



Published in final edited form as:

*J Biol Rhythms*. 2020 April ; 35(2): 214–222. doi:10.1177/0748730419900866.

## CIRCADA: Shiny Apps for Exploration of Experimental and Synthetic Circadian Time Series with an Educational Emphasis

Lisa Cenek<sup>(1),\*</sup>, Liubou Klindziuk<sup>(1),\*</sup>, Cindy Lopez<sup>(1),\*</sup>, Eleanor McCartney<sup>(2)</sup>, Blanca Martin Burgos<sup>(2)</sup>, Selma Tir<sup>(2)</sup>, Mary E. Harrington<sup>(2)</sup>, Tanya L. Leise<sup>(1)</sup>

<sup>(1)</sup>Mathematics and Statistics, Amherst College, Amherst, MA 01002

<sup>(2)</sup>Neuroscience Program, Smith College, Northampton MA 01063

### Abstract

Circadian rhythms are daily oscillations in physiology and behavior that can be assessed by recording body temperature, locomotor activity, or bioluminescent reporters, among other measures. These different types of data can vary greatly in waveform, noise characteristics, typical sampling rate, and length of recording. We developed two Shiny apps for exploration of these data, enabling visualization and analysis of circadian parameters like period and phase. Methods include the discrete wavelet transform, sine-fitting, the Lomb-Scargle periodogram, autocorrelation, and maximum entropy spectral analysis, giving a sense of how well each method works on each type of data. The apps also provide educational overviews and guidance for these methods, supporting the training of those new to this type of analysis. CIRCADA-E (Circadian App for Data Analysis - Experimental Time Series) allows users to explore a large curated experimental dataset with mouse body temperature, locomotor activity, and PER2::LUC rhythms recorded from multiple tissues. CIRCADA-S (Circadian App for Data Analysis - Synthetic Time Series) generates and analyzes time series with user-specified parameters, thereby demonstrating how accuracy of period and phase estimation depends on the type and level of noise, sampling rate, length of recording, and method. We demonstrate potential uses of the apps through two *in silico* case studies.

### Keywords

circadian rhythms; biological oscillations; data analysis; discrete wavelet transform; periodogram; mathematical analyses; Shiny app

### Introduction

Circadian rhythms are daily oscillations in physiology and behavior that continue in the absence of time cues and that have been widely observed in bacteria, archaea, fungi, plants, and animals (Hurley et al., 2016). In mice, for example, these rhythms can be observed in body temperature, locomotor activity, bioluminescent reporters of gene expression, and other measures (Leise et al., 2018). These different types of data vary greatly in waveform, noise

\*Co-first authors who contributed equally

The Authors declare that there is no conflict of interest.

characteristics, sampling rate, and length of recording. For instance, activity recordings often have high sampling rates, long durations, and square-like waveforms, while bioluminescence recordings tend to be sinusoid-like and only a few days in length. We developed two Shiny apps for exploration of these circadian data, enabling visualization and analysis of circadian parameters like period and phase. Shiny is an R package supporting the creation of open-source interactive web apps (<https://shiny.rstudio.com/>). The motivation behind our apps is two-fold: first, to create accessible tools for exploration and analysis of multiple forms of circadian data; second, to develop an education resource for circadian data analysis. The original apps were developed by the three undergraduate authors LC, LK, and CL at Amherst College with extensive feedback and testing provided by undergraduates at Smith College, with the app development itself providing multiple valuable educational opportunities, including learning how to code in R and Shiny, how to communicate across the quantitative and life sciences, and how to create effective visualizations of data.

Five widely used estimation methods are included in the apps as a representative set of distinct analysis tools: the discrete wavelet transform (DWT), sine-fitting, the Lomb-Scargle periodogram, autocorrelation, and maximum entropy spectral analysis (see Table 1 and the Methods). The focus is on methods applicable to recordings of body temperature, locomotor activity, and bioluminescent reporters, such as PER2::LUC, with at least 3 days of data sampled at least once per hour. The analyses considered here are not aimed at, for instance, detection of circadian rhythms in transcriptome or proteome data, which tend to have short duration and sparse sampling and for which appropriate tools are freely available, including JTK\_Cycle (Hughes et al., 2010), RAIN (Thaben and Westermark, 2014), MetaCycle (Wu et al., 2016), and LimoRhyde (Singer and Hughey, 2019), among others. Our educational apps complement existing circadian data analysis apps, such as Biodare (Zielinski et al., 2014), CATkit (Gierke and Cornelissen, 2016), Rethomics (Geissmann et al., 2019) and RhythmicAlly (Abhilash and Sheeba, 2019), by training users to be more knowledgeable in how best to use and interpret different types of analyses for a variety of data types.

