Time Series Explorer

Jeff Scargle
NASA Ames Research Center / San Jose State University

1 Scope of the Time Series Explorer

A central theme of this conference is the need for advanced algorithms for automated analysis of massive amounts of time series data. To address this problem Tom Loredo and I are developing the *Time Series Explorer* – an analysis toolkit and automated pipeline. Here I discuss a few algorithms for this system, to be described in more detail elsewhere. The underlying goals are to detect and characterize periodicities, correlations, time delays, random activity, transient events, and other astronomically interesting features, and deliver these results in ways suitable for subsequent exploration, machine learning, data mining, and visualization. Application contexts range from exploratory data analysis, such as combing massive amounts of time series for unknown signals, through projects tightly targeted on specific measurements.

Desiderata for such tools include ability to handle the variety of data modes and irregular sampling characteristic of modern astronomy. Suitability for automated analysis, providing inputs to machine learning, data mining, and visualization systems, and other interface issues are less well defined but are subjects of much current research. Issues of computational efficiency will not be discussed here.

At the core of this setting is construction and processing of data structures to represent the information content of observations, measurements, or computations. One approach is this sequence, each step operating on the results of the previous one and feeding information to the subsequent one:

1 Represent the raw data in terms of some measure of intensity

2 Agglomerate and/or smooth item 1

3 Identify statistically significant time-domain features in item 2

4 Render item 3 into astrophysical meaningful features

5 Present item 4 in a form suited for the context

Examples of contexts in item 5 are machine learning, data mining, publication, archiving and visualization. The following sections briefly sketch a few key algorithmic ideas for the Time Series Explorer: non-parametric time-domain models, periodograms, and correlation functions. In all cases the representation in item 1 consists of assigning a data cell to each measurement [9] (including time-tags for individual photons)

2 Optimal Histograms as Time-Domain Models

We start with a problem that seems to have no connection with time series analysis, namely representation of the probability distribution of some quantity from repeated measurements. The goal is to quantify features of the distribution such as mean value, variance, or skewness. The standard approach is to count the number of times the measurements falls within each member of a set of pre-selected evenly spaced intervals. Figure 1 demonstrates the serious problem that the resulting histogram is quite dependent on the choice of these bins. Shown are results based on one principle or another for fixing the number of bins. In practice the choice is almost always arbitrary with some degree of adjustment to bring out desired sought-after features.

In fact these data are photon arrival times take from GRB 551 at
`ftp://legacy.gsfc.nasa.gov/compton/data/batse/ascii_data/batse_tte/`
Optimal generation of histograms and deriving light curves from photon time of arrival measurements are essentially identical problems [9]! Data-adaptive histograms (of the same data) shown in Figure 2 use the Bayesian Blocks algorithm [9]. The bins are not fixed in number or constrained to be evenly spaced. The figure shows two different cases to emphasize the slight but noticeable dependence on the value of the one parameter of the algorithm, from the prior for the number of bins. The solid line is based on fixing the false positive rate in random noise at 5%, but is very insensitive to this value. The dashed line adjusts for the presence of a signal as described in [9]; this captures the smaller pulse at 2.3 seconds at the cost of introducing a probably spurious one at around 0.7 seconds. Even if spurious the latter pulse has very little area and thus would not much affect many post-processing results. The ideas behind the histograms shown here and other attempts at codifying the choice of bins are discussed in [5] and at the companion website
`http://www.astroml.org/book_figures/chapter5/fig_hist_binsize.html` .

