) is modeled as constant within errors. Of all possible such "step functions" this approach yields the best one by maximizing some goodness-of-fit measure.

Defining the times ending one block and starting the next as *change points*, the model of the whole observation interval contains these parameters:

1. $N_{\rm cp}$: the number of change points,
2. $t_k^{\rm cp}$: the change-point starting block k (and ending block $k-1$),
3. X_k: the signal amplitude in block k,

for $k = 1, 2, \ldots N_{\rm cp}+1$. There is one more block than there are change points: The first datum is always considered a change point, marking the start of the first block, and is therefore not a free parameter. If the last datum is a change point, it denotes a block consisting solely of that datum.

The key idea is that the blocks can be treated independently, in the sense that a block's fitness depends on its data only. Our simple model for each block has effectively two parameters. The first represents the signal amplitude, and is treated as a nuisance parameter to be determined after the change points have been located. The second parameter is the length of the interval spanned by the block. (The actual start and stop times of this interval are needed for piecing blocks together to form the final signal representation, but not for the fitness calculation.)

How many blocks? A key issue is how to determine the number of blocks, $N_{\rm blocks} = N_{\rm cp} + 1$. Nonparametric analysis invariably involves controlling in one way or another the complexity of the estimated representation. Typically, such regulation is considered a tradeoff of bias and variance, often implemented by adjusting a smoothing parameter.

But smoothing is one of the very things we are trying to avoid. The discontinuities at the block edges are regarded as assets, not liabilities to be smoothed over. So rather than smooth, we influence the number of blocks by defining a prior distribution for the number of blocks. Adjusting a parameter controlling the steepness of this prior establishes relative probabilities of smaller or larger numbers of blocks. In the usual fashion for Bayesian model selection in cases with high signal to noise, $N_{\rm blocks}$ is determined by the structure of the signal; with lower signal to noise, the prior becomes more and more important. In short, we are regulating not smoothness but complexity, much in the way that wavelet denoising (Donoho & Johnstone 1998) operates without smoothing over sharp features as long as they are supported by the data. The issues centered around the adopted prior and the determination of its parameter are further discussed in Section 1.9 below.

This segmented representation is in the spirit of nonparametric approximation and not meant to imply that we believe the signal is actually discontinuous. The sometimes crude and blocky appearance of this model may be awkward in visualization contexts, but for deriving physically meaningful quantities it is not. Blocky models are broadly useful in signal processing (Donoho 1994) and have several motivations. Their simplicity allows exact treatment of various quantities, such as the likelihood. We can optimize or marginalize the rate parameters exactly, giving simple formulas for the fitness function (see Section 3 and Appendix C). Furthermore, in many applications the estimated model itself is less important than quantities derived from it. For example, while smoothed plots of pulses within gamma-ray bursts (GRBs) make pretty pictures, one is really interested in pulse locations, lags, amplitudes, widths, rise and decay times, etc. All of these quantities can be determined directly from the locations, heights, and widths of the blocks—accurately and free of any smoothness assumptions.

1.3. Piecewise Linear and Exponential Models

Some researchers have applied segmentation methods with other block representations. For example, *piecewise linear* models have been used in measuring similarity among time series and in pattern matching (Lin et al. 2003) and to represent time series generated by nonlinear processes (Tong 1990). While such models may seem better than discontinuous step functions, their improved flexibility is somewhat offset by added complexity of the model and its interpretation. Note further that if continuity is imposed at the change points, a piecewise linear model has essentially the same number of degrees of freedom as does the simpler piecewise constant model.

We mention two such generalizations, one modeling the signal as linear in time across the block:

$$x(t) = \lambda(1 + a(t - t_0)), \tag{1}$$

and the second as exponential:

$$x(t) = \lambda e^{a(t-t_0)}. \tag{2}$$

In both cases λ is the signal strength at the fiducial time t_0 and the coefficient a determines the rate of variation over the block. Such models may be useful in spite of the caveats mentioned above and the added complexity of the block fitness functions. Hence we provide some details in Appendices C, C.9, and C.10.

1.4. Histograms

Generally speaking data analysis requires very different algorithms depending on whether or not the underlying measurements are sequential. However constructing a histogram of non-sequential measured values is very similar to estimation of a piecewise constant model for the same data treated as if they were sequential. Hence histograms can be constructed by simply ordering the measured values and applying our algorithm for event data.

This approach automatically leads to generalized histograms in which the bins adapt to the data and are neither constrained to be equal nor is their number or size pre-defined. This way of constructing density representations with histograms has many advantages. It avoids information loss due to arbitrarily chosen bins. As well it short-circuits the dependence of the density estimate on such choices, thereby circumventing the temptation to fiddle with bins to emphasize some desired feature and then ignore the trials factor in the significance analysis. In many astronomical applications there is a great advantage to structure estimation that imposes no preconditions on resolution, and indeed allows increased resolution where it is supported by the data. Once one determines the parameter in the prior on the number of bins, `ncp_prior`, one has an objective histogram procedure in which the number, individual sizes, and locations of the bins are determined solely and uniquely by the data.

A future publication will detail this approach to histograms and exhibit its advantages over non-adaptive histograms.

1.5. Data Modes

The algorithms developed here can be used with a variety of types of data, often called *data modes* in instrumentation contexts. An earlier paper (Scargle 1998) described several, with formulas for the corresponding fitness functions. Here we discuss data modes in a broader perspective. It is required that the measurements provide sufficient information to determine which block they belong to and then to compute the model fitness function for the block (cf. Section 2.3).

Almost any physical variable and any measurement scheme for it, discrete or quasi-continuous, can be accommodated. In the simple one-dimensional case treated here, the independent variable is time, wavelength, or some other quantity. The *data space* is the domain of this variable over which measurements were made—typically an interval, possibly interrupted by gaps during which the measuring system was not operating.

The measured quantity can be almost anything that yields information about the target signal. The three most common examples emphasized here are: (1) times of events (often called time-tagged event (TTE) data), (2) counts of events in time bins, and (3) measurements of a quasi-continuous observable at a sequence of points in time. For the first two cases the signal of interest is the *event rate*, proportional to the probability distribution regulating events that occur at discrete times due to the nature of the astrophysical process and/or the way it is recorded. We call case (3) *point measurements*, not to be confused with *point data* (also called event data). These modes have much in common, as they all comprise measurements that can be at any time; what differentiates them is their statistics, roughly speaking Bernoulli, Poisson, and Gaussian (or perhaps some other), respectively.

The archetypal example of (1) is light collected by a telescope and recorded as a set of detection times of individual photons to study source variability. Case (2) is similar, but with the events collected into bins—which do not have to be equal or evenly spaced. Case (3) is common when photons are not detected individually, such as in radio flux measurements. In all cases it is useful to represent the measurements with *data cells*, typically one for each measurement (see Section 2.2). In principle mixtures of cells from different data types can be handled, as described in the next section.

1.6. Mixed Data Modes

In the course of an observation sequence the data mode may change for any one of a number of reasons. For example, a buffer containing photon arrival times may overflow, triggering a switch to a less voluminous mode such as time-to-spill or binned data. Another example is GRB data in which a switch to smaller bins occurs near the time of the burst trigger. Our algorithm can analyze essentially arbitrary mixtures of data types within a single time series, simply by ensuring that the block cost function is based on whatever data lies within it. The term *mixed data modes* connotes measurements of the same quantity with different data modes at different times, whereas the related concept of *multivariate time series* refers to several simultaneous data streams with different data modes or perhaps even measuring very different physical quantities. Therefore implementation details will not be given here, but deferred to Section 4.2, where we discuss this more general context.

1.7. Gaps

In many cases there are subintervals of time over which no data were obtained,[5] for diverse reasons characterized as stochastic (weather, instrument malfunction), periodic (daily, monthly, annual cycles), and even sociological (telescope assignment committee vagaries, the perceived importance of future observations based on past data, and the reaction of the scientific community to same).

Mathematically these may be viewed as processes with random and deterministic components, but in practice one rarely knows enough for a statistical treatment to be useful. Accordingly, gaps are almost always treated as simply given, and we do so here.

Such data gaps have a nearly invisible affect on the algorithm, fundamentally due to the fact that it operates locally in the time domain. For event data all that matters is the *live time* during the block, i.e., the time over which data could have been registered. Other than correcting the total time span of any putative block containing data gaps by subtracting the corresponding *dead time*, gaps can be handled by ignoring them. Operationally, one simply treats the data right after a gap as immediately following the data right before it (and not delayed by the length of the gap). Think of this as squeezing the interval to eliminate the gaps, carrying out the analysis as if no gaps are present, and then undoing the squeezing by restoring the original times. This procedure is valid because event independence means that the fitness of a block depends on only its total live time and the events within it.

[5] These are very different from intervals in which no events happened to be detected due to low event rate.

For event data, this squeezing can be implemented by subtracting from each event time the sum of the lengths of all the preceding gaps. One small detail concerns the points just before and just after a gap. One might think their time intervals should be computed relative to the gap edges. But it follows from the nature of independent events (see Appendix B) that they can be computed as though the gap did not exist.

The only other subtlety lies in interpreting the model in and around gaps. There are two possibilities: A given gap (1) may lie completely within a block or (2) it may separate two blocks. Case (1) can be taken as evidence that the event rates before and after the gap are deemed the same within statistical fluctuations. Case (2), on the other hand, implies that the event rate did change significantly.

Of course the gaps must be restored for display and other post-processing analysis. Think of this as unsqueezing the data so that all blocks appear at their correct locations in time. Keeping in mind that there is no direct information about what happened during unobserved intervals, plots should probably include some indication that rates within gaps are unknown or uncertain, such as by use of dotted lines or shading in the gap for case (1) or leaving the gap interval completely blank in case (2).

For the case of point measurements the situation is different. In one sense there are no gaps at all, and in another sense the entire observation interval consists of many gaps separating tiny intervals over which the measurements were actually made. One is hard-pressed to make a statistical distinction between various reasons why there is not a measurement at a given time—e.g., detector and weather problems, or simply a choice as to how to allocate observing time (a choice that may even depend on the results of analyzing previous data). Basically, the blocks in this case represent intervals where whatever measurements were made in the interval are consistent with a signal that is constant over that interval.

Note that things would be different if one wanted to define a fitness function dependent on the total length of the block, not just its live time. This would arise, for example, if a prior on the block length were imposed. Such possibilities will not be discussed here, nor will the rare exceptions where a statistical treatment of the gaps is warranted.

1.8. Exposure Variations

In some applications the effective instrument response is not constant. The measurements then reflect true source variations modified by changes in overall throughput of the detection system. We use the term *exposure* for any such effect on the detected signal—e.g., detector efficiency, telescope effective area, beam pattern, and point-spread function effects. Exposure can be quantified by the ratio of the expected signal with and without any such effects. It may be calculable from properties of the observing system, determined after the fact through some sort of calibration procedure, or a combination of the two. Here we assume that this ratio is known and expressed as a number e_n, typically with $0 \leqslant e_n \leqslant 1$, for data cell n.

The adjustment for exposure is simple, namely, change the parameter representing the observed signal amplitude in the likelihood to what it would have been if the exposure had been unity. First compute the exposure e_n for data cell n. Then increase by the factor $1/e_n$ whatever quantity in the data cell represents the measured signal intensity. Specifically, for TTE data this parameter is the reciprocal of the interval of the corresponding data cell: $1/\Delta t_n$ (see Equation (20)), which is then replaced with $1/(e_n \Delta t_n)$. For bin counts the bin size can be multiplied by e_n or equivalently the count by $1/e_n$. For point measurements the amplitude measurement are multiplied by $1/e_n$ (and adjust any observational error parameters accordingly). In all cases the goal is to represent the data as closely as possible to what it would have been if the exposure had been constant. Of course this restoration is not exact in individual cases, but is correct on average.

For TTE data the fact that interval Δt_n as we define it in Equation (20) depends on the times of two different events (just previous to and just after the one under consideration) may seem to pose a problem. The exposures of these events will in general be different, so what value do we use for the given event? The comforting answer is that the only relevant exposure is that for the given event itself. In considering the interval from the previous to the current time, namely, $t_n - t_{n-1}$, t_{n-1} is regarded as simply a fiducial time and the distribution of this interval is given by Equation (B5) with λ the true rate adjusted by the exposure for event n, by the principle described in Appendix B.5 just after this equation. Similarly, by a time-reversal invariance argument, the distribution of the interval to the subsequent event, namely, $t_{n+1} - t_n$, also depends on only the same quantity. In summary, event independence (Appendix C) yields the somewhat counterintuitive fact that the probability distribution of $\Delta t_n = (t_{n+1} - t_{n-1})/2$ of the interval surrounding event n depends on only the effective event rate for event n.

1.9. Prior for the Number of Blocks

Earlier work (Scargle 1998) did not assign an explicit prior probability distribution for the number of blocks, i.e., the parameter $N_{\rm blocks}$. This omission amounts to using a flat prior, but in many contexts it is unreasonable to assign the same prior probability to all values. In particular, in most settings it is much more likely a priori that $N_{\rm blocks} \ll N$ than that $N_{\rm blocks} \approx N$. For this reason it is desirable to impose a prior that assigns smaller probability to a large number of blocks. We adopt this *geometric*[6] *prior* (Coram 2002) with the single parameter γ:

$$P(N_{\rm blocks}) = P_0 \gamma^{N_{\rm blocks}} \tag{3}$$

for $0 \leqslant N_{\rm blocks} \leqslant N$, and zero otherwise since $N_{\rm blocks}$ cannot be negative or larger than the number of data cells.

The normalization constant P_0 is easily obtained, giving

$$P(N_{\rm blocks}) = \frac{1-\gamma}{1-\gamma^{N+1}} \gamma^{N_{\rm blocks}}, \tag{4}$$

and the expected number of blocks is

$$\langle N_{\rm blocks} \rangle = P_0 \sum_{N_{\rm blocks}=0}^{N} N_{\rm blocks} \gamma^{N_{\rm blocks}} = \frac{N\gamma^{N+1}+1}{\gamma^{N+1}-1} + \frac{1}{1-\gamma}. \tag{5}$$

(Note that the estimated number of blocks is a discontinuous, monotonic function of γ, and because its jumps can be >1 it is not generally possible to force a prescribed number by adjusting this parameter.) See Coram & Lalley (2006) and references therein, such as Diaconis & Freedman (1993) and Diaconis & Freedman (1995), for discussions of the geometric and other priors with regard to overfitting and frequentist consistency in Bayesian regression, hierarchical priors, and priors directly on the time series representation itself (and not the number of blocks as done here).

[6] This name seems to arise from its relation to geometric series.

Other distributions might better represent known or assumed prior information in specific applications, or perhaps be useful as a generic prior for the kind of general purpose tools we present here, meant for a wide variety of applications. Three considerations drive our choice for the prior, namely, that it: (1) achieves the desideratum in the previous paragraph (by taking $\gamma < 1$), (2) has well-understood theoretical properties (Coram 2002; Coram & Lalley 2006), and (3) is simply implemented in the algorithm. This last point follows because the contribution of the prior to block fitness, given by the logarithm of Equation (3), can be implemented simply by adding the constant $\log \gamma$ (called `ncp_prior` in the MatLab™ code and in the discussion of computational issues below) to the fitness of each block. Values of γ larger than 1 almost certainly lead to extreme overfitting, namely, assigning each datum to a separate block. Determining the actual value to use in applications is discussed in Section 2.7 below.

Naturally, the prior influences the number of blocks in the optimal representation. This connection is of some importance since it means that the parameter γ affects the representation, its visual appearance, and the values of quantities derived from it. Accordingly, one can think of γ as a free parameter that can be varied to adjust the amount of structure in the block representation, controlling, e.g., its total variation. However note that the discontinuities at the block edges are not rounded in such an adjustment, which is therefore different from bandpass filtering, e.g., and more analogous to wavelet denoising (Donoho & Johnstone 1998) in the sense of retaining sharply defined structures supported by the data. Of course one may experiment with different values, as is usually done with smoothing parameters. However it is important to have an objective, principled way to select the value of this parameter—the subject of Section 2.7.

1.10. Related Work

Some references to related work are to be found in Jackson et al. (2005), especially papers by Hubert and Kehagias. More recent work includes (Mannila & Salmenkivi 2001; Du & Kou 2012; Xie et al. 2012).

2. OPTIMUM SEGMENTATION OF DATA ON AN INTERVAL

Piecewise constant modeling of sequential measurements on a time interval $\mathcal{T}$ is most conveniently implemented by seeking an *optimal partition* of the ordered set of data cells within $\mathcal{T}$. In this special case of segmentation, the segments cover the whole set with no overlap between them (Appendix B). Segmentations with overlap are possible, for example, in the case of correlated measurements, but are not considered here. One can envision our quest for the optimal segmentation as nothing more than finding the best step function, or piecewise constant model, fit to the data—defined by maximizing a specific fitness measure as detailed in Section 2.4.

We introduce our algorithm in a somewhat abstract setting because the formalism developed here applies to other data analysis problems beyond time series analysis. It implements Bayesian Blocks or other one-dimensional segmentation ideas for any model fitness function that satisfies a simple additivity condition. It improves the previous approximate segmentation algorithm (Scargle 1998) by achieving an exact, rigorous solution of the multiple change-point problem, guaranteed to be a global optimum, not just a local one.

The rest of this section describes the structure (Sections 2.1–2.3, 2.5), fitness (Section 2.4; details for specific data modes are in Section 3), and optimization (Section 2.6) of the model, as well as the complexity penalty parameter γ (Section 2.7), error analysis (Section 2.8), and multivariate data (Section 2.9).

2.1. Partitions

Partitions of a time interval $\mathcal{T}$ are simply collections of non-overlapping blocks (defined below in Section 2.3), defined by specifying the number of its blocks and the block edges:

$$\mathcal{P}(I) \equiv \{N_{\rm blocks};\ n_k,\ k = 1, 2, 3, \ldots N_{\rm blocks}\}, \qquad (6)$$

where the n_k are indices of the data cells (Section 2.2) defining times called *change points* (see Section 2.5).