The five methods use quite different approaches to estimating rhythmic characteristics of time series and produce different subsets of the desired information about the time series, as shown in Table 2. All five methods generate estimates of the period (cycle length), but not all estimate the phase (reported as the peak time) and the amplitude (defined as half the peak-to-trough distance). The Lomb-Scargle method provides a p-value for the significance of the dominant periodicity, and autocorrelation provides a measure of the strength of rhythmicity (rhythmicity index, RI). Some methods may have biases, depending on the waveform, trend, and type of noise, and the methods exhibit differences in accuracy of the estimates that depend on these time series characteristics. For example, sine-fitting assumes a symmetric waveform and so may produce a peak time estimate that is too late for a wave form that rises sharply then gradually decreases. These differences can sometimes be hard to predict, as the underlying theory generally assumes no trend and simple Gaussian noise. By providing results and visualizations of the application of these five methods to a variety of data types, the CIRCADA apps allow their effectiveness to be assessed for each case.

The first app, CIRCADA-E (Circadian App for Data Analysis - Experimental Time Series; see Supplemental Figure 1), allows users to explore a large set of data from (Leise et al.,

2018) with mouse body temperature, locomotor activity, and PER2::LUC recordings from multiple tissues (publicly accessible at <https://osf.io/seyhp/>). Samples from old and young mice under 24h and 20h T-cycles on normal and high-fat diets are available in this data. In our experience, finding large, interesting, public data sets in a usable form for undergraduate courses can be challenging. By directly interfacing a rich dataset with multiple methods of analysis and visualization in a user-friendly app, CIRCADA-E supports a variety of training and educational activities. For instance, students learning to work in a circadian lab can practice generating and testing hypotheses about circadian rhythms measures using this curated set of data with built-in analysis. Possible student projects include comparing distributions of body temperature periods for animals under the 24h vs 20h T-cycles, comparing period and peak time estimates for body temperature vs locomotor activity rhythms, or practicing analysis of results from 2×2 or 2×2×2 factorial design experiments by testing for differences in the rhythmicity index (RI) of the SCN *in vitro* rhythms or in the amplitude of the body temperature rhythms. The app also facilitates direct comparison of the results of different methods to assess which might work best for a particular type of data and how accurate and reliable those results may be. Visualization is a major feature of the apps, with graphs of raw data, processed data, and outputs of each method, encouraging regular visual checking of whether results appear consistent with the data.

The second app, CIRCADA-S (Circadian App for Data Analysis - Synthetic Time Series; see Supplemental Figure 2), generates and analyzes time series with user-specified parameters, listed in Table 3. CIRCADA-S can be used to explore how accuracy of period and phase estimation depends on sampling rate, length of recording, type and level of noise, and method used. This app also offers educational overviews and guidance for each method to provide further training support. As an example, Supplemental Figure 2 displays the CIRCADA-S Lomb-Scargle tab, which demonstrates how increasing the oversampling factor increases resolution of the period estimate and provides a link to a popup window with explanatory details about the method. Results from the five methods for a set of generated time series samples are summarized in the CIRCADA-S Summary tab, shown in Figure 1, including histograms of the period errors to facilitate comparison. For example, the DWT, autocorrelation, and Lomb-Scargle methods produce highly discretized period estimates, with the discrete values depending on the sampling interval length for DWT and autocorrelation and more heavily on the number of cycles for the Lomb-Scargle. In contrast, sine-fitting and MESA generate more fine-grained estimates with resolution independent of sampling interval and number of cycles, with sine-fitting estimates more tightly clustered compared to MESA. Note that higher resolution does not necessarily correlate with higher accuracy of period estimation. As another example, Supplemental Figure 3 shows part of the CIRCADA-S DWT tab with an interactive demonstration of the DWT time series decomposition (which is also available in CIRCADA-E). There is also an Advice tab with suggested guidelines for using each method and brief glossary of the user-set parameter options.