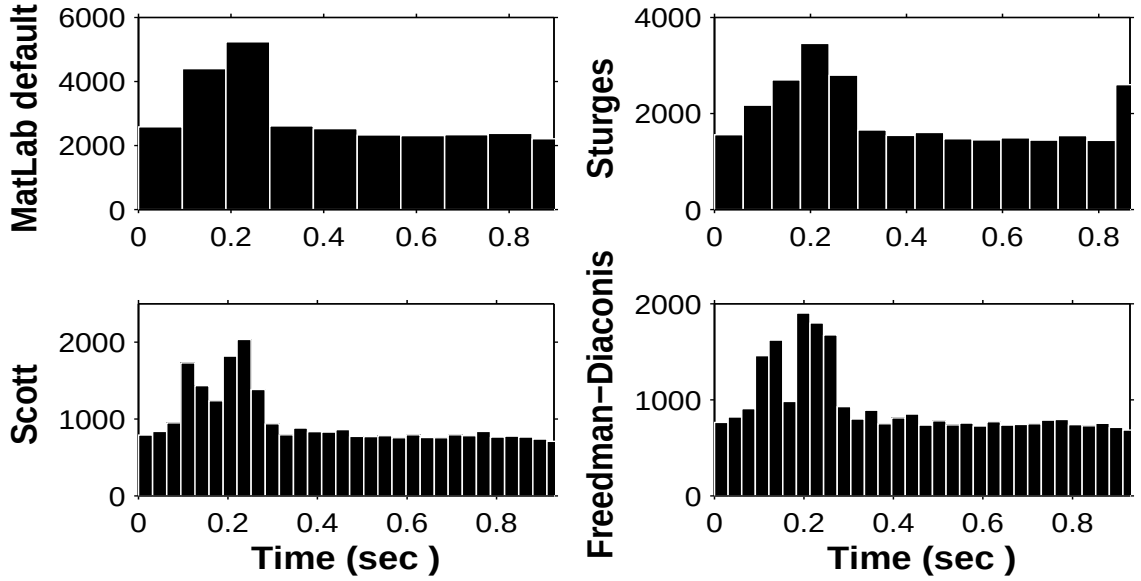

Figure 1: Histograms of 28,904 measurements with four different bin rules: (a) Matlab's default (10 bins). (b) Sturges: $1 + log_2 N$ bins [12] (c) Scott: bins size = $\frac{3.49\sigma}{N^{1/3}}$ [11] (d) Freedman-Diaconis: bins size = $\frac{2IQR}{N^{1/3}}$; IQR is the interquartile range [4].

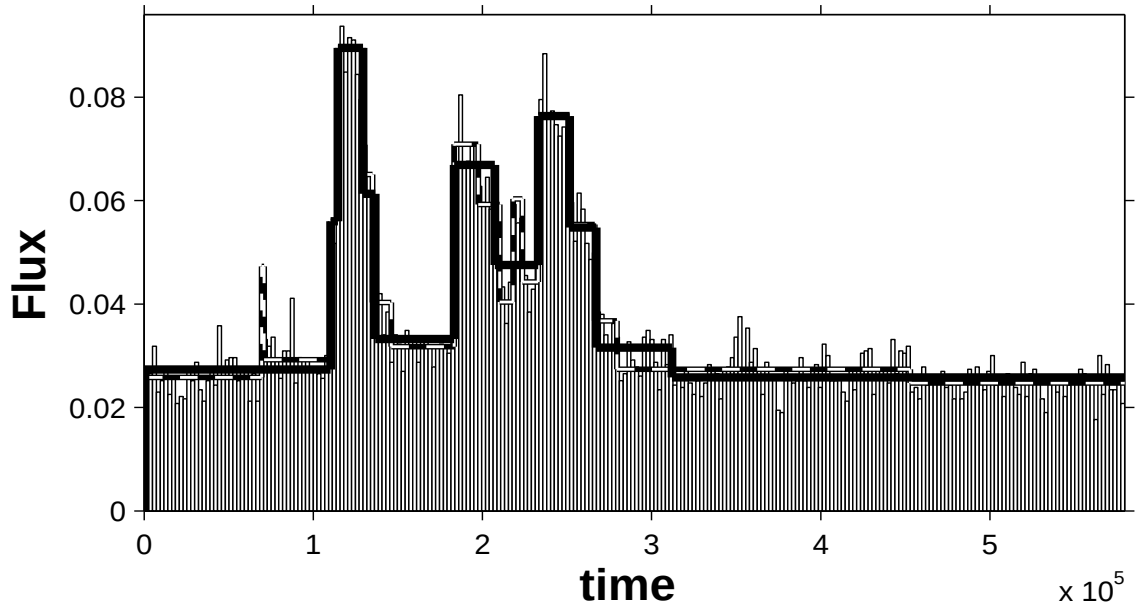

Figure 2: Bayesian Block histograms of the same data as in Fig. 1. Solid lines, and dashed sometimes hidden by the former, are with two different values of the bin-number prior parameter. Cf. the light histogram with evenly spaced bins.