Appendix B gives a few mathematical details about partitions, including justification of the restriction of the change points to coincide with data points and the result that the number of possible partitions (i.e., the number of ways N cells can be arranged in blocks) is 2^N. This number is exponentially large, rendering an explicit exhaustive search of partition space utterly impossible for all but very small N. Our algorithm implicitly performs a complete search of this space in time of order N^2, and is practical even for $N \sim 1{,}000{,}000$, for which approximately $10^{300{,}000}$ partitions are possible. The beauty of the algorithm is that it finds the optimum among all partitions without an exhaustive explicit search, which is obviously impossible for almost any value of N arising in practice.

2.2. Data Cells

For input to the algorithm the measurements are represented with *data cells*. For the most part there is one cell for each measurement, although in the case of TTE data two or more events with identical time tags may be combined into a single cell. A convenient data structure is an array containing the cells ordered by the measurement times.

Specification of the contents of the cells must meet two requirements. First, they must include time information allowing determination of which cells lie in a block given its start and stop times. Post-processing steps such as plotting the blocks may in addition use the actual times, either absolute or relative to a specified origin.

The other requirement is that the fitness of a block can be computed from the contents of all the cells in it (Sections 2.4 and 3). For the three standard cases the relevant data, roughly speaking, are: (1) intervals between events (Section 3.1); (2) bin sizes, locations, and counts (Section 3.2); and (3) measured values augmented by a quantifier of measurement uncertainty (Section 3.3). These same quantities enable construction of the resulting step function for post-processing steps such as computing signal parameters.

2.3. Blocks of Cells

A *block* is any set of consecutive cells, either an element of the optimal representation or a candidate for it. Each block represents a subinterval (within the full range of observation times) over which the amplitude of the signal can be estimated from the contents of its cells (Section 2.2). A block can be as small as one cell or as large as all of the cells.

Our time series model consists of a set of blocks partitioning the observations. All model parameters are constant within each block but undergo discrete jumps at the change points (Section 2.5) marking the edges of the blocks. The model is

visualized by plotting rectangles spanning the intervals covered by the blocks, each with height equal to the signal intensity averaged over the interval. The concept of *fitness of a block* is fundamental to everything else in this paper. As we will see in the next section the fitness of a partition is the sum of the fitnesses of the blocks comprising it.

2.4. Fitness of Blocks and Partitions

Since the goal is to represent the data as well as possible within a given class of models, we maximize a quantity measuring the fitness of models in the given class—here, the class of all piecewise constant models. Alternatively, one can minimize an error measure. Both operations are called *optimization*. The algorithm relies on the fitness being block-additive, i.e.,

$$F[\mathcal{P}(\mathcal{T})] = \sum_{k=1}^{N_{\rm blocks}} f(B_k), \tag{7}$$

where $F[\mathcal{P}(\mathcal{T})]$ is the total fitness of the partition $\mathcal{P}$ of interval $\mathcal{T}$ and $f(B_k)$ is the fitness of block k. The latter can be any convenient measure of how well a constant signal represents the data within the block. Typically, additivity results from independence of the observational errors. We here ignore the possibility of correlated errors, which could make the fitness of one block depend on that of its neighbors. Remember correlation of observational errors is quite separate from correlations in the signal itself.

All model parameters are marginalized except the n_k specifying block edges. Then the total fitness depends on only these remaining parameters—i.e., on the detailed specification of the partition by indicating which cells fall in each of its blocks. The best model is found by maximizing F over all possible such partitions.

2.5. Change Points

In the time series literature, a point at which a statistical model undergoes an abrupt transition, by one or more of its parameters jumping instantaneously to a new value, is called a *change point*. This is exactly what happens at the edges of the blocks in our model. In principle change points can be at arbitrary times. However, following the data cell representation and without any essential loss of generality, they can be restricted to coincide with a data point (Appendix B).

A few comments on notation are in order. We take blocks to start at the data cell identified by the algorithm as a change point and to end at the cell previous to the subsequent change point. A slight variation of this convention is discussed below in Section 4.4 in connection with allowing the possibility of empty blocks in the context of event data. One might adopt other conventions, such as apportioning the change-point data cell to both blocks, but we do not do so here. Even though the first data cell in the time series always starts the first block, our convention is that it is not considered a change point. In the code presented here the first change point marks the start of the second block. For $k > 1$ the kth block starts at index n_{k-1} and ends at $n_k - 1$. The first block always starts with the very first data cell. The last block always terminates with the very last data cell. If the last cell is a change point, it defines a block consisting of only that one cell. The set of change points is empty if the best model consists of a single block, meaning that the time series is sensibly constant over the whole observation interval. The number of blocks is one greater than the number of change points.

2.6. The Algorithm

We now outline the basic algorithm yielding the desired optimum partitions. The details of this *dynamic programming*[7] approach (Bellman 1961; Hubert et al. 2001; Dreyfus 2002) are in Jackson et al. (2005). It follows the spirit of mathematical induction: Beginning with the first data cell, at each step one more cell is added. The analysis makes use of results stored from all previous steps. Remarkably, the algorithm is exact and yields the optimal partition of an exponentially large space in time of order N^2. The iterations normally continue until the whole interval has been analyzed. However its recursive nature allows the algorithm to function in a *trigger mode*, halting when the first change point is detected (Section 4.3).

Let $\mathcal{P}^{\rm opt}(R)$ denote the optimal partition of the first R cells. In the starting case $R = 1$, the only possible partition (one block consisting of the first cell by itself) is trivially optimal. Now assume we have completed step R, identifying the optimal partition $\mathcal{P}^{\rm opt}(R)$. At this (and each previous) step, the value of optimal fitness itself is stored in array `best` and the location of the last change point of the optimal partition is stored in array `last`.

It remains to show how to obtain $\mathcal{P}^{\rm opt}(R+1)$. For some r consider the set of all partitions (of these first $R+1$ cells) whose last block starts with cell r (and by definition ends at $R+1$). Denote the fitness of this last block by $F(r)$. By the subpartition result in Appendix B the only member of this set that could possibly be optimal is that consisting of $\mathcal{P}^{\rm opt}(r-1)$ followed by this last block. By the additivity in Equation (7) the fitness of said partition is the sum of $F(r)$ and the fitness of $\mathcal{P}^{\rm opt}(r-1)$ saved from a previous step:

$$A(r) = F(r) + \begin{cases} 0 & r = 1 \\ \texttt{best}(r-1), & r = 2, 3, \ldots, R+1. \end{cases} \tag{8}$$

$A(1)$ is the special case where the last block comprises the entire data array and thus no previous fitness value is needed. Over the indicated range of r this equation expresses the fitness of all partitions $\mathcal{P}(R+1)$ that can possibly be optimal. Hence the value of r yielding the optimal partition $\mathcal{P}^{\rm opt}(R+1)$ is the easily computed value maximizing $A(r)$:

$$r^{\rm opt} = \text{argmax}[A(r)]. \tag{9}$$

At the end of this computation, when $R = N$, it only remains to find the locations of the change points of the optimal partition. The needed information is contained in the array `last` in which we have stored the index $r^{\rm opt}$ at each step. Using the corollary in Appendix B it is a simple matter to use the last value in this array to determine the last change point in $P^{\rm opt}(N)$, peel off the end section of `last` corresponding to this last block, and repeat. That is to say, the set of values

$$cp_1 = \texttt{last}(N);\ cp_2 = \texttt{last}(cp_1 - 1);\ cp_3 = \texttt{last}(cp_2 - 1);\ \ldots \tag{10}$$

[7] Bellman's explanation of his choice of this name, before the word "programming" took on its current computational connotation, is interesting. The Secretary of Defense at the time "... had a pathological fear and hatred of the word, research.... You can imagine how he felt, then, about the term, mathematical.... I felt I had to do something to shield... the Air Force from the fact that I was really doing mathematics inside the RAND Corporation.... I was interested in planning... But planning is not a good word for various reasons. I decided therefore to use the word, programming. I wanted to get across the idea that this was dynamic... it's impossible to use the word, dynamic, in a pejorative sense. Try thinking of some combination that will possibly give it a pejorative meaning. It's impossible. Thus, I thought dynamic programming was a good name. It was something not even a Congressman could object to (Dreyfus 2002, pg. 25)."

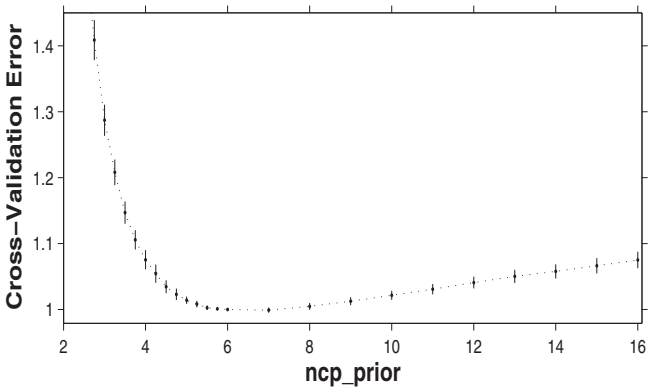

Figure 1. Cross-validation error of BATSE TTE data (averaged over 532 GRBs, eight random subsamples, and time) for a range of values of $-\log \gamma =$ with 3σ error bars.

are the index values giving the locations of the change points, in reverse order. Note that the positions of the change points are not necessarily fixed until the very last iteration, although in practice it turns out that they become more or less "frozen" once a few succeeding change points have been detected. MatLab™ (The Mathworks, Inc.) code for optimal partitioning of event data is given in Appendix A and included as supplementary data files.

2.7. Fixing the Parameter in the Prior Distribution for $N_{\rm blocks}$

As mentioned in Section 1.9, the output of the algorithm is dependent on value of the parameter γ, characterizing the assumed prior distribution for the number of blocks, Equation (3). In many applications the results are rather insensitive to the value as long as the signal-to-noise ratio is even moderately large. Nevertheless extreme values of this parameter give bad results in the form of clearly too few or too many blocks. In any case one must select a value to use in applications.

This situation is much like that of selecting a smoothing parameter in various data analysis applications, e.g., density estimation. In such contexts there is no perfect choice but instead a tradeoff between bias and variance. Here the tradeoff is between a conservative choice not fooled by noise fluctuations but potentially missing real changes, and a liberal choice better capturing changes but yielding some false detections. Several approaches have proven useful in elucidating this tradeoff. Merely running the algorithm with a few different values can indicate a range over which the block representation is reasonable and not very sensitive to the parameter value (cf. Figure 1).

The discussion of fitness functions below in Section 3 gives implementation details of an objective method for calibrating `ncp_prior` as a function of the number of data points. The procedure uses the fact that this parameter controls the *false positive* rate—i.e., the probability p_1 of falsely reporting detection of a change point. That is p_1 is defined to be the relative frequency with which the algorithm reports the presence of a change point in data with no signal present. It is convenient to use the complementary quantity

$$p_0 \equiv 1 - p_1, \tag{11}$$

the frequency with which the algorithm correctly rejects the presence of a change point in such data by returning a null list of change-point times. Therefore it is also the probability that a change point reported by the algorithm with this value of `ncp_prior` is indeed statistically significant—hence we call it the *correct detection rate* for single change points.

The needed `ncp_prior`–p_0 relationship is easily found by noting that the rates of correct and incorrect responses to fluctuations in simulated pure noise can be controlled by adjusting the value of `ncp_prior`. The procedure is: generate a synthetic pure noise time series; apply the algorithm for a range of `ncp_prior`; and select the smallest value that yields false detection frequency equal or less than the desired rate, such as 0.05. The values of `ncp_prior` determined in this way are averaged over a large number of realizations of the random data. The result depends on only the number of data points and the adopted value of p_0:

$$\text{ncp\_prior} = \psi(N, p_0). \tag{12}$$

Results from simulations of this kind are given below for the various fitness functions in Sections 3.1–3.3. We have no exact formulas, but rather fits to these numerical simulations.

The above discussion is useful in the simple problem of deciding whether or not a signal is present above a background of noisy observations. In other words, we have a procedure for assigning a value of `ncp_prior` that results in an acceptable frequency of spurious change points, or false positives, when searching for a single statistically significant change. Real-time triggering on transients (Section 4.3) is an example of this situation, as is any case where detection of a single change point is the only issue in play.

But elucidating the shape of an actual detected signal lies outside the scope of the above procedure, since it is based on a pure noise model. A more general goal is to limit the number of both false negatives and false positives in the context of an extended signal. The choice of the parameter value here depends on the nature of the signal present and the signal-to-noise level. One expects that somewhat larger values of `ncp_prior` are necessary to guard against corruption of the estimate of the signal's shape due to errors at multiple change points.

This idea suggests a simple extension of the above procedure. Assume that a value of p_0, the probability of correct detection of an *individual* change point, has been adopted and the corresponding value of `ncp_prior` determined with pure noise simulations as outlined above and expressed in Equation (12). For a complex signal our goal is correct detection of not just one, but several change points, say $N_{\rm cp}$ in number. The trick is to treat each of them as an independent detection of a single change point with success rate p_0. The probability of all $N_{\rm cp}$ successes follows from the law of compound probabilities:

$$p(N_{\rm cp}) = p_0^{N_{\rm cp}}. \tag{13}$$

There are problems with this analysis in that the following are not true.

1. Change-point detection in pure noise and in a signal are the same.
2. The detections are independent of each other.
3. We know the value of $N_{\rm cp}$.

All of these statements would have to be true for Equation (13) to be rigorously valid. We propose to regard the first two as approximately true and address the third as follows. Decide that the probability of correctly detecting all the change points should be at least as high as some value p_*, such as 0.95. Apply the algorithm using the value of `ncp_prior` $= \psi(N, p_*)$ given

by the pure noise simulation. Use Equation (13) and the number of change points thus found to yield a revised value

$$\mathrm{ncp\_prior} = \psi(N, p_*^{1/N_{\mathrm{cp}}}). \quad (14)$$

Stopping when the iteration produces no further modification of the set of change points, one has the recommended value of `ncp_prior`. This ad hoc procedure is not rigorous, but it establishes a kind of consistency and has proven useful in all the cases where we have tried it (e.g., Norris et al. 2010, 2011).

Figure 1 shows another approach, based on cross-validation of the data being analyzed. See Arlot & Celisse (2010) for a recent review of this procedure, and Hogg (2008) for an application in a context similar to that here. This study uses the collection of raw TTE data at the Burst and Transient Source Experiment (BATSE) Web site ftp://legacy.gsfc.nasa.gov/compton/data/batse/ascii_data/batse_tte/. The files for each of 532 GRBs contain time tags for all photons detected for that burst. The energy and detector tags in the data files were not used here, but Section 4.1 shows an example using the former. An ordinary 256-bin histogram of all photon times for each of 532 GRBs was taken as the true signal for that burst. Eight random subsamples smaller by a factor of eight were analyzed with the algorithm using the fitness in Equation (19). The average rms error between these block representations (evaluated at the same 256 time points) and the histogram is roughly flat over a broad range. While this illustration with a relatively homogeneous data set should obviously not be taken as universal, the general behavior seen here—determination of a broad range of nearly equally optimal values of `ncp_prior`—is characteristic of a wide variety of situations.

2.8. Analysis of Variance

Assessment of uncertainty is an important part of any data analysis procedure. The observational errors discussed throughout this paper are propagated by the algorithm to yield corresponding uncertainties in the block representation and its parameters. The propagation of stochastic variability in the astronomical source is a separate issue, called *cosmic variance*, and is not discussed here.

Since the results here comprise a complete function defined by a variable number of parameters, quantification of uncertainty is considerably more intricate than for a single parameter. In particular, one must specify precisely which of the block representation's aspects is at issue. Here we discuss three aspects: (1) the full block representation, (2) the very presence of the change points themselves, and (3) locations of change points.

A straightforward way to deal with (1) is by bootstrap analysis. As described in Efron & Tibshirani (1998), for time series data this procedure is rather complicated in general. However, resampling of event data in the manner appropriate to the bootstrap is trivial. The procedure is to run the algorithm on each of many bootstrap samples and evaluate the resulting block representations at a common set of evenly spaced times. In this way, models with different numbers and locations of change points can be added, yielding means and variances for the estimated block light curves. The bootstrap variance is an indicator of light-curve uncertainty. In addition, comparison of the bootstrap mean with the block representation from the actual data adds information about modeling bias. The former is rather like a model average in the Bayesian context. This average typically smooths out the discontinuous block edges present in any one representation. In some applications the bootstrap mean may be more useful than the block representation.

This method does not seem to be useful for studying uncertainty in the change points themselves, in particular their number, presumably because the duplication of data points due to the replacement feature of the resampling yields excess blocks (but with random locations and small amplitude variance, and therefore with little effect on the mean light curve).

Issue (2) refers to an assessment of the statistical significance of the identification of a given change point. For a given change point we suggest quantification of this uncertainty by evaluating the ratio of the fitness functions for the two blocks on either side of that change point to that of the single block that would exist if the change point were not there. The corresponding difference of the (logarithmic) fitness values should be adjusted by the value of the constant parameter `ncp_prior`, for consistency with the way fitness is computed in the algorithm.

Finally, (3) is easily addressed in an approximate way by fixing all but one change point and computing fitness as a function of the location of that change point. This assessment is approximate because it neglects inter-change-point dependences. One then converts the run of the fitness function with change-point location into a normalized probability distribution, giving comprehensive statistical information about that location (mean, variance, confidence interval, etc.)

Sample results of all of these uncertainty measures in connection with analysis of a GRB light curve are shown below in Section 4.1 and Figure 8.