### Use in varied educational settings

To better understand how these apps might be used, we applied them in a variety of educational settings. In a very introductory class, CIRCADA-S was used within a classroom

demonstration to help students understand the Lomb-Scargle periodogram. In a more advanced course, CIRCADA-E was used by a student to address a research question about the interaction of light cycles on body temperature cycles. Within an undergraduate chronobiology research lab, all students completed an online exercise with brief quizzes to guide their exploration of CIRCADA-E. In a week-long chronobiology course for graduate students, CIRCADA-E was used to help students create simulated data with characteristics of their own data set, and results from the simulations were discussed in terms of how to select optimal data analysis tools. Students in all setting reported that this was a helpful tool and increased their understanding of methods to analyze rhythmic data. A sample online lesson, “Learning about the Analysis of Rhythmic Data,” which guides new users through the apps and methods, is posted on the OSF site (<https://osf.io/er6wj/>), with checkpoint questions at regular intervals to support self-testing for comprehension.

### ***In silico* case study: sampling rate versus duration**

To demonstrate the effect of sampling rate versus duration of the time series on period estimate accuracy, we generated 1,000 noisy sinusoids with sampling interval 0.125h for 16 days. The time series were subsampled to generate larger sampling intervals, and days dropped from the end to generate shorter durations. Results for the sine-fitting and Lomb-Scargle methods are shown in Figure 2 as examples. Each additional day in the time series tends to have a greater effect in increasing accuracy than does shortening the time step. For example, the period error in the sine-fit estimates is reduced by nearly 50% by going from 3 days to 5 days (requiring 36 vs 60 time points for 2h sampling), while increasing the sampling from once every 2 hours to 8 times an hour for 3 days (36 vs 576 time points) only reduces the error by about 16%.

### ***In silico* case study: statistical power**

CIRCADA-S can also be used to study how many samples might be necessary to achieve sufficient statistical power in an experiment, given a fixed duration and sampling interval. Sets of synthetic time series with characteristics selected to be similar to those of the experimental data can be generated to estimate the likely statistical power for different numbers of samples, for example, to detect a difference in period between two groups. See Figure 3 for an example showing that the addition of a cycle to the data can greatly strengthen the power and reduce the number of samples needed. Results of the app’s analysis are automatically downloaded to a file named CIRCADA-S-output.csv to facilitate this type of further exploration.

### **Running CIRCADA**

To run the apps, download and save the compressed CIRCADA folder from <https://osf.io/er6wj/> to your computer, then open CIRCADA-E.R or CIRCADA-S.R in RStudio and click “Run app”. All required packages will be installed as needed by running the app. A tutorial (GettingStarted.pdf) is available at <https://osf.io/er6wj/> for learning the features of the apps and includes suggestions for training activities and projects. An online lesson is also provided on the OSF site.

## Methods

The Shiny code for both apps is freely available at <https://osf.io/er6wj/>, with easily adaptable R code provided for each of the methods. Figures were generated using R 3.5.3 (<https://www.r-project.org/>) in RStudio 1.2.1335 (<https://www.rstudio.com/>). Experimental data used in CIRCADA-E are from (Leise et al., 2018), using methods as described in that paper where they overlap with those used in the app. The data is accessed from the Open Science Framework site <https://osf.io/seyh/> using the *osfr* R package (<https://github.com/CenterForOpenScience/osfr>). The R code used to analyze that data in batch form is also available in Rmarkdown format at <https://osf.io/seyh/>.

### Discrete wavelet transform (DWT):

Time series are detrended using the DWT before applying other methods (with the exception that linear detrending is used if time series cover 4 days or less). The apps use the function `wavMRDSum` from the *wmtsa* R package (<https://cran.r-project.org/web/packages/wmtsa/>) to calculate details, smooths, and their sums. The `s12` filter is used for the DWT, with the exception that `s8` is used if fewer than 120 time points are present. To avoid edge effects, portions are removed from each end after detrending, depending on the total length of the time series (8h if only 3 days; 12 if 4 days; 18 if 5–6 days; 36h if 7 or more days). Because some edge-effected intervals may still be present in the shorter time series, the DWT is more appropriate for time series with at least 5 cycles, preferably at least 7, so that edge effects can be fully removed while leaving multiple cycles for analysis. The DWT period estimate is the mean peak-to-peak time, and the peak time estimate is the circular mean of the peak times, converted to radians using the estimated period using the *CircStats* R package (<https://cran.r-project.org/web/packages/CircStats/>). The DWT amplitude estimate is the mean peak height.