3 Periodicities

In many of the contexts discussed in this conference detection of periodic signals in noise is of great importance. This topic has a huge literature. Here I only want to mention a very simple approach that does not seem to have been utilized very much in astronomy. The idea is simply to estimate the autocorrelation function using the Edelson and Krolik algorithm [2] for arbitrarily spaced data, and then apply the fast Fourier transform to yield an estimate of the power spectrum. These estimates are known as "slotted" correlation functions and power spectra in other fields (e.g. [1], apparently unaware this idea in astronomy, but making comparisons with a popular astronomical tool [6, 7, 8] for detecting periodicities). As demonstrated in [10] this approach is valuable for estimation of cross-correlation functions, cross-spectra, plus time-frequency and time-scale distributions.

As in many cases implementation of the grand idea is almost trivial but difficulties lie in details and small practical matters. While there is no pre-determined binning in time, one must establish bins in the *time lag* that need to be evenly spaced if the FFT is used to estimate the power spectrum. This fact necessitates care in bin selection and opens up the very dependence decried in Section 1. Depending on the sampling some bins can be empty of the cross-products fundamental to the Edelson and Krolik part of the algorithm; these can be handled by simple interpolation with little difficulty. Finally the power estimated with this technique is not guaranteed to be positive. This is somewhat rare, and can be ameliorated by simply taking the absolute value of the estimate. This ad hoc scheme seems to work well in simulations, but I do not know of a theoretical justification.

4 HOPing through the Time Domain

In this section we investigate the possible use for time series analysis of an algorithm developed for another domain entirely – topological analysis of density distributions. The group-finding algorithm **HOP** [3], developed to characterize the distribution of galaxies, is quite general and applies in any dimension. For any function $\mathbf{f}$ and adjacencies defined for objects in a set $\mathbf{S}$, it yields a unique, parameter-free partition of $\mathbf{S}$ into groups – one for each local maxima of $\mathbf{f}$, each being a connected set such that $\mathbf{f}$ decreases monotonically away from the maximum. In short the algorithm finds all of the peaks of $\mathbf{f}$ in $\mathbf{S}$ and the connected structures flowing from them – mountain peaks and their watersheds. The basic idea of **HOP** is a simple hill climbing prescription, associating each object with its neighbors that have larger values of $\mathbf{f}$. In other words, given a set $\mathbf{S}$ of spatially distributed objects assign a value of $\mathbf{f}$ to each one and identify the objects ("neighbors") adjacent to it. Then iteratively replace the index of each object with that of its neighbor with the largest value of $\mathbf{f}$. Rapid

convergence is obtained when no index value changes and each object is associated with a local maximum of **f**. Now take the objects to be individual photons (instead of galaxies) described by a set of detection times t_i, and define the function for each such event as

$$f_i = \frac{2}{t_{i+1} - t_{i-1}} \tag{1}$$

This quantity is a convenient, if noisy, estimate of the intensity at the time t_i. **HOP** identifies peaks in the light curve and the photons naturally associated with them.

A plot of the intensity averaged over the groups obtained in this way is typically too noisy to be very useful. However applying **HOP** to a new time series where these output groups are treated as input objects yields a less noisy, more or less down-sampled representation. This iterative version, **IHOP**, is demonstrated in Fig. 3. This straightforward approach can obviously be applied to tasks such as pulse-hunting in gamma ray bursts and may well have further use in the same context as in Section 2.

I am grateful to Tom Loredo for his collaboration in this project.