2.9. Multivariate Time Series

Our algorithm's intentionally flexible data interface not only allows the processing of a wide variety of data modes, but also facilitates joint analysis of mode combinations. This feature allows one to obtain the optimal block representation of several concurrent data streams with arbitrary modes and sample times. This analysis is joint in the sense that the change points are constrained to be at the same times for all the input series; in other words, the block edges for all of the input data series line up. The representation is optimal for the data as a whole but not for the individual time series.

To interpret the result of a multivariate analysis one can study the blocks in the different series in two ways: (1) separately, but with the realization that the locations of their edges are determined by all the data; or (2) in a combined representation. The latter requires that there be a meaningful way to combine amplitudes. For example, the plot of a joint analysis of event and binned data could simply display the combined event rate for each block, perhaps adjusting for exposure differences. For other modes, such as photon events and radio frequency fluxes, a joint display would have to involve a spectral model or some sort of relative normalization. The example in Section 4.2 below will help clarify these issues.

The idea extending the basic algorithm to incorporate multiple time series is simple. Each datum in any mode has a time tag associated with it—for example, the event time, the time of a bin center, or the time of a point measurement. The joint change points are allowed to occur at any one of these times. Hence, the times from all of the separate data streams are collected together into a single ordered array; the ordering means that the times—as well as the measurement data—from the different modes are interleaved. The schematic in Figure 2 shows how the individual concatenated times and data series are placed in separate blocks in a matrix (top) and then redistributed (bottom)

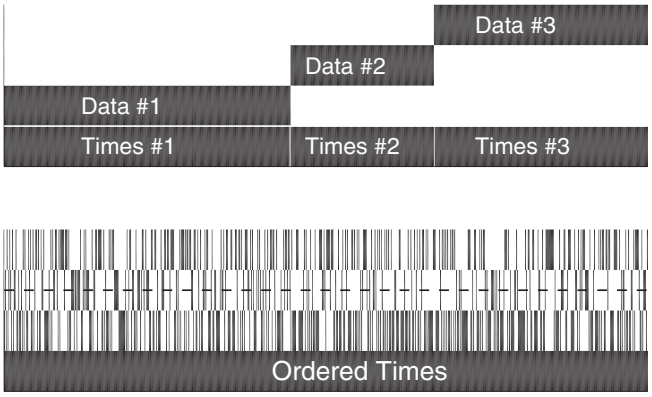

Figure 2. Schematic depicting an example of how three data series are first concatenated into a matrix (top) and then redistributed by ordering the combined time tags (bottom). The cost functions for the series can then be computed from the data in horizontal slices (e.g., dashed line) and combined, allowing the change points to be at any of the time tags.

by ordering the combined times. Then the fitness function for a given data series can be obtained from the corresponding data slice (e.g., the horizontal dashed line in the figure, for Series #2). The zero entries in these slices (indicated by white space in the figure) are such that the fitness function for data from each series is evaluated for only the appropriate data and mode combination. The overall fitness is then simply the sum of those for the several data series. The details of this procedure are described in the code provided in Appendix A.

2.10. Comparison with Theoretical Optimal Detection Efficiency

How good is the algorithm at extracting weak signals in noisy data? This section gives evidence that it achieves detection sensitivity closely approaching ideal theoretical limits. The formalism in Arias-Castro et al. (2005) treats detection of geometric objects in data spaces of arbitrary dimension using multiscale methods. The one-dimensional special case in Section II of this reference is essentially equivalent to our problem of detecting a single block in noisy time series.

Given N measurements normalized so that the observational errors $\sim N(0, \sigma)$ (normally distributed with zero mean and variance σ^2), these authors show that the threshold for detection is

$$A_1 = \sigma \sqrt{2 \log N}. \quad (15)$$

This result is asymptotic (i.e., valid in the limit of large N). It is valid for a frequentist detection strategy based on testing whether the maximum of the inner product of the model with the data exceeds the quantity in Equation (15) or not. These authors state "In short, we can efficiently and reliably detect intervals of amplitude roughly $\sqrt{2 \log N}$, but not smaller" (Arias-Castro et al. 2005, pg. 2406). More formally the result is that asymptotically their test is powerful for signals of amplitude greater than A_1 and powerless for weaker signals.

It is of interest to see how well our algorithm stacks up against these theoretical results, since the two analysis approaches (matched filter test statistic versus Bayesian model selection) are fundamentally different. Consider a simulation consisting of normally distributed measurements at arbitrary times in an interval. These variates are taken to be zero mean normal, except over an unknown subinterval where the mean is a fixed constant. In this experiment the events are evenly spaced, but only their order matters, so the results would be the same for arbitrary

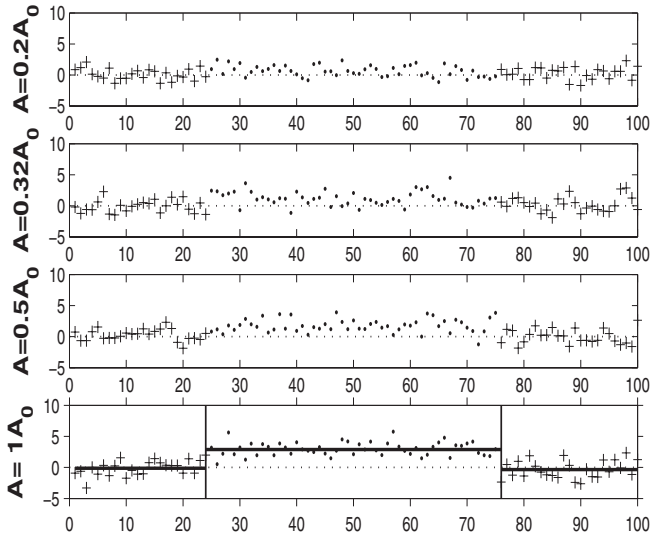

Figure 3. One hundred unit variance normally distributed measurements—zero mean (+) except for a block of events 25–75 (dots). In the four panels the block amplitudes are 0.2, 0.32, 0.5, and 1.0 in units of the Arias-Castro et al. (2005) threshold $\sqrt{2 \log N}$. Thick lines show the blocks, where detected, with thin vertical lines at the change points.

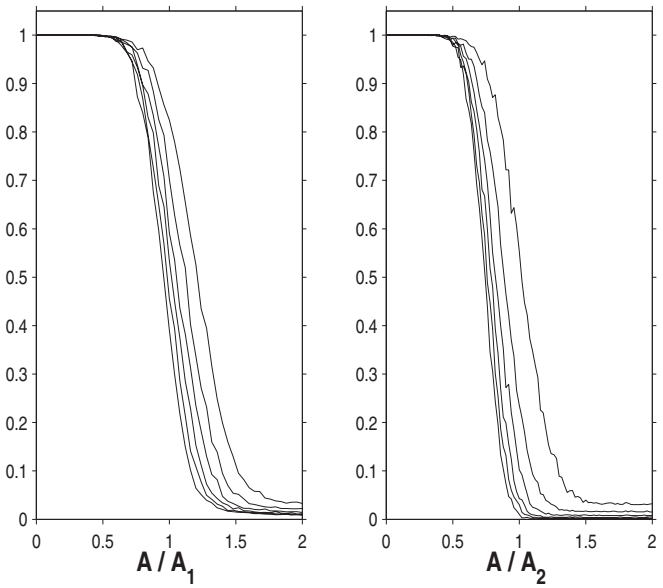

Figure 4. Error in finding a single block vs. simulated block amplitude in units of Arias-Castro et al.'s (2005) threshold amplitude. The curves (from right to left) are for $N = 32$, 64, 128, 256, 1024, and 2048.

spacing of the events. Figure 3 shows synthetic data for four simulated realizations with different values for this constant. The solid line is the Bayesian Blocks representation, using the posterior in Equation (C50). For the small amplitudes in the first panels no change points are found; these weak signals are completely missed. In the last panel the signal is detected and correctly represented.

Figure 4 reports some results of detection of the same step-function process shown in Figure 3, averaged over many different realizations of the observational error process and for several different values of N. The lines are plots of a simple error metric (combing the errors in the number of change points and their locations) as a function of the amplitude of the test signal. The left panel is for the case where the number of points

in the putative block is held fixed, whereas the right panel is for the cases where this number taken to be proportional to N, sometimes a more realistic situation. We have adopted the following definition for the threshold in this case:

$$A_2 = 11.3\sigma \sqrt{\frac{\log N}{N}}. \quad (16)$$

This formula is roughly consistent with the asymptotic result for this case, namely, $A \gtrsim \sigma/\sqrt{N}$ (E. Arias-Castro 2012, private communication) with an arbitrary factor for plotting purposes.

Our method achieves small errors when the signal amplitude is on the order or even somewhat smaller than the limit stated by Arias-Castro et al. (2005), showing that we are indeed close to their theoretical limit. The main difference here is that our results are for specific values of N and the theoretical results are asymptotic in N.

3. BLOCK FITNESS FUNCTIONS

To complete the algorithm, all that remains is to define the *model fitness function* appropriate to a particular data mode. By Equation (7) it is sufficient to define a *block fitness function*, which can be any convenient measure of how well a constant signal represents the data in the block. Naturally, this measure will depend on all data in the block and not on any outside it. As explained in Section 2.4, it cannot depend on any model parameters other than those specifying the locations of the block edges. In practice this means that block height (signal amplitude) must somehow be eliminated as a parameter. This can be accomplished, for example, by taking block fitness to be the relevant likelihood either maximized or marginalized with respect to this parameter. Either choice yields a quantity good for comparing alternative models, but not necessarily for assessing goodness of fit of a single model. Note that these measures as such do not satisfy the additivity condition Equation (7). As long as the cell measurement errors are independent of each other, the likelihood of a string of blocks is the product of the individual values, but not the required sum. But simply taking the logarithm yields the necessary additivity.

There is considerable freedom in choosing fitness functions to be used for a given type of data. The examples described here have proven useful in various circumstances, but the reader is encouraged to explore other block-additive functions that might be more appropriate in a given application. For all cases considered in this paper the fitness function depends on data in the block through summary parameters called *sufficient statistics*, capturing the statistical behavior of the data. If these parameters are sums of quantities defined on the cells, the computations are simplified; however, this condition is not essential.

Two types of factors in the block fitness can be ignored. A constant factor C appearing in the likelihood *for each data cell* yields an overall constant term in the derived logarithmic fitness function for the whole time series, namely, $N \log C$. Such a term is independent of all model parameters and therefore irrelevant for the model comparison in the optimization algorithm. In addition, while a term in the block fitness that has the same value *for each block* does affect total model fitness, it contributes a term proportional to the number of blocks, and which therefore can be absorbed into the parameter derived from the prior on the number of blocks (cf. Section 1.9).

Many of the data modes discussed in the following subsections were operative in the BATSE experiment on the NASA *Compton Gamma Ray Observatory*, the *Swift* Gamma-Ray Burst Mission, the *Fermi Gamma-ray Space Telescope*, and many X-ray and other high-energy observatories. They are also relevant in a wide range of other applications.

In the rest of this section we exhibit expressions that serve as practical and reliable fitness functions for the three most common data modes: event data, binned data, and point measurements with normal errors. In each case rules for selection of the value of `ncp_prior` (cf. Section 1.9) are also provided. Some refinements of this discussion and some other less common data modes are discussed in Appendix C.

3.1. Event Data

For series of times of discrete events it is natural to associate one data cell (Section 2.2) with each event. The following derivation of the appropriate block fitness will elucidate exactly what information the cells must contain to allow evaluation of the fitness for the full multi-block model.

In practice the event times are integer multiples of some small unit (Appendix C.1) but it is often convenient to treat them as real numbers on a continuum. For example, the fitness function is easily obtained starting with the unbinned likelihood known as the Cash statistic (Cash 1979); a thorough discussion is in Tompkins (1999). If $M(t, \theta)$ is a model of the time dependence of a signal the unbinned log-likelihood is

$$\log L(\theta) = \sum_n \log M(t_n, \theta) \; - \int M(t, \theta) dt, \quad (17)$$

where the sum is over the events and θ represents the model parameters. The integral is over the observation interval and is the expected number of events under the model. Our block model is constant with a single parameter, $M(t, \lambda) = \lambda$, so for block k

$$\log L^{(k)}(\lambda) = N^{(k)} \log \lambda \; - \lambda T^{(k)}, \quad (18)$$

where $N^{(k)}$ is the number of events in block k and $T^{(k)}$ is the length of the block. The maximum of this likelihood is at $\lambda = N^{(k)}/T^{(k)}$, yielding

$$\log L^{(k)}_{\max} + N^{(k)} = N^{(k)}(\log N^{(k)} - \log T^{(k)}). \quad (19)$$

The term $N^{(k)}$ is taken to the left side because its sum over the blocks is a constant (N, the total number of events) that is model-independent and therefore irrelevant. Moreover note that changing the units of time, say by a scale factor α, changes the log-likelihood by $-N^{(k)} \log(\alpha)$, irrelevant for the same reason. This felicitous property holds for other maximum likelihood fitness functions and removes what would otherwise be a parameter of the optimization. This effective scale invariance and the simplicity of Equation (19) make its block sum the fitness function of choice to find the optimum block representation of event data. A possible exception is the case where detection of more than one event at a given time is not possible, e.g., due to detector, *dead time*, in which case the fitness function in Appendices C and C.2 may be more appropriate.

It is now obvious what information a cell must contain to allow evaluation of the sufficient statistics $N^{(k)}$ and $T^{(k)}$ by summing two quantities over the cells in a block. First, it must contain the number of events in the cell. (This is typically one, but can be more depending on how duplicate time tags are handled; see the code section in Appendix A, dealing with duplicate time tags,

or ones that are so close that it makes sense to treat them as identical.) Second, it must contain the interval

$$\Delta t_n = (t_{n+1} - t_{n-1})/2, \tag{20}$$

representing the contribution of cell n to the length of the block. This interval contains all times closer to event n than to any other. It is defined by the midpoints between successive events, and generalizes to data spaces of any dimension, where it is called the *Voronoi tessellation* of the data points (Okabe et al. 2000; Scargle 2001a, 2001b). Because $1/\Delta t_n$ can be regarded as an estimate of the local event rate at time t_n, it is natural to visualize the corresponding data cell as the unit-area rectangle of width Δt_n and height $1/\Delta t_n$. These ideas lead to the comment in Section 1.8 that the event-by-event adjustment for exposure can be implemented by shrinking Δt_n by the exposure factor e_n.

It is interesting to note that the actual locations of the (independent) events within their block do not matter. The fitness function depends on only the number of events in the block, not their locations or the intervals between them. This result flows directly from the nature of the underlying independently distributed, or Poisson, process (see Appendix B).

We conclude this section with evaluation of the calibration of `ncp_prior` from simulations of signal-free observational noise as described in Section 2.7. The results of extensive simulations for a range of values of N and the adopted false positive rate p_0 introduced in Equation (11) were found to be well fit with the formula

$$\text{ncp\_prior} = 4 - 73.53 p_0 N^{-0.478}. \tag{21}$$

For example, with $p_0 = 0.01$ and $N = 1000$ this formula gives `ncp_prior` $= 7.61$.

3.2. Binned Event Data

The expected count in a bin is the product $\lambda e W$ of the true event rate λ at the detector, a dimensionless exposure factor e (Section 1.8), and the width of the bin W. Therefore the likelihood for bin n is given by the Poisson distribution

$$L_n = \frac{(\lambda e_n W_n)^{N_n} e^{-\lambda e_n W_n}}{N_n!}, \tag{22}$$

where N_n is the number of events in bin n, λ is the actual event rate in counts per unit time, e_n is the exposure averaged over the bin, and W_n is the bin width in time units. Defining *bin efficiency* as $w_n \equiv e_n W_n$, the likelihood for block k is the product of the likelihoods of all its bins:

$$L^{(k)} = \prod_{n=1}^{M^{(k)}} L_n = \lambda^{N^{(k)}} e^{-\lambda w^{(k)}}. \tag{23}$$

Here $M^{(k)}$ is the number of bins in block k,

$$w^{(k)} = \sum_{n=1}^{M^{(k)}} w_n \tag{24}$$

is the sum of the bin efficiencies in the block, and

$$N^{(k)} = \sum_{n=1}^{M^{(k)}} N_n \tag{25}$$

is the total event count in the block. The factor $(e_n W_n)^{N_n}/N_n!$ has been discarded because its product over all the bins in all the blocks is a constant (depending on the data only) and therefore irrelevant to model fitness. The log-likelihood is

$$\log L^{(k)} = N^{(k)} \log\lambda - \lambda w^{(k)}, \tag{26}$$

identical to Equation (18) with $w^{(k)}$ playing the role of $T^{(k)}$, a natural association since it is an effective block duration. Moreover in retrospect it is understandable that unbinned and binned event data have the same fitness function, especially in view of the analysis in Appendix C.1 where ticks are allowed to contain more than one event and are thus equivalent to bins. In addition, the way variable exposure is treated here could just as well have been applied to event data in the previous section. Note that in all of the above, the bins are not assumed to be equal or contiguous—there can be arbitrary gaps between them (Section 1.7).

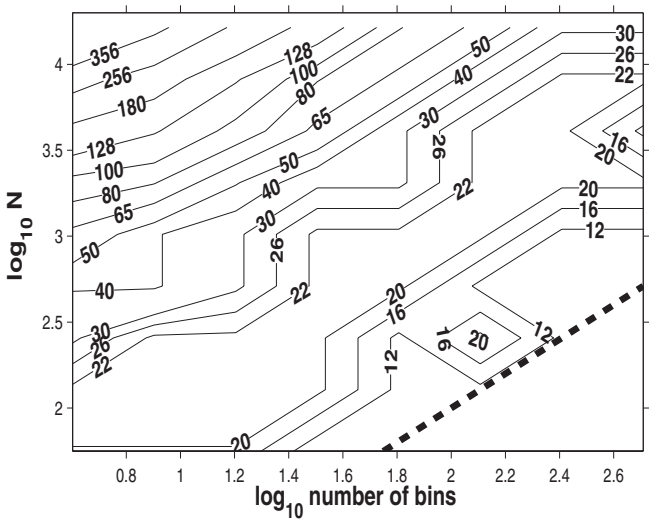

Figure 5. Simulation study, based on the false positive rate of 0.05, to determine ncp_prior $= -\log(\gamma)$ for binned data. Contours of this parameter are shown as a function of the number of bins and number of data points (logarithmic x- and y-axes, respectively). The heavy dashed line indicates the undesirable region where the numbers of bins and data points are equal.

We now turn to the determination of ncp_prior for binned data. Figure 5 is a contour plot of the values of this parameter based on a simulation study with bins containing independently distributed events. These contours are very insensitive to the false positive rate, which was taken as 0.05 in this figure.

3.3. Point Measurements

A common experimental scenario is to measure a signal $s(t)$ at a sequence of times t_n, $n = 1, 2, \ldots, N$ in order to characterize its time dependence. Inevitable corruption due to observational errors is frequently countered by smoothing the data and/or fitting a model. As with the other data modes Bayesian Blocks is a different approach to this issue, making use of knowledge of the observational error distribution and avoiding the information loss entailed by smoothing. In our treatment the set of observation times t_n, collectively known as the *sampling*, can be anything—evenly spaced points or otherwise. Furthermore we explicitly assume that the measurements at these times are independent of each other, which is to say the errors of observation are statistically independent.

Typically these errors are random and additive, so that the observed time series can be modeled as

$$x_n \equiv x(t_n) = s(t_n) + z_n \quad n = 1, 2, \ldots N. \tag{27}$$

The observational error z_n, at time t_n, is known only through its statistical distribution. Consider the case where the errors are taken to obey a normal probability distribution with zero mean and given variance:

$$P(z_n)dz_n = \frac{1}{\sigma_n\sqrt{2\pi}}\ e^{-\frac{1}{2}(\frac{z_n}{\sigma_n})^2}dz_n. \quad (28)$$

Using Equations (27) and (28), if the model signal is the constant $s = \lambda$, the likelihood of measurement n is

$$L_n = \frac{1}{\sigma_n\sqrt{2\pi}}\ e^{-\frac{1}{2}(\frac{x_n-\lambda}{\sigma_n})^2}. \quad (29)$$

Since we assume independence of the measurements the block k likelihood is

$$L^{(k)} = \prod_n L_n = \frac{(2\pi)^{-\frac{N_k}{2}}}{\prod_m \sigma_m}\ e^{-\frac{1}{2}\sum_n(\frac{x_n-\lambda}{\sigma_n})^2}. \quad (30)$$

Both the products and sum are over those values of the index such that t lies in block k. The quantities multiplying the exponentials in both the above equations are irrelevant because they contribute an overall constant factor to the total likelihood.

We now derive the maximum likelihood fitness function for this data mode (with other forms based on different priors relegated to Appendices C, C.4, C.5, C.6, and C.7). The quantities,

$$a_k = \frac{1}{2}\sum_n \frac{1}{\sigma_n^2} \quad (31)$$

$$b_k = -\sum_n \frac{x_n}{\sigma_n^2} \quad (32)$$

$$c_k = \frac{1}{2}\sum_n \frac{x_n^2}{\sigma_n^2}, \quad (33)$$

appear in all versions of these fitness functions; the first two are sufficient statistics.

As usual we need to remove the dependence of Equation (30) on the parameter λ, and here we accomplish this result by finding the value of λ which maximizes the block likelihood, that is by maximizing

$$-\frac{1}{2}\sum_n \left(\frac{x_n-\lambda}{\sigma_n}\right)^2. \quad (34)$$

This is easily found to be

$$\lambda_{\max} = \sum_n \frac{x_n}{\sigma_n^2} \Big/ \sum_{n'} \frac{1}{\sigma_{n'}^2} \quad (35)$$

$$= -b_k/2a_k. \quad (36)$$

As expected, this maximum likelihood amplitude is just the weighted mean value of the observations x_n within the block, because defining the weights,

$$w_n = \frac{\frac{1}{\sigma_n^2}}{\sum_{n'}\left(\frac{1}{\sigma_{n'}^2}\right)}, \quad (37)$$

yields

$$\lambda_{\max} = \sum_n w_n x_n. \quad (38)$$

Inserting Equation (36) into the log of Equation (30) with the irrelevant factors omitted yields the corresponding maximum value of the log-likelihood itself:

$$\log L_{\max}^{(k)} = -\frac{1}{2}\sum_n \left(\frac{x_n + \frac{b_k}{2a_k}}{\sigma_n}\right)^2, \quad (39)$$

where again the sums are over the data in block k. Expanding the square

$$\log L_{\max}^{(k)} = -\frac{1}{2}\left[\sum_n \frac{x_n^2}{\sigma_n^2} + \frac{b_k}{a_k}\sum_n \frac{x_n}{\sigma_n^2} + \frac{b_k^2}{4a_k^2}\sum_n \frac{1}{\sigma_n^2}\right], \quad (40)$$

dropping the first term (quadratic in x), which also sums to a model-independent constant, and using Equations (31) and (32) we arrive at

$$\log L_{\max}^{(k)} = b_k^2/4a_k. \quad (41)$$

As expected each data cell must contain x_n and σ_n but we now see that these quantities enter the fitness function through the summands in Equations (31) and (32) defining a_k and b_k (c_k does not matter), namely, $1/(2\sigma_n^2)$ and $-x_n/\sigma_n^2$. The way the corresponding block summations are implemented is described in Appendix A (cf. data mode #3).

A few additional notes may be helpful. In the familiar case in which the error variance is assumed to be time-independent, σ can be carried as an overall constant and σ_n does not have to be specified in each data cell. The t_n are only relevant in determining which cells belong in a block and do not enter the fitness computation explicitly. And the fitness function in Equation (41) is manifestly invariant to a scale change in the measured quantity, as is the alternative form derived in Appendix C, Equation (C42). That is to say, under the transformation

$$x_n \to a x_n, \quad \sigma_n \to a\sigma_n, \quad (42)$$

corresponding, for example, to a simple change in the units of x and σ, the fitness does not change.

Figure 6 exhibits a simulation study to calibrate ncp_prior for normally distributed point measurements. For illustration the pure noise data simulated were normally distributed with a mean of 10 and unit variance. The left-hand panel shows how the false positive rate is diminished as ncp_prior is increased, for the eight values of N listed in the caption. The horizontal line is at the adopted false positive rate of 0.05; the points at which these curves cross below this line generate the curve shown in the bottom panel. The linear fit in the latter depicts the relation ncp_prior $= 1.32 + 0.577\ \log_{10}(N)$. This relation is insensitive to the signal-to-noise ratio in the simulations.

4. EXAMPLES

The following subsections present illustrative examples with sample data sets, demonstrating block representation for TTE data, multivariate time series, triggering, the empty block problem for TTE data, and data on the circle.

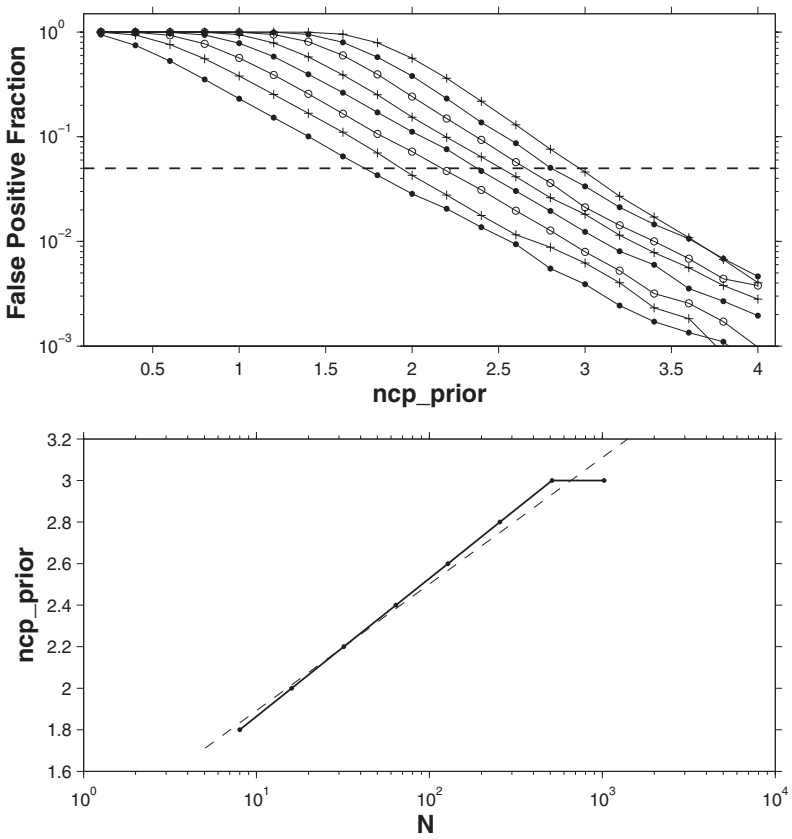

Figure 6. Simulations of point measurements (Gaussian noise with signal-to-noise ratio of 10) to determine ncp_prior $= -\log(\gamma)$. Top: false positive fraction p_0 vs. value of ncp_prior with separate curves for the values $N = 8$, 16, 32, 64, 128, 256, 512, and 1024 (left to right; alternating dots, + and circles). The points at which the rate becomes unacceptable (here 0.05; dashed line) determines the recommended values of ncp_prior shown as a function of N in the bottom panel.

4.1. BATSE Gamma-ray Burst TTE Data

Trigger 551 in the BATSE catalog (4B catalog name 910718) was chosen to exemplify analysis of TTE data, as it has moderate pulse structure. See Section 2.7 for a description of the data source. Figure 7 shows analysis of all of the event data in the top panels, and separated into the four energy channels in the lower panels. On the left are optimal block representations and the right shows the corresponding data in 32 evenly spaced bins.

In all five cases the optimal block representations based on the block fitness function for event data in Equation (19) are depicted for two cases, using values the values of `ncp_prior`: (1) from Equation (21) with $p_0 = 0.05$ (solid lines); and (2) found with the iterative scheme described in Section 2.7 (lightly shaded blocks bounded by dashed lines). These two results are identical for all cases except Channel 3, where the iterative scheme's more conservative control of false positives yields fewer blocks (9 instead of 13).

Note that the ordinary histograms of the photon times in the right-hand panels leave considerable uncertainty as to what the significant and true structures are. In the optimal block representations, two salient conclusions are clear: (1) There are three pulses and (2) they are most clearly delineated at higher energies.

Figure 8 depicts the error analysis procedures described above in Section 2.8, applied to one channel of these data.

4.2. Multivariate Time Series

This example in Figure 9 demonstrates the multivariate capability of Bayesian Blocks by analyzing data consisting of three different modes sampled randomly from a synthetic signal. TTEs, binned data, and normally distributed measurements were independently drawn from the same signal and analyzed separately, yielding the block representations depicted with thin lines.

The joint analysis of the data combined using the multivariate feature described above in Section 2.9 is represented as the thick dashed line. None of these analyses are perfect, of course, due to the statistical fluctuations in the data. The combined analysis finds a few spurious change points, but overall these do not represent serious distortions of the true signal. The individual analyses are somewhat poorer at capturing only the true change points. Hence, in this example the combined analysis makes effective use of disparate data modes from the same signal.

4.3. Real-time Analysis: Triggers

Because of its incremental structure, our algorithm is well suited for real-time analysis. Starting with a small amount of data, the algorithm typically finds no change points at first. Then, by determining the optimal partition up to and including the most recently added data cell, the algorithm effectively tests for the presence of the first change point. The real-time mode can be selected simply by triggering on the condition `last`$(R) > 1$ inserted into the code shown in Appendix A, just before the end of the basic iterative loop on R. For the entry of 1 in an element of array `last` means that the optimal partition consists of the whole array encountered so far. It is thus obvious that this first indication of change point cannot yield more than one change point.

Thus the algorithm can be set to return at the first significant change point. Other more complicated halting or return

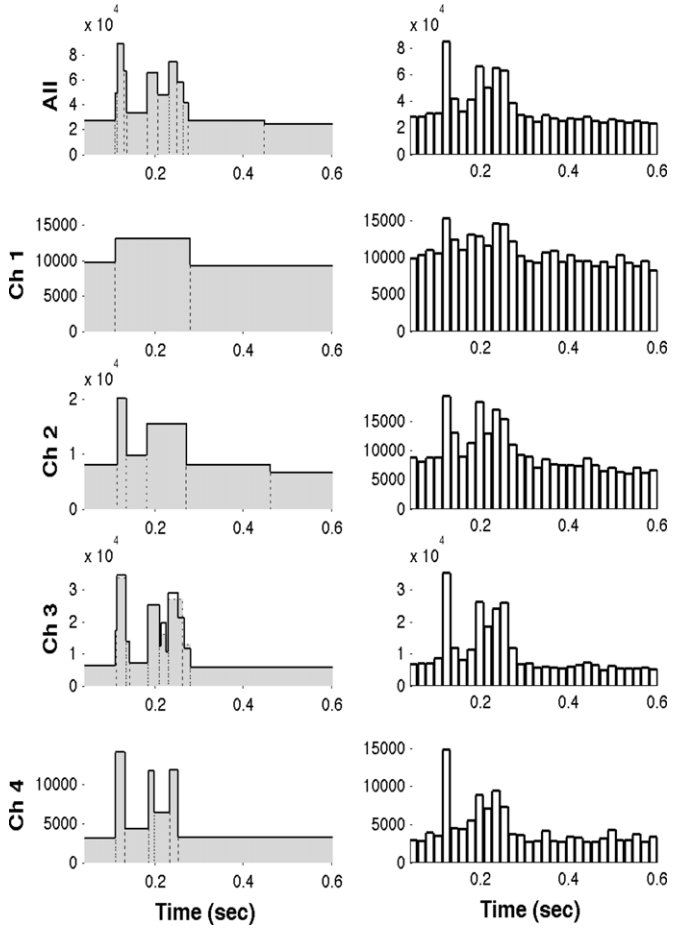

Figure 7. BATSE TTE data for Trigger 0551. Top panels: all photons. Other panels: photons in the four BATSE energy channels. Left column shows Bayesian block representations: default ncp_prior = solid lines; iterated ncp_prior = shaded/dashed lines. Right column: ordinary evenly spaced binned histograms.

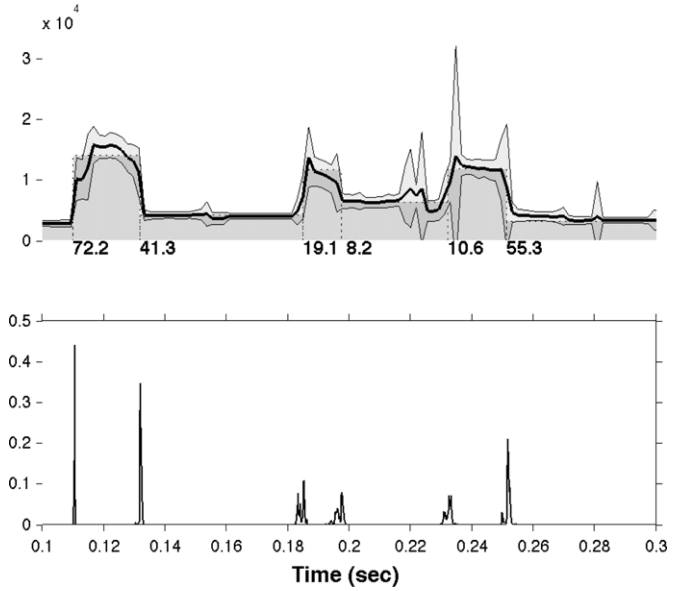

Figure 8. Error analysis for the data in Channel 4 from Figure 7, zooming in on the time interval with most of the activity. Top: heavy solid line is bootstrap mean (256 realizations), with thin lines giving the $\pm 1\sigma$ rms deviations, all superimposed on the BB representation. Bottom: approximate posterior distribution functions for the locations of the change points, obtained by fixing all of the others.

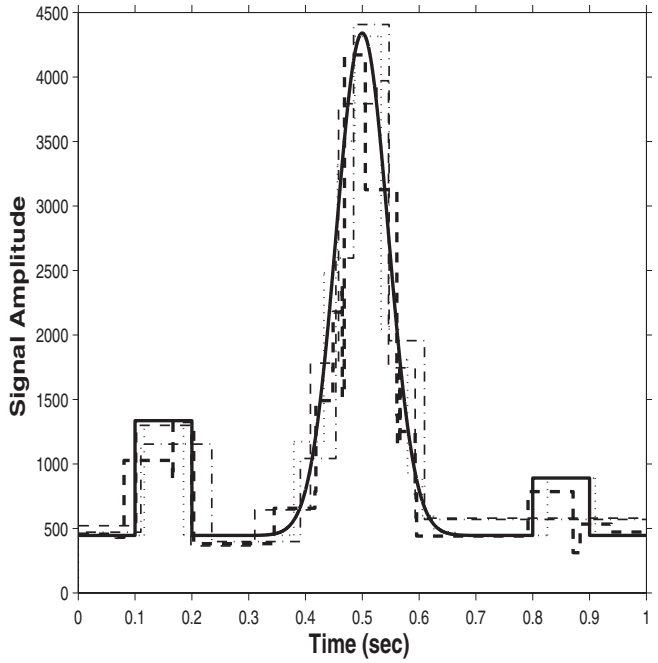

Figure 9. Multivariate analysis of synthetic signal consisting of two blocks surrounding a Gaussian shape centered on the interval [0, 1] (solid line). Optimal blocks for three independent data series drawn randomly from the probability distribution corresponding to this signal are thin lines: 1024 event times (dash), 4096 events in 32 bins (dot-dash), and 32 random amplitudes normally distributed with mean equal to the signal at random times uniformly distributed on [0, 1] and constant variance (dots). The thicker dashed line is the combined analysis of all three.

conditions can be programmed into the algorithm, such as returning after a specified number of change points have been found, or when the location of a change point has not moved for a specified length of time, etc. Essentially, any condition on the change points or the corresponding blocks can be imposed as a halt-and-return condition.