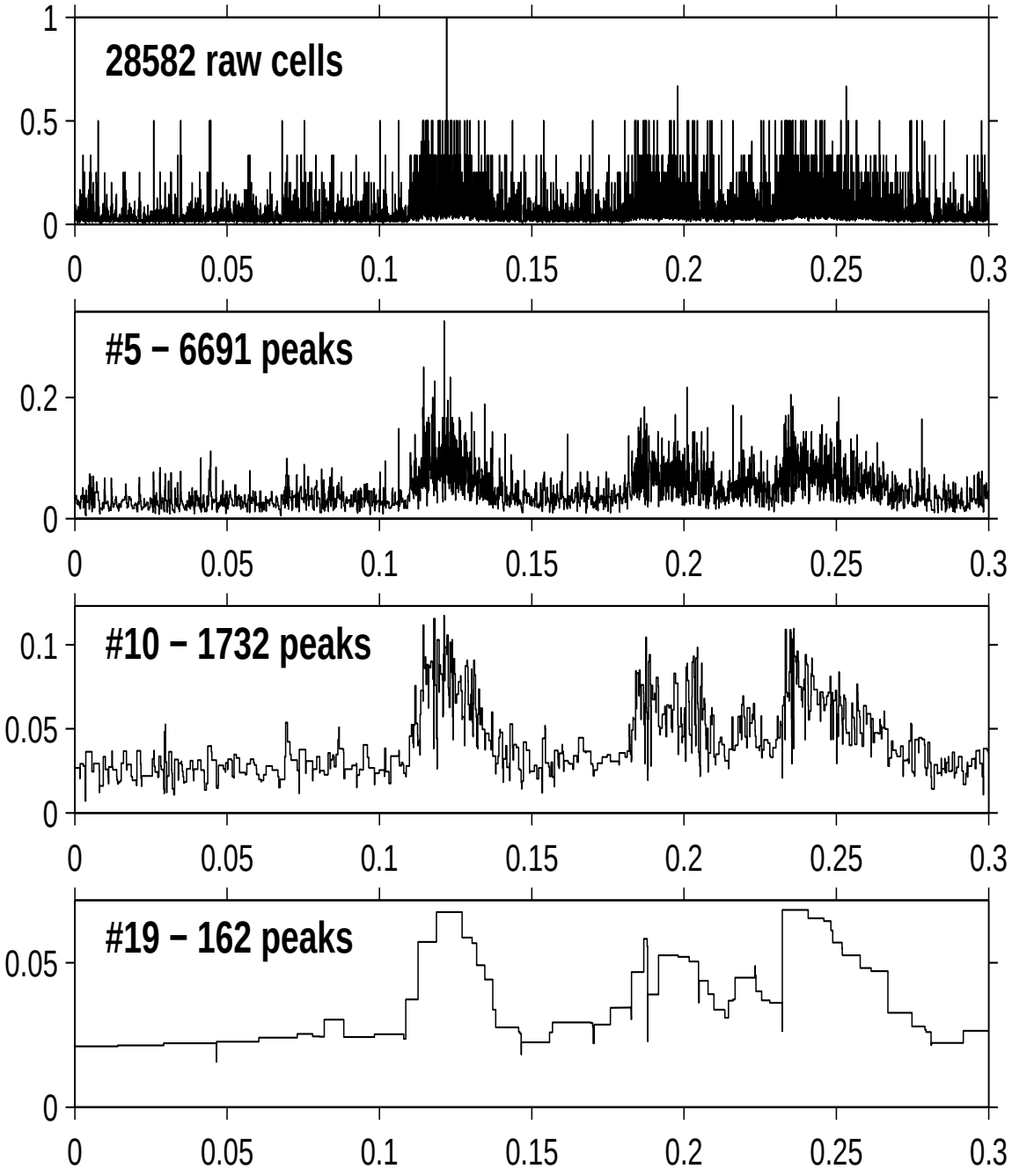

Figure 3: Raw data (top panel) represented as the valuse of f_i in Eq. (1) for the intervals between photons [9]. The next three panels are after 5, 10 and 19 iterations of IHOP.

References

[1] P. M.T. Broerson, S. de Waele and R. Bos, The Accuracy of Time Series Analysis for Laser-Doppler Velocimetry, in 10th International Symposium on Application of Laser Techniques to Fluid Mechanics, Lisbon, July 10-13, 1 (2000)

[2] R. Edelson and J. Krolik, J. H. 1988, ApJ 333, 646 (1988)

[3] D. Eisenstein, and P. Hut, Ap.J. 498, 137 (1998)

[4] D. Freedman, P. Diaconis, Probability Theory and Related Fields, **57**, 453 (1981)

[5] Z. Ivezic, A. Connolly, J. VanderPlas and A. Gray, Statistics, Data Mining, and Machine Learning in Astronomy, (Princeton University Press, 2014)

[6] N. Lomb, Ap&SS **39**, 447 (1976).

[7] J. Scargle, J. ApJ, **263**, 835 (1982).

[8] J. Scargle, ApJ, **343**, 874 (1989).

[9] J. Scargle, J. Norris, B. Jackson, J. Chiang, ApJ, **764**, 167 (2013).

[10] J. Scargle, S. Keil, and S. P. Worden, ApJ, **771**, 33 (2013).

[11] D. W. Scott, WIREs Computational Statistics **2**, 497 (2010).

[12] H. A. Sturges, Journal of the American Statistical Association, **65**, 66 (1926)