The real-time mode is mainly of use to detect the first sign of a time-dependent signal rising significantly above a slowly varying background. For example, in a photon stream the resulting *trigger* may indicate the presence of a new bursting or transient source.

The conventional way to approach problems of this sort is to report a detection if and when the actual event rate, averaged over some interval, exceeds one or more pre-set thresholds. See Band (2002) for an extensive discussion, as well as Fenimore et al. (2001), McLean et al. (2004), and Schmidt (2000) for other applications in high-energy astrophysics. One must consider a wide range of configurations: "Burst Alert Telescope uses about 800 different criteria to detect GRBs, each defined by a large number of commandable parameters" (McLean et al. 2004, 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 averaging over an interval in which it is presumed there is no signal, similar considerations apply to this interval.

Our segmentation algorithm greatly simplifies the above considerations, since pre-defined bin sizes and locations are not needed, and the background is automatically determined in real time. In practice there can be a slight complication for a

continuously accumulating data stream, since the N^2 dependence of the computing time may eventually make the computations unfeasible. A simple countermeasure is to analyze the data in a sliding window of moderate size—large enough to capture the desired changes but not so large that the computations take too long. Slow variations in the background in many cases could mandate something like a sliding window anyway.

Because of additional complexities, such as accounting for background variability and the Pandora's box that spectral resolution opens (Band 2002), we will defer a serious treatment of triggers to a future publication.

We end with a few comments on the *false alarm* (also called *false positive*) rate in the context of triggers. The considerations are very similar to the tradeoff discussed in the context of the choice for the parameter `ncp_prior` described in Sections 1.9, 2.7, and 3 for the various data modes. Even if no signal is present, a sufficiently large (and therefore rare) noise fluctuation can trigger any algorithm's detection criteria. Unavoidably, all detection procedures embody a tradeoff between sensitivity and rate of false alarms. Other things being equal, making an algorithm more able to trigger on weak signals renders it more sensitive to noise fluctuations. Conversely, making an algorithm shun noise fluctuations renders it insensitive to weak signals. In practice one chooses a balance of these competing factors based on the relative importance of avoiding false positives and not missing weak signals. Hence there can be no universal prescription.

4.4. Empty Blocks

Recall that blocks are taken to begin and end with data cells (Section 2.5). This convention means that no block can be empty: Each much contain at least its initiating data cell. Hence in the case of event data, blocks cannot represent intervals of zero event rate. This constraint is of no consequence for the other two data modes. There is nothing special about zero (or even negative) signals in the case of point measurements. Zero signal would be indicated by intervals containing only measured values not significantly different from zero. There is also no issue for binned data as nothing prevents a block from consisting of one or more empty data bins. In many event data applications zero signal may never occur (e.g., if there is a significant background over the entire observation interval). But in other cases it may be useful to represent such intervals in the form of a truly empty block, with corresponding zero height.

Allowing such null blocks is easily implemented in a post-processing step applied to each of the change points. The idea is to consider reassignment of data cells at the start or end of a block to the adjoining block while leaving the block lengths unchanged. For a given change point separating a pair of two blocks ("left" and "right") there are two possibilities: (1) the datum marking the change point itself, currently initiating the right block, can be moved from the right to the left block; and (2) the datum just prior to the change point itself, currently ending the left block, can be moved from the left block to the right block. Straightforward evaluation of the relevant fitness functions establishes whether one of these moves increases the fitness of the pair, and if so, which one. (It is impossible that this calculation will favor both moves (1) and (2); taken together they yield no net change and therefore leave fitness unchanged.)

The suggested procedure is to carry out this comparison for each change point in turn and adjust the populations of the blocks accordingly. We have not proved that this ad hoc prescription yields globally optimal models with the non-emptiness constraint removed, but it is obvious that the prescription can only increase overall model fitness. It is quite simple computationally and there is no real downside to using it routinely, even if the moves are almost never triggered. A code fragment to implement this procedure is given in Appendix A.

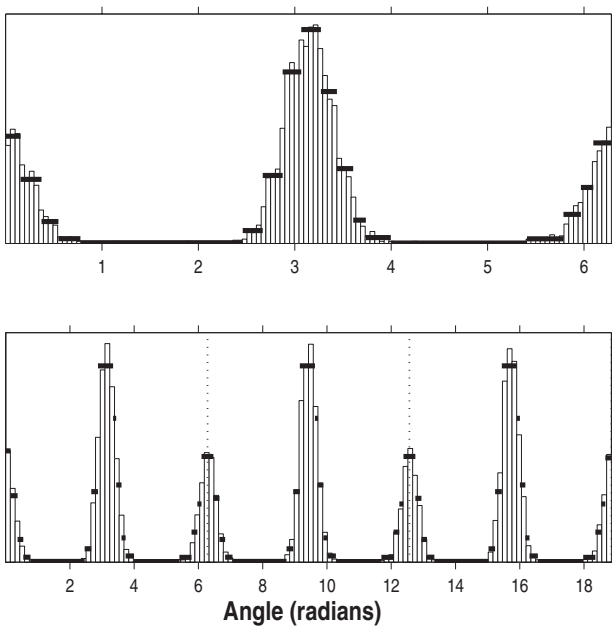

Figure 10. Data on the circle: events drawn from two normal distributions, centered at π and 0, the latter with some points wrapping around to values below 2π. Optimal blocks are depicted with thick horizontal bars superimposed on ordinary histograms. Top: block representation on the interval $[0, 2\pi]$. Bottom: block representation of three concatenated copies of the same data on $[0, 6\pi]$. Vertical dotted lines at 2π and 4π indicate boundaries between the copies. The blocks in the central copy, between these lines, are not influenced by end effects and are the correct optimal representation of these circular data.

4.5. Blocks on the Circle

Each of the data spaces discussed so far has been a linear interval with a well-defined beginning and end. A circle does not have this property. Our algorithm cannot be applied to data defined on a circle, such as directional measurements, because it starts with the first data point and iteratively works its way forward along the interval to the last point. (Of course the case where the measured value is confined to a specific subinterval of the circle is not a problem.) Hence the first and last points are treated as distant, not as the pair of adjacent points that they are. Any choice of starting point, such as the coordinate origin 0 for angles on $[0, 2\pi]$, disallows the possibility of a block containing data just before and after it (on the circle). In short, the iterative (mathematical induction-like) structure of the algorithm prevents it from being independent of the choice of origin, which on a circle is completely arbitrary. We have been unable to find a solution to this problem using a direct application of dynamical programming.

However, there is a method that provides exact solutions at the cost of about one order of magnitude more computation time. First unfold the data with an arbitrary choice for the fiducial origin. The resulting series starts at this origin, continues with the subsequent data points in order, and ends at the datum just prior to the fiducial origin. Think of cutting a loop of string and straightening it out.

The basic algorithm is then applied to the data series obtained by concatenating three copies of the unfolded data. The

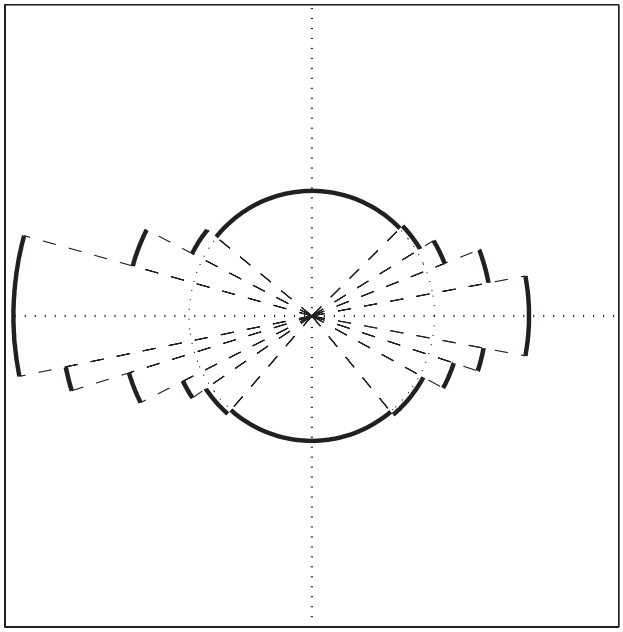

Figure 11. Optimal block representation of the same data as in Figure 10 (cf. the middle third of the bottom panel) plotted on the circle. The origin corresponds to the positive x-axis, and scale of the radius of the circle is arbitrary.

underlying idea is that the central copy is insulated from any effects of the discontinuity introduced by the unfolding. In extensive tests on simulated data this algorithm performed well. One check is whether or not the two sets of change points adjacent to the two divisions between the copies of the data are always equivalent (modulo the length of the circle). These results suggest but do not prove correctness for all data; there may be pathological cases for which it fails. Of course this N^2 computation will take ∼9 times as long as it would if the data were on a simple linear interval.

Figure 10 shows simulated data representing measurements of an angle on the interval $[0, 2\pi]$. In this case the procedure outlined above captures the central block (bottom panel) straddling the origin that is broken into two parts if the data series is taken to start at zero (upper panel). Note that the two blocks just above 0 and below 2π in the upper panel are rendered as a single block in the central cycle in the bottom panel. Figure 11 shows the same data shown in Figure 10 plotted explicitly on a circle.

As a footnote, one application that might not be obvious is the case of GRB light curves, which are short enough that the background is accurately constant over the duration of the burst. If all of the data are rescaled to fit on a circle, then the pre- and post-burst background would automatically be subsumed into a single block (covering intervals at the beginning and end of the observation period). This procedure would be applicable to bursting light curves of any kind if and only if the background signal is constant, so that the event rates before and after the main burst are the same.

5. CONCLUSIONS AND FUTURE WORK

The Bayesian Blocks algorithm finds the optimal step-function model of time series data by implementing the dynamical programming algorithm of Jackson et al. (2005). It is guaranteed to find the representation that maximizes any block-additive fitness function, in time of order N^2, and replaces the greedy approximate algorithm in Scargle (1998). Its real-time mode triggers on the first statistically significant rate change in a data stream.

This paper addresses the following issues in the use of the algorithm for a variety of data modes: gaps and exposure variations, piecewise linear and piecewise exponential models, the prior distribution for the number of blocks, multivariate data, the empty block problem (for event data), data on the circle, dispersed data, and analysis of variance ("error analysis"). The algorithm is shown to closely approach the theoretical detection limit derived in Arias-Castro et al. (2005).

Work in progress includes extensions to generalized data spaces, such as those of higher dimensions (Scargle 2001b), and speeding up the algorithm.

This work was supported by Joe Bredekamp and the NASA Applied Information Systems Research Program. We especially recognize the Center for Applied Mathematics, Computation and Statistics (CAMCOS) in the Department of Mathematics, San Jose State University for support through the Henry Woodward Fund. J.D.S. is grateful for the hospitality of the following institutions during various phases of this work: the Institute for Pure and Applied Mathematics at the University of California at Los Angeles, the Banff International Research Station, the Keck Institute for Space Studies at Caltech, the Kavli Institute for Particle Astrophysics at Stanford University and the Statistical and Mathematical Sciences Institute at Duke University. We are grateful to Tom Loredo, Glen MacLachlan, Erik Petigura, Jake Vanderplas, Zeljko Ivezic, Ery Arias-Castro, Sam Kou, Lin Lin, Talvikki Hovatta, and Marc Coram for helpful comments, and to Alice Allen for help with the posting at "The Engineering Deck: Astrophysics Source Code Library" on the Starship Asterisk Web site: http://asterisk.apod.com/. We are also grateful to the anonymous referee for useful suggestions.

APPENDIX A

REPRODUCIBLE RESEARCH: MATLAB CODE

This paper implements the spirit of Reproducible Research, a publication protocol initiated by John Claerbout (Claerbout 1990) and developed by others at Stanford and elsewhere. The underlying idea is that the most effective way of publishing research is to include everything necessary to reproduce all of the results presented in the paper. In addition to all relevant mathematical equations and the reasoning justifying them, full implementation of this protocol requires that the data files and computer programs used to prepare all figures and tables are included. Cogent arguments for Reproducible Research, an overview of its development history, and honest assessment of its successes and failures, are eloquently described in Donoho et al. (2009).

Following this discipline, all of the MatLab code and data files used in preparing this paper are available as auxiliary material. Included is the file `read_me.txt` with details and a script `reproduce_figures.m` that erases all of the figure files and regenerates them from scratch. In some cases the default parameters implement shorter simulation studies than those that were used for the figures in the paper, but one of the features of Reproducible Research is that such parameters and other aspects of the code can be changed and experimented with at will. Accordingly, this collection of scripts includes illustrative exemplars of the use of the algorithms and serves as a tutorial for the methods. In addition, a set of IDL routines (with extension `.pro`) are included in the auxiliary material file. These scripts are

not literal translations of the MatLab code, and in particular do not include the figure production feature, but they do implement the algorithm in much the same way.

Here is a commented version of the key fragment of the MatLab script (named `find_blocks.m`) for the basic algorithm described in this paper:

```
% For data modes 1 and 2:
% nn_vec is the array of cell populations.
% Preliminary computation:
block_length=tt_stop-[tt_start 0.5*(tt(2:end)
 +tt(1:end-1))' tt_stop];
...
%-------------------------------------------
% Start with first data cell; add one cell at
   each iteration
%-------------------------------------------
best = [];
last = [];
for R = 1:num_points
   % Compute fit_vec:fitness of putative last
      block (end at R)
    if data_mode == 3% Measurements, normal
     errors
        sum_x_1 = cumsum(cell_data(R:-1:1,
         1))'; %sum(x/sig^2)
        sum_x_0 = cumsum(cell_data(R:-1:1,
         2))'; %sum(1/sig^2)
        fit_vec=((sum_x_1(R:-1:1)) .^ 2).
        /(4*sum_x_0(R:-1:1));
    else
        arg_log = block_length(1:R) - block_
         length(R+1);
        arg_log(find(arg_log <= 0)) = Inf;
        nn_cum_vec = cumsum(nn_vec(R:-1:1));
        nn_cum_vec = nn_cum_vec(R:-1:1);
        fit_vec = nn_cum_vec .* (log(nn_cum_
         vec) - log(arg_log));
    end
    [best(R), last(R)] = max([0 best] +
      fit_vec - ncp_prior);
end
%-------------------------------------------
% Now find changepoints by iteratively peeling
   off the last block
%-------------------------------------------
index = last(num_points);
change_points = [];
while index > 1
    change_points = [index change_points];
    index = last(index - 1);
end
```

Additional information about Bayesian Blocks can be found at The Engineering Deck: Astrophysics Source Code Library at http://asterisk.apod.com/viewtopic.php?f=35&t=29458.

APPENDIX B

MATHEMATICAL DETAILS

Partitions of arrays of data cells are crucial to the block modeling that our algorithm implements. This appendix collects a few mathematical facts about partitions and the nature of independent events.

B.1. Definition of Partitions

A partition of a set is a collection of its subsets that add up to the whole with no overlap. Formally, a partition is a set of elements, or blocks $\{B_k\}$ satisfying

$$I = \bigcup_k B_k \tag{B1}$$

and

$$B_j \bigcap B_k = \emptyset \text{ (the empty set) for } j \neq k. \tag{B2}$$

Note that these conditions apply to the partitions of the time series data by sets of data cells. The data cells themselves may or may not partition the whole observation interval, as either the completeness in Equation (B1) or the no-overlap condition in Equation (B2) may be violated.

B.2. Reduction of Infinite Partition Space to a Finite One

For a continuous independent variable, such as time, the space of all possible partitions is infinitely large. We address this difficulty by introducing a construct in which $\mathcal{T}$ and its partitions are represented in terms of a collection of N discrete *data cells* in one-to-one correspondence with the measurements. The cells may form a partition of $\mathcal{T}$, as, for example, with event data with no gaps (see Section 3.1), but it is not necessary that they do so. The blocks that make up the partitions are sets of data cells contiguous with respect to time order of the cells. That is, a given block consists of exactly all cells with observation times within some subinterval of $\mathcal{T}$.

Now consider two sets of partitions of $\mathcal{T}$: (1) all possible partitions, and (2) all possible collections of cells into blocks. Set (1) is infinitely large since the block boundaries consist of arbitrary real numbers in $\mathcal{T}$ but set (2) is a finite subset of (1). Nevertheless, under reasonable assumptions about the data mode, any partition in (1) can be obtained from some partition in (2) by deforming boundaries of its blocks without crossing a data point. Because the potential of a block to be an element of the optimum partition (see the discussion of block fitness in Section 3) is a function of the content of the cells, such a transformation cannot substantially change the fitness of the partition.

B.3. The Number of Possible Partitions

How many different partitions of N cells are possible? Represent a partition by an ordered set of N zeros and ones, with one indicating that the corresponding time is a change point, and zero that it is not. With two choices at each time, the number of combinations is

$$N_{\text{partitions}} = 2^N. \tag{B3}$$

Except for very short time series this number is too large for an exhaustive search, but our algorithm nevertheless finds the optimum over this space in a time that scales as only N^2.

B.4. A Result for Subpartitions

We here define *subpartitions* and prove an elementary corollary that is key to the algorithm.

Definition. *A subpartition of a given partition $\mathcal{P}(I)$ is a subset of the blocks of $\mathcal{P}(I)$.*

It is obvious that a subpartition is a partition of that subset of $\mathcal{T}$ consisting of those blocks. Although not a necessary condition for the result to be true, in all cases of interest here the blocks in the subpartition are contiguous, and thus form a partition of a subinterval of $\mathcal{T}$. It follows that:

Theorem. *A subpartition $\mathcal{P}'$ of an optimal partition $\mathcal{P}(I)$ is an optimal partition of the subset I' that it covers.*

For if there were a partition of I', different from and fitter than $\mathcal{P}'$, then combining it with the blocks of $\mathcal{P}$ not in $\mathcal{P}'$ would, by the block additivity condition, yield a partition of $\mathcal{T}$ fitter than $\mathcal{P}$, contrary to the optimality of $\mathcal{P}$.

We will make use of the following corollary:

Corollary. *Removing the last block of an optimal partition leaves an optimal partition.*

B.5. Essential Nature of the "Poisson" Process

The term *Poisson process* refers to events occurring randomly in time and *independently of each other*. That is, the times of the events,

$$t_n, n = 1, 2, \ldots, N, \tag{B4}$$

are independently drawn from a given probability distribution. Think of the events as darts thrown randomly at the interval. If the distribution is flat (i.e., the same all over the interval of interest) we have a *constant rate Poisson process*. In this special case, a point is just as likely to occur anywhere in the interval as it is anywhere else; but this need not be so. What must be so in general—the essential nature of the Poisson process from a physical point of view—is the above-mentioned independence: each dart is not at all influenced by the others. Throwing darts that have feathers or magnets, although random, is not a Poisson process if these accoutrements cause the darts to repel or attract each other.

This key property of independence determines all of the other features of the process. Most important are a set of remarkable properties of interval distributions (Papoulis 1965). The time interval between a given point t_0 and the time t of the next event is exponentially distributed

$$P(\tau)d\tau = \lambda e^{-\lambda\tau} d\tau, \tag{B5}$$

where $\tau = t - t_0$. The remarkable aspect is that it does not matter how t_0 is chosen; in particular the distribution is the same whether or not an event occurs at t_0. This fact makes the implementation of event-by-event exposure straightforward (Section 1.8).

Note that we have not mentioned the Poisson distribution itself. The number of events in a fixed interval does obey the Poisson distribution, but this result is subsidiary to, and follows from, event independence. In this sense a better name than *Poisson process* is *independent event process*.

In representing intensities of such processes, one scheme is to represent each event as a delta function in time. But a more convenient way to extract rate information incorporates the time intervals between photons. (A method for analyzing event data based solely on inter-event time intervals has been developed in Prahl (1999).) Specifically, for each photon consider the interval starting half way back to the previous photon and ending half way forward to the subsequent photon. This interval, namely,

$$\left[\frac{t_n - t_{n-1}}{2}, \frac{t_{n+1} - t_n}{2}\right], \tag{B6}$$

is the set of times closer to t_n than to any other time,[8] and has length equal to the average of the two intervals connected by photon n, namely,

$$\Delta t_n = \frac{t_{n+1} - t_{n-1}}{2}. \tag{B7}$$

Then the reciprocal

$$x_n \equiv \frac{1}{\Delta t_n} \tag{B8}$$

is taken as an estimate of the signal amplitude corresponding to observation n. When the photon rate is large, the corresponding intervals are small, demonstrates the data cell concept, including the simple modifications to account for variable exposure and for weighting by photon energy.

Prahl (1999) has derived a statistic for event clustering in Poisson process data that tests departures from the known interval distribution by evaluating the likelihood over a restricted interval range. Prahl's statistic is

$$M_N = \frac{1}{N} \sum_{\Delta T_i < C^*} \left(1 - \frac{\Delta T_i}{C^*}\right), \tag{B9}$$

where ΔT_i is the interval between events i and $i + 1$, and

$$C^* \equiv \frac{1}{N} \sum \Delta T_i \tag{B10}$$

is the empirical mean interval. In other settings, the fact that this statistic is a global measure of departure of the distribution (used here only locally, over one block) may be useful in the detection of periodic (and other global) signals in event data.

APPENDIX C

OTHER BLOCK FITNESS FUNCTIONS

This appendix describes fitness function for a variety of data modes.

C.1. Event Data: Alternate Derivation

The Cash statistic used to derive the fitness function in Equation (19) is based on representation of event times as real numbers. Of course time is not measured with infinite precision, so it is interesting to note that a more realistic treatment yields the same formula.

Typically the data systems' finest time resolution is represented as an elementary quantum of time, which will be called a *tick* since it is usually set by a computer clock. Measured values are expressed as integer multiples of it; cf. Section 2.2.1 of Scargle (1998). We assume that n_m, the number of events (e.g., photons) detected in tick m obeys a Poisson distribution:

$$L_m = \frac{(\lambda dt)^{n_m}\, e^{-\lambda dt}}{n_m!} = \frac{\Lambda^{n_m}\, e^{-\Lambda}}{n_m!}, \tag{C1}$$

[8] These intervals form the Voronoi tessellation of the total observation interval. See Okabe et al. (2000) for a full discussion of this construct, highly useful in spatial domains of 2, 3, or higher dimension; see also Scargle (2001a) and Scargle (2001b).

where dt is the length of the tick. The event rates λ and Λ are counts per second and per tick, respectively. Time here is given in units such as seconds, but a representation in terms of (dimensionless) integer multiples of dt is sometimes more convenient.

Due to event independence the block likelihood is the product of these individual factors over all ticks in the block. Assuming that all ticks have the same length dt this is

$$L^{(k)} = \prod_{m=1}^{M^{(k)}} \frac{(\lambda dt)^{n_m} e^{-\lambda dt}}{n_m!}, \tag{C2}$$

where $M^{(k)}$ is the number of ticks in block k. Note that non-events are included via the factor $e^{-\lambda dt}$ for each tick with $n_m = 0$. When this expression is used to compute the likelihood for the whole interval (i.e., product of the block likelihoods over all blocks of the model) the denominator contributes the factor

$$\frac{1}{\prod_k \prod_m^{M^{(k)}} n_m!} = \frac{1}{\prod_m n_m!}, \tag{C3}$$

where on the right-hand side the product is over all the ticks in the whole interval. For low event rates where n_m never exceeds 1, this quantity is unity. No matter what, it is a constant, fixed once and for all given the data; in model comparison contexts it is independent of model parameters and hence irrelevant. Dropping it, noting that $\prod_{m=1}^{M^{(k)}} e^{-\lambda dt}$ is just $e^{-\lambda M^{(k)} dt} = e^{-\lambda M^{(k)}}$. Collecting together all factors for ticks with the same number of events Equation (C2) simplifies to

$$L^{(k)} = e^{-\lambda M^{(k)}} \prod_{n=0}^{\infty} (\lambda dt)^{n H^{(k)}(n)}, \tag{C4}$$

where $H^{(k)}(n)$ is the number of ticks in the block with n events. Noting that

$$\sum_{n=0}^{\infty} n H^{(k)}(n) = N^{(k)}, \tag{C5}$$

where $N^{(k)}$ is the total number of events in block k, we have simply

$$L^{(k)} = (\lambda dt)^{N^{(k)}} e^{-\lambda M^{(k)}}. \tag{C6}$$

In order for the model to depend on only the parameters defining the block edges, we need to eliminate λ from Equation (C6). One way to do this is to find the maximum of this likelihood as a function of λ, which is easily seen to be at $\lambda = (N^{(k)}/M^{(k)})$, yielding

$$L^{(k)}_{\max} = \left(\frac{N^{(k)} dt}{M^{(k)}} \right)^{N^{(k)}} e^{-N^{(k)}}. \tag{C7}$$

The exponential contributes the overall constant factor $e^{-\sum_k N^{(k)}} = e^{-N}$ to the full model. Moving this ultimately irrelevant factor to the left-hand side, noting that $M^{(k)} = (T^{(k)}/dt)$, and taking the log, we have for the maximum likelihood block fitness function

$$\log L^{(k)}_{\max} + N^{(k)} = N^{(k)} (\log N^{(k)} - \log M^{(k)}), \tag{C8}$$

equivalent to Equation (19).

An alternative way to eliminate λ is to marginalize it as in the Bayesian formalism. That is, one specifies a prior probability distribution for the parameter and integrates the likelihood in Equation (C6) times this prior. Since the current context is generic, not devoted to a specific application, we seek a distribution that expresses no particular prior knowledge for the value of λ. It is well known that there are several practical and philosophical issues connected with such so-called *non-informative* priors. Here we adopt this simple flat, normalized prior:

$$P(\lambda) = \begin{cases} P^{(\Delta)} & \lambda_1 \leqslant \lambda \leqslant \lambda_2 \\ 0 & \text{otherwise} \end{cases}, \tag{C9}$$

where the normalization condition yields

$$P^{(\Delta)} = \frac{1}{\lambda_2 - \lambda_1} = \frac{1}{\Delta\lambda}. \tag{C10}$$

Thus Equation (C6), with λ marginalized, is the posterior probability

$$P^{(k)}_{\text{marg}} = P^{(\Delta)} \int_{\lambda_1}^{\lambda_2} (\lambda dt)^{N^{(k)}} e^{-\lambda T^{(k)}} d\lambda \tag{C11}$$

$$= \frac{P^{(\Delta)}}{T^{(k)}} \left(\frac{dt}{T^{(k)}} \right)^{N^{(k)}} \int_{z_1}^{z_2} z^{N^{(k)}} e^{-z} dz \tag{C12}$$

where $z_{1,2} = T^{(k)} \lambda_{1,2}$. In terms of the *incomplete gamma function*

$$\gamma(a, x) \equiv \int_0^x z^{a-1} e^{-z} dz, \tag{C13}$$

we have, utilizing $M^k = (T^{(k)}/dt)$,

$$\log P^{(k)}_{\text{marg}} = \log \frac{P^{(\Delta)}}{T^{(k)}} - N^{(k)} \log M^{(k)} + \log[\gamma(N^{(k)} + 1, z_2) - \gamma(N^{(k)} + 1, z_1)]. \tag{C14}$$

The infinite range $z_1 = 0, z_2 = \infty$, gives

$$\log P^{(k)}_{\text{marg}(\infty)} = \log \frac{P^{(\Delta)}}{T^{(k)}} + \log \Gamma(N^{(k)} + 1) - N^{(k)} \log M^{(k)}. \tag{C15}$$

This prior is unnormalized (and therefore sometimes regarded as improper). Technically $P^{(\Delta)}$ approaches zero as $z_2 \to \infty$, but is retained here in order to formally retain the scale invariance to be discussed at the end of this section.

Another commonly used prior is the so-called conjugate Poisson distribution

$$P(\lambda) = C\, \lambda^{\alpha - 1} e^{-\beta\lambda}. \tag{C16}$$

As noted by Gelman et al. (1995, pg. 49) this "prior density is, in some sense, equivalent to a total count of $\alpha - 1$ in β prior observations," a relation that might be useful in some circumstances. The normalization constant $C = (\beta^\alpha / \Gamma(\alpha))$, and with this prior the marginalized posterior probability distribution is

$$P_{\text{cp}} = C \int_0^{\infty} \lambda^{N^{(k)} + \alpha - 1} e^{-\lambda(M^{(k)} + \beta)} d\lambda, \tag{C17}$$

yielding

$$\log P_{\text{cp}} - \log C = \log \Gamma(N^{(k)} + \alpha) - (N^{(k)} + \alpha) \log(M^{(k)} + \beta). \tag{C18}$$

Note that for $\alpha = 1, \beta = 1$ this prior and posterior reduce to those in Equations (28) and (29) of Scargle (1998).

Equations (19), (C14), (C15), and (C18) are all invariant under a change in the units of time. The case of Equation (C15) is slightly dodgy, as mentioned above, but otherwise is a direct result of expressing $N^{(k)}$ and $M^{(k)}$ as dimensionless counts, of events and time ticks, respectively. (Further, in the case of Equation (C14), z_1 and z_2 are dimensionless.) As mentioned above, the simplicity of Equation (19) recommends it in general, but specific prior information (e.g., as represented by Equation (C16)) may suggest use of one of the other forms.

C.2. 0–1 Event Data: Duplicate Time Tags Forbidden

In Mode 2 duplicate time tags are not allowed, the number of events detected at a given tick is 0 or 1, and the corresponding tick likelihood is

$$L_m = e^{-\lambda dt} = 1 - p \qquad n_m = 0 \tag{C19}$$

$$= 1 - e^{-\lambda dt} = p \qquad n_m = 1, \tag{C20}$$

where λ is the model event rate, in events per unit time. From the Poisson distribution $p = 1 - e^{-\lambda dt}$ is the probability of an event, and $1 - p = e^{-\lambda dt}$ that of no event. Note that p or λ interchangeably specify the event rate. Since independent probabilities multiply, the block likelihood is the product of the tick likelihoods:

$$L^{(k)} = \prod_{m=1}^{M^{(k)}} L_m = p^{N^{(k)}}(1-p)^{M^{(k)}-N^{(k)}}, \tag{C21}$$

where $M^{(k)}$ is the number of ticks in block k and $N^{(k)}$ is the number of events in the block.

There are again two ways to proceed. The maximum of this likelihood occurs at $p = (N^{(k)}/M^{(k)})$ and is

$$L^{(k)}_{\max} = \left(\frac{N^{(k)}}{M^{(k)}}\right)^{N^{(k)}} \left(1 - \frac{N^{(k)}}{M^{(k)}}\right)^{M^{(k)}-N^{(k)}}. \tag{C22}$$

Using the logarithm of the maximum likelihood,

$$\log L^{(k)}_{\max} = N^{(k)} \log\left(\tfrac{N^{(k)}}{M^{(k)}}\right) + (M^{(k)} - N^{(k)}) \log\left(1 - \tfrac{N^{(k)}}{M^{(k)}}\right), \tag{C23}$$

yields the fitness function, additive over blocks.

As in the previous subsection, an alternative is to marginalize λ:

$$P^{(k)} = \int L^{(k)} P(\lambda) d\lambda, \tag{C24}$$

where $P(\lambda)$ is the prior probability distribution for the rate parameter. With the flat prior in Equation (C9)[9] the posterior, marginalized over λ is

$$P^{(k)}_{\rm marg} = P^{(\Delta)} \int_{\lambda_1}^{\lambda_2} (1 - e^{-\lambda dt})^N (e^{-\lambda dt})^{M^{(k)}-N^k} d\lambda. \tag{C25}$$

Changing variables to $p = 1 - e^{-\lambda dt}$, with $dp = dt\ e^{-\lambda dt} d\lambda$, this integral becomes

$$P^{(k)}_{\rm marg} = \frac{P^{(\Delta)}}{dt} \int_{p_1}^{p_2} p^{N^{(k)}} (1-p)^{M^{(k)}-N^{(k)}-1} dp, \tag{C26}$$

with $p_{1,2} = 1 - e^{-\lambda_{1,2} dt}$, and expressible in terms of the *incomplete beta function*

$$B(z; a, b) = \int_0^z u^{a-1}(1-u)^{b-1} du \tag{C27}$$

as follows:

$$\log P^{(k)}_{\rm marg} - \log \frac{P^{(\Delta)}}{dt} = \log[B(p_2; N^{(k)} + 1, M^{(k)} - N^{(k)}) - B(p_1; N^{(k)} + 1, M^{(k)} - N^{(k)})]. \tag{C28}$$

The case $p_1 = 0$, $p_2 = 1$ yields the ordinary *beta function*:

$$\log P^{(k)}_{0\to 1} - \log \frac{P^{(\Delta)}}{dt} = \log B(N^{(k)} + 1, M^{(k)} - N^{(k)}), \tag{C29}$$

differing from Equation (21) of Scargle (1998) by one in the second argument, due to the difference between a prior flat in p and one flat in λ. All of the Equations (C23), (C28), and (C29), can be used as fitness functions in the global optimization algorithm and, as with Mode 1, are invariant to a change in the units of time.

A brief aside: One might be tempted to use intervals between successive events instead of the actual times, since in some sense they express rate information more directly. However, as we now prove, the likelihood based on intervals is essentially equivalent to that in Equation (C6). It is a classic result (Papoulis 1965) that intervals between (time-ordered) consecutive independent events (occurring with a probability uniform in time, with a constant rate λ) are exponentially distributed:

$$P(dt)dt = \lambda e^{-\lambda dt} U(dt) dt, \tag{C30}$$

where $U(x)$ is the unit step function:

$$U(x) = 1 \quad x \geqslant 0$$
$$= 0 \quad x < 0.$$

Pretend that the data consist of the inter-event intervals, and that one does not even know the absolute times. The likelihood of our constant rate Poisson model for interval $dt_n \geqslant 0$ is

$$L_n = \lambda e^{-\lambda\ dt_n}, \tag{C31}$$

so the block likelihood is

$$L^{(k)} = \prod_{n=1}^{N^{(k)}} \lambda\ e^{-\lambda\ dt_n} = \lambda^{N^{(k)}} e^{-\lambda M^{(k)}}, \tag{C32}$$

the same as in Equation (C6), except that here $N^{(k)}$ is the number of inter-event intervals, one less than the number of events.

Prahl (1999) derived a statistic for event clustering, by testing for significant departures from the known interval distribution, by evaluating the likelihood over a restricted interval range. This statistic is

$$M_N = \frac{1}{N} \sum_{\Delta T_i < C^*} \left(1 - \frac{\Delta T_i}{C^*}\right), \tag{C33}$$

where ΔT_i is the interval between events $\mathcal{T}$ and $i + 1$, N is the number of terms in the sum, and

$$C^* \equiv \frac{1}{N} \sum \Delta T_i \tag{C34}$$

is the empirical mean of the relevant intervals. In some settings, the fact that this statistic is a global measure (as opposed to the local—over one block at a time—ones used here) may be useful in the detection of global signals, such as periodicities, in event data.

[9] In Scargle (1998) we used p as the independent variable, and chose a prior flat (constant) as a function of p. Here, we use a prior flat as a function of the rate parameter.

C.3. Time-to-spill Data

As discussed in Section 2.2.3 of Scargle (1998), reduction of the necessary telemetry rate is sometimes accomplished by recording only the time of detection of every Sth photon, e.g., with $S = 64$ for the BATSE time-to-spill mode. This data mode has the attractive feature that its time resolution is greater when the source is brighter (and possibly more active, so that more time resolution is useful). With slightly revised notation, the likelihood in Equation (32) of Scargle (1998) simplifies to

$$L^{(k)}_{\rm TTS} = \lambda^{SN^{(k)}_{\rm spill}} e^{-\lambda M^{(k)}}, \tag{C35}$$

where $N^{(k)}_{\rm spill}$ is the number of spill events in the block and $M^{(k)}$ is as usual the length of the block in ticks. With $N = N^{(k)}_{\rm spill} S$ this is identical to the Poisson likelihood in Equation (C2), and in particular the maximum likelihood is at $\lambda = (N^{(k)}_{\rm spill} S/M^{(k)})$ and the corresponding fitness function is

$$\log L^{(k)}_{\rm max,TTS} - \log N = SN^{(k)}_{\rm spill}\,[\log(N^{(k)}_{\rm spill} S) - \log M^{(k)}] \tag{C36}$$

just as in Equation (19) with $N^{(k)} = SN^{(k)}_{\rm spill}$, and with the same property that the unit in which block lengths are expressed is irrelevant.

C.4. Point Measurements: Alternative Form

An alternative form can be derived by inserting Equation (38) instead of Equation (36) into the log of Equation (30) as in Section 3.3. The result is

$$\log L^{(k)}_{\rm max} = -\frac{1}{2}\sum_n \left(\frac{x_n - \sum_{n'} w_{n'} x_{n'}}{\sigma_n}\right)^2. \tag{C37}$$

Expanding the square gives

$$\log L^{(k)}_{\rm max} = -\frac{1}{2}\left[\sum_n \left(\frac{x_n}{\sigma_n}\right)^2 - 2\sum_n \left(\frac{x_n}{\sigma_n^2}\right)\left(\sum_{n'} w_{n'} x_{n'}\right) + \left(\sum_{n'} w_{n'} x_{n'}\right)^2 \sum_n \frac{1}{\sigma_n^2}\right] \tag{C38}$$

$$= -\frac{1}{2}\sum_{n'}\left(\frac{1}{\sigma_{n'}^2}\right)\left[\sum_n w_n x_n^2 - 2\left(\sum_n w_n x_n\right)\left(\sum_{n'} w_{n'} x_{n'}\right) + \left(\sum_{n'} w_{n'} x_{n'}\right)^2 \sum_n w_n\right] \tag{C39}$$

$$= -\frac{1}{2}\sum_{n'}\left(\frac{1}{\sigma_{n'}^2}\right)\left[\sum_n w_n x_n^2 - 2\left(\sum_n w_n x_n\right)^2 + \left(\sum_{n'} w_{n'} x_{n'}\right)^2\right] \tag{C40}$$

$$= -\frac{1}{2}\sum_{n'}\left(\frac{1}{\sigma_{n'}^2}\right)\left[\sum_n w_n x_n^2 - \left(\sum_n w_n x_n\right)^2\right] \tag{C41}$$

yielding

$$\log L^{(k)}_{\rm max} = -\frac{1}{2}\left[\sum_{n'}\left(\frac{1}{\sigma_{n'}^2}\right)\right]\sigma_X^2 \tag{C42}$$

where

$$\sigma_X^2 \equiv \sum_n w_n x_n^2 - \left(\sum_n w_n x_n\right)^2 \tag{C43}$$

is the *weighted average variance* of the measured signal values in the block. It makes sense that the block fitness function is proportional to the negative of the variance: the best constant model for the block should have minimum variance.

C.5. Point Measurements: Marginal Posterior, Flat Prior

First, consider the simplest choice, the flat, unnormalizable prior

$$P(\lambda) = P^* \quad (\text{for all values of } \lambda), \tag{C44}$$

giving equal weight to all values. The marginal posterior for block k is then, from Equation (30),

$$P^k = P^* \frac{(2\pi)^{-\frac{N_k}{2}}}{\prod_n \sigma_n} \int_{-\infty}^{\infty} e^{-\frac{1}{2}\sum_n (\frac{x_n-\lambda}{\sigma_n})^2} d\lambda. \tag{C45}$$

Using the definitions introduced above in Equations (31)–(33) we have

$$P^k = P^* \frac{(2\pi)^{-\frac{N_k}{2}}}{\prod_n \sigma_n} \int_{-\infty}^{\infty} e^{-(a_k\lambda^2 + b_k\lambda + c_k)}\, d\lambda. \tag{C46}$$

Using standard "completing the square," letting $z = \sqrt{a_k}(\lambda + (b_k/2a_k))$, giving

$$z^2 = a_k\left(\lambda + \frac{b_k}{2a_k}\right)^2 = a_k\left(\lambda^2 + \frac{\lambda b_k}{a_k} + \frac{b_k^2}{4a_k^2}\right) = a_k\lambda^2 + b_k\lambda + c_k + \frac{b_k^2}{4a_k} - c_k, \tag{C47}$$

and then using

$$\int_{-\infty}^{+\infty} e^{-z^2}\frac{dz}{\sqrt{a_k}} = \sqrt{\frac{\pi}{a_k}}, \tag{C48}$$

we have

$$P^k = P^* \frac{(2\pi)^{-\frac{N_k}{2}}}{\prod_n \sigma_n}\sqrt{\frac{\pi}{a_k}} e^{\left(\frac{b_k^2}{4a_k}\right) - c_k}. \tag{C49}$$

From this result, the log-posterior fitness function is

$$\log P_0^k - A_k = \log\left(P^*\sqrt{\frac{\pi}{a_k}}\right) + \left(\frac{b_k^2}{4a_k}\right) - c_k, \tag{C50}$$

where

$$A_k = -\frac{N_k}{2}\log(2\pi) - \sum \log(\sigma_n) \tag{C51}$$

and the subscript 0 refers to the fact that the marginal posterior was obtained with the unnormalized prior. The second and third terms in Equation (C50) are invariant under the transformation (Equation (42)). Further, since the integral of $P(\lambda)$ with respect to λ must be dimensionless, we have $P^* \sim (1/\lambda) \sim (1/x)$, so P^* and $\sqrt{a_k}$ have the same a-dependence, yielding a formal invariance for Equation (C50). However, the prior in Equation (C44) is not normalizable, so that technically P^* is undefined. A way to make practical use of this formal invariance is simply to include a constant P^* that has the proper dimension (x^{-1}).

C.6. Point Measurements: Marginal Posterior, Normalized Flat Prior

Marginalizing the likelihood in Equation (30) with the prior in Equation (C9), yields for the marginal posterior for block k:

$$P^k = P^{(\Delta)} \frac{(2\pi)^{-\frac{N_k}{2}}}{\prod_n \sigma_n} \int_{\lambda_1}^{\lambda 2} e^{-\frac{1}{2}\sum_n (\frac{x_n-\lambda}{\sigma_n})^2} d\lambda. \quad \text{(C52)}$$

As before

$$P^k = P^{(\Delta)} \frac{(2\pi)^{-\frac{N_k}{2}}}{\prod_n \sigma_n} \int_{\lambda_1}^{\lambda 2} e^{-(a_k\lambda^2+b_k\lambda+c_k)}\, d\lambda. \quad \text{(C53)}$$

Now complete the square by letting $z = \sqrt{a_k}(\lambda + (b_k/2a_k))$, giving

$$z^2 = a_k\left(\lambda + \frac{b_k}{2a_k}\right)^2 = a_k\left(\lambda^2 + \frac{\lambda b_k}{a_k} + \frac{b_k^2}{4a_k^2}\right)$$
$$= a_k\lambda^2 + b_k\lambda + \frac{b_k^2}{4a_k} + c_k - c_k, \quad \text{(C54)}$$

so we have

$$P^k = P^{(\Delta)} \frac{(2\pi)^{-\frac{N_k}{2}}}{\prod_n \sigma_n} e^{\left(\frac{b_k^2}{4a_k}-c_k\right)} \int_{z_1}^{z_2} e^{-z^2} \frac{dz}{\sqrt{a_k}}, \quad \text{(C55)}$$

where

$$z_{1,2} = \sqrt{a_k}\left(\lambda_{1.2} + \frac{b_k}{2a_k}\right). \quad \text{(C56)}$$

Finally, introducing the error function

$$\text{erf}(x) = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt, \quad \text{(C57)}$$

we have

$$P^k = P^{(\Delta)} \frac{\sqrt{\pi}}{2} \frac{(2\pi)^{-\frac{N_k}{2}}}{\sqrt{a_k}\prod_n \sigma_n} e^{\left(\frac{b_k^2}{4a_k}-c_k\right)} [\text{erf}(z_2) - \text{erf}(z_1)]. \quad \text{(C58)}$$

Taking the log gives the final expression

$$\log P_\Delta^k - A_k = \log\left(P^{(\Delta)}\sqrt{\frac{\pi}{a_k}}\right) + \left(\frac{b_k^2}{4a_k} - c_k\right) + \log\left[\frac{\text{erf}(z_2)-\text{erf}(z_1)}{2}\right], \quad \text{(C59)}$$

where the subscript Δ indicates the fact that this result is based on the finite range prior in Equation (C9). Note that this fitness function is manifestly invariant under the transformation in Equation (42), for the same reasons discussed at the end of the previous section, plus the invariance of $z_{1,2}$. In the limits $z_1 \to -\infty$ and $z_2 \to \infty$, $\text{erf}(z_2) - \text{erf}(z_1) \to 2$, and we recover Equation (C50)—but remember that in this limit the invariance is only formal.

C.7. Point Measurements: Marginal Posterior, Gaussian Prior

Finally, consider using the following normalized Gaussian prior for λ:

$$P(\lambda) = \frac{1}{\sigma_0\sqrt{2\pi}} e^{-\frac{1}{2}(\frac{\lambda-\lambda_0}{\sigma_0})^2} \quad \text{(C60)}$$

corresponding to prior knowledge that roughly speaking λ most likely lies in the range $\lambda_0 \pm \sigma_0$, with a normal distribution. This prior is not to be confused with the Gaussian form for the likelihood in Equation (29).

Equation (30), when λ is marginalized with this prior, becomes

$$L^{(k)} = \frac{1}{\sigma_0\sqrt{2\pi}} \left[\frac{(2\pi)^{-(\frac{N_k}{2})}}{\prod_{n'} \sigma_{n'}}\right] \times \int e^{-\frac{1}{2}\left[\lambda^2\left(\frac{1}{\sigma_0^2}+\sum_n \frac{1}{\sigma_n^2}\right)+\lambda\left(-\frac{2\lambda_0}{\sigma_0^2}-\frac{2x_n}{\sigma_0^2}\right)+\left(\frac{\lambda_0^2}{\sigma_0^2}+\sum_n \frac{x_n^2}{\sigma_n^2}\right)\right]}, \quad \text{(C61)}$$

so with

$$a_k = \frac{1}{2}\left(\frac{1}{\sigma_0^2} + \sum_n \frac{1}{\sigma_n^2}\right) \quad \text{(C62)}$$

$$b_k = -\left(\frac{\lambda_0}{\sigma_0^2} + \sum_n \frac{x_n}{\sigma_n^2}\right) \quad \text{(C63)}$$

and

$$c_k = \frac{1}{2}\left(\frac{\lambda_0^2}{\sigma_0^2} + \sum_n \frac{x_n^2}{\sigma_n^2}\right) \quad \text{(C64)}$$

and Equation (C46) is recovered, so that Equation (C50), with the redefined coefficients in Equations (C62)–(C64), gives the final fitness function.

Any of the log fitness functions in Equations (C42), (C50), or (C59) can be used for the point measurement data mode in this section. This choice should be made based on convenience or the relevant prior information.

C.8. Data with Dispersed Measurements

Throughout it has been presumed that two things are small compared with any relevant timescales: errors in the determination of times of events, and the intervals over which individual measurements are obtained as averages. These assumptions justify treatment of the corresponding data modes as points in Sections 3.1 and 3.3, respectively. Below are discussions of data that are dispersed because of (1) random errors in event times and (2) measurements that are summations or averages over non-negligible intervals. Binned data, an example of the latter, have already been treated in Section 3.2 and are not discussed here.

A simple ad hoc way to deal with both of these situations is to compute kernel functions for each data point, representing the window or error distribution in either of the two above contexts. Each such function would be centered at the corresponding measured value, evaluated at all of the data points, and normalized to represent unit intensity. Each such kernel would be maximum at the data point at which it is centered, but distribute some weight to the other data cells. The sum of all of these kernels would then be a set of weights at each measurement, which could then be treated as ordinary event data but with fractional rather than unit weights. The ad hoc aspect of this approach lies in the way the fitness function is extended. The following subsections provide more rigorous analysis.

C.8.1. Uncertain Event Locations

Timing of events is always uncertain at some level. Here we treat the case where the error distribution is wide enough to make the point approximation inappropriate. Rare for photon

time series, with microsecond timing errors, this situation is more common in other contexts and with other independent variables. With overlapping error distributions even the order of events can be uncertain. In the context described in Section 1.4 one often wants to construct histograms from measurements with errors—errors that may be different for each point (then called *heteroscedastic errors*).

A simple modification of the fitness function described in Section 3.1 addresses this kind of data. On the right-hand side of Equation (19) $N^{(k)}$ quantifies the contribution of the individual events within block k. In extending the reasoning leading to this fitness function, the main issue concerns events with error distributions that have fractional overlap with the extent of block k—for events distributed entirely outside (inside) obviously contribute in no way (fully) to block fitness. By the law for the sum of probabilities of independent events, in the log-likelihood implicit in Equations (17) and (18) $N^{(k)}$ is replaced by the sum of the areas under the probability distributions overlapping block k, namely, $\sum_{i\in k} p^{(i)}$ summed over all events with significant contribution to block k, and $p^{(i)}$ is the integral of the overlapping part of the error distribution, a fraction between 0 and 1. Thus we have

$$\log L^{(k)}(\lambda) = \log\lambda \sum_{i\in k} p^{(i)} \; - \lambda T^{(k)} \tag{C65}$$

in place of Equation (19), with the analogous constant term on the left-hand side of that equation dropped. This result holds because a given datum falling inside and outside a block are mutually exclusive events.

Implementing this relationship in the algorithm is easily accomplished. For a given event and the interval assigned to it (cf. Figure 12) sum the overlap fractions with that interval of all events—including that event itself. These quantities could be approximated with very simple or complex quadrature schemes, depending on the context and the way in which the relevant distributions are represented. Normally the array `nn_vec`, as in the code fragment in Appendix A, is all 1's (or counts of events with identical time tags there are any); but here replace it with these summed event weights. This construction automatically assigns the correct fractional weights to the block with no further alteration of the algorithm.

C.8.2. Measurements in Extended Windows

This section discusses the case of *distributed measurements* in the sense that the time of measurement is either uncertain or is effectively an interval rather than a point. (This is different from the use of this term in Section 3.3 to describe the distribution of the measurement error in the dependent variable.) Measurements may refer to a quantity averaged over a range of values of t, not at a single time as in Section 3.3 and Appendices C.4–C.7. In the context of histograms (Section 1.4) the measured quantity becomes the independent variable, and the dependent variable is an indicator marking the presence of the measurement there. In both cases the measurement can be thought of as distributed over an interval, not just at a point.

In this case the data cell array would be augmented by the inclusion of a window function, indicating the variation of the instrumental sensitivity:

$$x = \{x_n, t_n, w_n(t - t_n)\} \;\; n = 1, 2, \ldots, N, \tag{C66}$$

where $w_n(t)$ describes, for the value reported as X_n, the relative weights assigned to times near t_n.

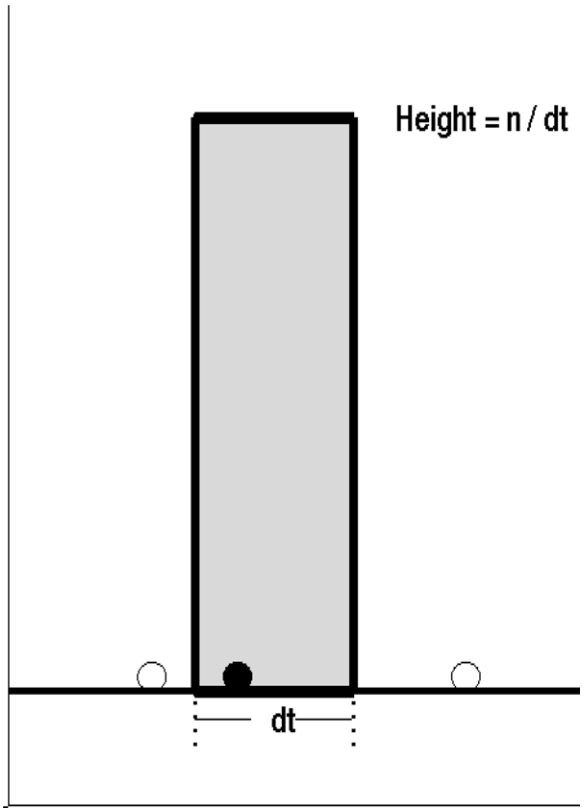

Figure 12. Voronoi cell of a photon. Three successive photon detection times are circles on the time axis. The vertical dotted lines underneath delineate the time extent (dt) of the cell and the height of the rectangle—n/dt, where n is the number of photons at exactly the same time (almost always 1)—is the local estimate of the signal amplitude. If the exposure at this time is less than unity, the width of the rectangle shrinks in proportion, the area of the rectangle is preserved, so the height increases in inverse proportion, yielding a larger estimate of the true event rate.

This is a nontrivial complication if the window functions overlap, but can nevertheless be handled with the same technique.

We assume the standard piecewise constant model of the underlying signal, that is, a set of contiguous blocks:

$$B(x) = \sum_{j=1}^{N_b} B^{(j)}(x), \tag{C67}$$

where each block is represented as a *boxcar* function:

$$B^{(k)}(x) = \begin{cases} B_j & \zeta_j \leqslant x \leqslant \zeta_{j+1} \\ 0 & \text{otherwise} \end{cases} \tag{C68}$$

the ζ_j are the change points, satisfying

$$\min(x_n) \leqslant \zeta_1 \leqslant \zeta_2 \leqslant \cdots \zeta_j \leqslant \zeta_{j+1} \leqslant \cdots \leqslant \zeta_{N_b} \leqslant \max(x_n) \tag{C69}$$

and the B_j are the heights of the blocks.

The value of the observed quantity, y_n, at x_n, under this model is

$$\begin{aligned} \hat{y}_n \;\; &= \int w_n(x) B(x) dx \\ &= \int w_n(x) \textstyle\sum_{j=1}^{N_b} B^{(j)}(x) dx \\ &= \textstyle\sum_{j=1}^{N_b} \int w_n(x) B^{(j)}(x) dx \\ &= \textstyle\sum_{j=1}^{N_b} B_j \int_{\zeta_j}^{\zeta_{j+1}} w_n(x) dx, \end{aligned} \tag{C70}$$

so we can write

$$\hat{y}_n = \sum_{j=1}^{N_b} B_j G_j(n), \tag{C71}$$

where

$$G_j(n) \equiv \int_{\zeta_j}^{\zeta_{j+1}} w_n(x)dx \tag{C72}$$

is the inner product of the nth weight function with the support of the jth block. The analysis in Bretthorst (1988) shows how to deal with the non-orthogonality that is generally the case here. (If the weighting functions are delta functions, it is easy to see that $G_j(n)$ is non-zero if and only if x_n lies in block j, and since the blocks do not overlap the product $G_j(n)G_k(n)$ is zero for $j \neq k$, yielding orthogonality, $\sum_N G_j(n)G_k(n) = \delta_{j,k}$. And of course there can be some orthogonal blocks, for which there happens to be no "spill over," but these are exceptions.)

The averaging process in this data model induces dependence among the blocks. The likelihood, written as a product of likelihoods of the assumed independent data samples, is

$$P(\text{Data}|\text{Model}) = \prod_{n=1}^{N} P(y_n|\text{Model}) \tag{C73}$$

$$= \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi\sigma_n^2}} e^{-\frac{1}{2}(\frac{y_n - \hat{y}_n}{\sigma_n})^2} \tag{C74}$$

$$= \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi\sigma_n^2}} e^{-\frac{1}{2}\left(\frac{y_n - \sum_{j=1}^{N_b} B_j G_j(n)}{\sigma_n}\right)^2} \tag{C75}$$

$$= Q e^{-\frac{1}{2}\left(\frac{y_n - \sum_{j=1}^{N_b} B_j G_j(n)}{\sigma_n}\right)^2}, \tag{C76}$$

where

$$Q \equiv \prod_{n=1}^{N} \frac{1}{\sqrt{2\pi\sigma_n^2}}. \tag{C77}$$

After more algebra and adopting a new notation, symbolized by

$$\frac{y_n}{\sigma_n^2} \rightarrow y_n \tag{C78}$$

and

$$\frac{G_k(n)}{\sigma_n^2} \rightarrow G_k(n), \tag{C79}$$

we arrive at

$$\log P(\{y_n\}|B) = Q e^{-\frac{H}{2}}, \tag{C80}$$

where

$$H \equiv \sum_{n=1}^{N} y_n^2 - 2\sum_{j=1}^{N_b} B_j \sum_{n=1}^{N} y_n G_j(n) + \sum_{j=1}^{N_b}\sum_{k=1}^{N_b} B_j B_k \sum_{n=1}^{N} G_j(n)G_k(n). \tag{C81}$$

The last two equations are equivalent to Equations (3.2) and (3.3) of Bretthorst (1988), so that the orthogonalization of the basis functions and the final expressions follow exactly as in that reference.

C.9. Piecewise Linear Model: Event Data

Here we outline the computations of a fitness function for the piecewise linear model in the case of event data. This means that the event rate for a block is assumed to be linear, as in Equation (1).

For convenience we take the fiducial time t_0 to be t_2, the time at the end of the block. Take t_1 to be the time at the beginning, so $M = t_2 - t_1$ is the length of the block, and the signal x is $\lambda(1 - aM)$ at the beginning of the block and λ at the end, and varies linearly in between.

The block likelihood for the case of event data t_i is

$$L(\lambda, a) = \sum_{i=1}^{N_k} \log[\lambda(1 + a(t_i - t_2))] - \int_{t_1}^{t_2} \lambda(1 + a(t - t_2))dt, \tag{C82}$$

where the sum is over the N_k events in the block and the integral is over the time interval covered by the block. Simplifying we have

$$L(\lambda, a) = N_k \log\lambda + \sum_{i=1}^{N_k} \log[(1 + a(t_i - t_2))] - \lambda\left[(1 - at_2)t + \frac{a}{2}t^2\right]_{t_1}^{t_2} \tag{C83}$$

$$L(\lambda, a) = N_k \log\lambda + \sum_{i=1}^{N_k} \log[(1 + a(t_i - t_2))] - \lambda M_k\left(1 - \frac{a}{2}M_k\right). \tag{C84}$$

Now let us compute the maximum likelihood as a function of λ and a, starting by setting

$$\frac{\partial L}{\partial \lambda} = \frac{N_k}{\lambda} - M_k\left(1 - \frac{a}{2}M_k\right) = 0 \tag{C85}$$

so that at the maximum of this likelihood we have

$$\lambda = \frac{N_k}{M_k\left(1 - \frac{a}{2}M_k\right)} \tag{C86}$$

and therefore

$$L(\lambda_{\max}, a) = N_k \log\left[\frac{N_k}{M_k\left(1 - \frac{a}{2}M_k\right)}\right] + \sum_{i=1}^{N_k} \log[(1 + a(t_i - t_2))] - N_k \tag{C87}$$

$$\frac{\partial L}{\partial a} = N_k \log\left[\frac{N_k}{M_k\left(1 - \frac{a}{2}M_k\right)}\right] + \sum_{i=1}^{N_k} \log[(1 + a(t_i - t_2))] - N_k \tag{C88}$$

$$\frac{\partial L}{\partial a} = \sum_{i=1}^{N_k} \frac{(t_i - t_2)}{1 + a(t_i - t_2)} + \frac{\lambda}{2}M_k^2 = 0 \tag{C89}$$

$$\frac{1}{N_k}\sum_{i=1}^{N_k}\frac{(t_i-t_2)}{1+a(t_i-t_2)}+\frac{\frac{1}{2}M_k}{\left(1-\frac{a}{2}M_k\right)}=0 \tag{C90}$$

$$f(a)=\frac{1}{N_k}\sum_{i=1}^{N_k}\frac{(t_i-t_2)}{1+a(t_i-t_2)}+\frac{\frac{1}{2}M_k}{\left(1-\frac{a}{2}M_k\right)} \tag{C91}$$

$$f'(a)=-\frac{1}{N_k}\sum_{i=1}^{N_k}\frac{(t_i-t_2)^2}{[1+a(t_i-t_2)]^2}-\frac{\frac{1}{4}M_k^2}{\left(1-\frac{a}{2}M_k\right)^2} \tag{C92}$$

$$\lambda=-\frac{2}{M_k^2}\sum_{i=1}^{N_k}\frac{(t_i-t_2)}{1+a(t_i-t_2)} \tag{C93}$$

$$\frac{N_k}{\left(1-\frac{a}{2}M_k\right)}=-\frac{2}{M_k}\sum_{i=1}^{N_k}\frac{(t_i-t_2)}{1+a(t_i-t_2)} \tag{C94}$$

$$\frac{\left(1-\frac{a}{2}M_k\right)}{N_k}=-\frac{M_k}{2\sum_{i=1}^{N_k}\frac{(t_i-t_2)}{1+a(t_i-t_2)}} \tag{C95}$$

$$1-\frac{a}{2}M_k=-\frac{1}{2}M_kN_k\left(\sum_{i=1}^{N_k}\frac{(t_i-t_2)}{1+a(t_i-t_2)}\right)^{-1} \tag{C96}$$

$$a=\frac{2}{M_k}+N_k\left(\sum_{i=1}^{N_k}\frac{(t_i-t_2)}{1+a(t_i-t_2)}\right)^{-1}. \tag{C97}$$

C.10. Piecewise Exponential Model

In this case we model the signal as varying exponentially across the time interval contained in the block, as in Equation (2).

That is to say, if $M=t_2-t_1$ is the length of the block, the signal is λe^{-aM} at the beginning of the block and λ at the end.

The block likelihood for the case of event data t_i is

$$L(\lambda,a)=\sum_{i=1}^{N_k}\log[\lambda e^{a(t_i-t_2)}]-\int_{t_1}^{t_2}\lambda e^{a(t-t_2)}dt, \tag{C98}$$

where the sum is over the N_k events in the block and the integral is over the time interval covered by the block. For convenience we take the fiducial time t_2 (at which the signal is equal to λ) to be the end of the block, and t_1 is therefore the beginning.

This expression can be simplified:

$$L(\lambda,a|B)=N_k\log\lambda+a\sum_i(t_i-t_2)-\lambda\left[\frac{e^{a(t-t_2)}}{a}\right|_{t_1}^{t_2}dt \tag{C99}$$

$$L(\lambda,a|B)=N_k\log\lambda+a\sum_i(t_i-t_2)-\lambda\left(\frac{1-e^{-aM}}{a}\right). \tag{C100}$$

Now let us compute the maximum likelihood as a function of λ and a:

$$\frac{\partial L}{\partial\lambda}=\frac{N_k}{\lambda}-\left(\frac{1-e^{-aM}}{a}\right) \tag{C101}$$

and therefore at the maximum we have

$$\lambda=\frac{aN_k}{1-e^{-aM}} \tag{C102}$$

$$\frac{\partial L}{\partial a}=\sum_i(t_i-t_2)-[N_k(1-e^{-aM})^{-1}][(M+a^{-1})e^{-aM}-a^{-1}] \tag{C103}$$

$$L_{\max}(a)=N_k\log\left(\frac{aN_k}{1-e^{-aM}}\right)+a\sum_i(t_i-t_2)-\frac{aN_k}{1-e^{-aM}}\left(\frac{1-e^{-aM}}{a}\right) \tag{C104}$$

$$L_{\max}(a)=N_k\log\left(\frac{aN_k}{1-e^{-aM}}\right)+a\sum_i(t_i-t_2)-N_k \tag{C105}$$

$$\frac{\partial L_{\max}(a)}{\partial a}=N_k\left(\frac{1-e^{-aM}}{aN_k}\right)Q+\sum_i(t_i-t_2), \tag{C106}$$

where

$$Q=N_k[(1-e^{-aM})^{-1}-a(1-e^{-aM})^{-2}Me^{-aM}] \tag{C107}$$

$$\frac{\partial L_{\max}(a)}{\partial a}=\frac{N_k}{a}-MN_k\frac{e^{-aM}}{(1-e^{-aM})}+\sum_i(t_i-t_2). \tag{C108}$$

To solve for the value of a that makes this derivative zero (to find the maximum of the likelihood) we will use Newton's method to find the zeros of

$$f(a)=\frac{\partial L_{\max}(a)}{\partial a}/N_k=\frac{1}{a}-Me^{-aM}(1-e^{-aM})^{-1}+S, \tag{C109}$$

where

$$S=\frac{1}{N_k}\sum_i(t_i-t_2) \tag{C110}$$

is the mean of the differences between the event times and the time at the end of the block. The iterative equation is

$$a_{k+1}=a_k-\frac{f(a_k)}{f'(a_k)}, \tag{C111}$$

and since S is a constant we have

$$f'(a)=-\frac{1}{a^2}-M[-Me^{-aM}(1-e^{-aM})^{-1}-Me^{-aM}(1-e^{-aM})^{-2}e^{-aM}] \tag{C112}$$

$$f'(a)=-\frac{1}{a^2}+M^2e^{-aM}(1-e^{-aM})^{-1}[1+e^{-aM}(1-e^{-aM})^{-1}], \tag{C113}$$

and defining

$$Q(a)=e^{-aM}(1-e^{-aM})^{-1}, \tag{C114}$$

we have

$$f'(a)=-\frac{1}{a^2}+M^2Q(a)[1+Q(a)] \tag{C115}$$

and

$$a_{k+1}=a_k-\frac{a_k^{-1}-MQ(a_k)+S}{-a_k^{-2}+M^2Q(a_k)[1+Q(a_k)]}. \tag{C116}$$

REFERENCES

Arias-Castro, E., Donoho, D., & Huo, X. 2005, ITIT, 51, 2402

Arlot, S., & Celisse, A. 2010, Statistics Surveys, 4, 40, (http://www.di.ens.fr/sierra/pdfs/2010_Arlot_Celisse_SS.pdf)

Band, D. 2002, ApJ, 578, 806

Bellman, R. 1961, Commun. ACM, 4, 284

Bretthorst, G. L. 1988, Bayesian Spectrum Analysis and Parameter Estimation (Lecture Notes in Statistics; Berlin: Springer)

Capra, F. 2007, The Science of Leonardo (New York: Doubleday)

Cash, W. 1979, ApJ, 228, 939

Claerbout, J. 1990, Active Documents and Reproducible Results, Stanford Exploration Project Report, 67, 139

Coram, M. 2002, PhD thesis, Stanford University

Coram, M., & Lalley, S. P. 2006, AnSta, 34, 1233

Diaconis, P., & Freedman, D. 1993, AnSta, 21, 2108

Diaconis, P., & Freedman, D. 1995, Probab. Math. Stat., 15, 243

Donoho, D. 1994, in Recent Advances in Wavelet Analysis, Smooth Wavelet Decompositions with Blocky Coefficient Kernels, ed. L. Schumaker & G. Webb (San Diego, CA: Academic), 259

Donoho, D., & Johnstone, I. 1998, AnSta, 26, 879

Donoho, D., Maleki, A., Rahman, I., Shahram, M., & Stodden, V. 2009, CSE, 11, 8

Dreyfus, S. 2002, Oper. Res., 50, 48

Du, C., & Kou, S. 2012, JSM 2012 Online Program (Alexandria, VA: American Statistical Association), abstract #306040

Efron, B., & Tibshirani, R. 1998, An Introduction to the Bootstrap (New York: CRC Press)

Fenimore, E., Palmer, D., Galassi, M., et al. 2001, in AIP Conf. Proc. 662, Gamma-Ray Burst and Afterglow Astronomy 2001, The Trigger Algorithm for the Burst Alert Telescope on Swift, ed. G. R. Ricker & R. K. Vanderspek (Melville, NY: AIP), 491

Gelman, A., Carlin, J., Stern, H., & Rubin, D. 1995, Bayesian Data Analysis (London: Chapman and Hall)

Gregory, P., & Loredo, T. 1992, ApJ, 398, 146

Hogg, D. W. 2008, arXiv:0807.4820

Horvath, I., Norris, J. P., Scargle, J. D., & Balázs, L. G. 2005, NCimC, 28, 291

Hubert, L., Arabie, P., & Meulman, J. 2001, Combinatorial Data Analysis: Optimization by Dynamic Programming (Philadelphia, PA: SIAM)

Jackson, B., Scargle, J. D., Barnes, D., et al. 2005, ISPL, 12, 105

Lin, J., Keogh, E., Lonardi, S., & Chiu, B. 2003, DMKD '03 Proc. of the 8th ACM SIGMOD Workshop on Research Issues in Data Mining and Knowledge Discovery, ed. M. J. Zaki & C. C. Aggarwal (New York: ACM), 2

Mannila, H., & Salmenkivi, 2001, in Proc. Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ed. F. Provost & R. Srikant (New York: ACM), 341

McLean, K., Fenimore, E., Palmer, D., et al. 2004, in AIP Conf. Proc. 727, Gamma-Ray Bursts: 30 Years of Discovery, ed. E. Fenimore & M. Galassi (Melville, NY: AIP), 667

Norris, J., Gehrels, N., & Scargle, J. 2010, ApJ, 717, 411

Norris, J., Gehrels, N., & Scargle, J. 2011, ApJ, 735, 23

Okabe, A., Boots, B., Sugihara, K., & Chiu, S. N. 2000, Spatial Tessellations: Concepts and Applications of Voronoi Diagrams (2nd ed.; New York: Wiley)

Ò Ruanaidh, J. J., & Fitzgerald, W. J. 1996, Numerical Bayesian Methods Applied to Signal Processing (New York: Springer)

Papoulis, A. 1965, Probability, Random Variables, and Stochastic Processes (New York: McGraw-Hill)

Prahl, J. 1999, arXiv:astro-ph/9909399

Qin, Y., Liang, E-W., Yi, S-X., et al. 2013, ApJ, 763, 15

Scargle, J. 1998, ApJ, 504, 405

Scargle, J. 2001a, in AIP Conf. Proc. 567, Bayesian Inference and Maximum Entropy Methods in Science and Engineering, 19th International Workshop, ed. J. Rychert, G. Erickson, & R. Smith (Melville, NY: AIP), 245

Scargle, J. 2001b, in Bayesian Blocks in Two or More Dimensions: Image Segmentation and Cluster Analysis, Contribution to Workshop on Bayesian Inference and Maximum Entropy Methods in Science and Engineering (MAXENT 2001) (Baltimore, MD: Johns Hopkins University), arXiv:math/0111128

Schmidt, M. 2000, in AIP Conference Proc. 526, Derivation of a Sample of Gamma-Ray Bursts from BATSE DISCLA Data, ed. R. M. Kippen (Melville, NY: AIP), 8

Tompkins, W. 1999, PhD thesis, Stanford University (http://arxiv.org/pdf/astro-ph/0202141v1.pdf)

Tong, H. 1990, Non-Linear Time Series: A Dynamical System Approach (Oxford: Oxford Univ. Press)

Way, M., Gazis, P., & Scargle, J. 2011, ApJ, 727, 48

Xie, Y., Huang, J., & Willett, R. 2012, in Proc. 2012 IEEE Statistical Signal Processing Workshop (Ann Arbor, MI: IEEE), 60