### Autocorrelation:

The R command `acf` from the *stats* R package (<https://stat.ethz.ch/R-manual/R-devel/library/stats/html/00Index.html>) is used to calculate the autocorrelation of detrended time series. The maximum peak in the autocorrelation corresponding to a lag between 12h and 40h is used to estimate the period and rhythmicity index (RI), after first convolving with (1/8, 1/4, 1/4, 1/4, 1/8) to smooth the correlogram.

### Sine-fitting:

The R command `nls` (nonlinear least squares) from the *stats* R package is used to fit the three parameters of the sinusoidal curve to each time series. These parameters are then converted to amplitude, period, and phase angle for use in reporting estimates. Each fit is run 8 times for different initial parameter values and the best fit (smallest root mean square error, RMSE) is chosen.

### Lomb-Scargle periodogram:

The R command `lsp` is used to calculate the Lomb-Scargle periodogram, from the *lomb* R package (<https://cran.r-project.org/web/packages/lomb>). The highest peak corresponding in the period range 16–34h is chosen as the period estimate. Higher period resolution may be

obtained by increasing the oversampling factor, as shown in the periodogram in Supplemental Figure 2; oversampling factor 8 is used for the period estimate. In the CIRCADA-E app, significance of peaks is assessed using a randomization procedure via `randlsp`, which produces a more valid result when the theoretical assumptions may not hold, such as normally distributed uncorrelated noise. However, the randomization procedure is computationally intensive, so the CIRCADA-S app only runs it for a representative time series and uses the theoretical p-value for the full set of samples (which is a quick calculation). The time series is shuffled into a random order, for which a Lomb-Scargle estimate is generated. This process is repeated many times to generate a peak power distribution for shuffled time series (as shown in the histogram at the bottom of Supplemental Figure 2), which can be used to estimate a p-value: the probability that the Lomb-Scargle estimate for the original time series occurred by chance. Using randomization to calculate the p-value is computationally intensive because it requires many repetitions to generate a sufficient distribution; 1,000 are used in the apps, the minimal number for reasonable estimates, which in practice tend to be similar to the theoretical values. For example, for the sample synthetic time series analyzed in Supplemental Figure 2, the theoretical threshold value of 6.31 for significance level = 0.01 (reported in the app tables and shown as a horizontal orange line in the periodogram) is near the value generated by the randomization method, indicated by the vertical orange line in the lower histogram.

#### Maximum entropy spectral analysis (MESA):

The R command `spec.ar` from the *stats* R package is used to compute the MESA periodogram, using the Yule-Walker method with order  $48/\text{TimeStep}$ . The number of frequencies checked is  $2^{(5+\text{ceiling}(\log_2(N)))}$ , where  $N$  is the number of time points. The highest peak corresponding to period between 16h and 34h is chosen as the period estimate.

#### Generation of random time series:

Each sample time series in CIRCADA-S is the sum of a periodic function plus random noise plus trend. The periodic function has period  $\tau$  equal to a normally distributed random number with the selected mean and standard deviation (all samples will have the same period if the standard deviation is set to zero), user-selected amplitude  $A$ , and one of three waveforms, where  $t_{peak}$  is the time of the first peak. The sinusoidal waveform is defined as  $A \cos(2\pi(t - t_{peak})/\tau)$ ; the square waveform is defined to be the step function that equals  $2A$  when time  $t$  is within  $\tau/4$  of a multiple of  $\tau$ , and zero otherwise; and the asymmetric triangular wave is a piecewise linear function with peak value  $2A$  at time  $t_{peak}$ , decreases linearly down to equal zero at time  $t_{peak} + \tau/2$ , remains at zero until time  $t_{peak} + 5\tau/6$ , at which point it increases linearly to equal  $2A$  at time  $t_{peak} + \tau$ , then repeats periodically every  $\tau$  hours. For Gaussian noise, a vector of normally distributed random numbers with mean 0 and standard deviation as selected by the user; if noise scaled by the signal is selected, each value in this vector is multiplied by the corresponding value of the time series minus the overall minimum value, divided by the mean difference from the minimum to normalize. For pink and brown noise, the R command `noise` is used from the package `tuner`, setting `kind = "pink"` and `kind = "power"` with `alpha=2`, respectively. A linear trend is defined as  $mt$  with slope  $m$  randomly chosen between  $1/2$  and  $2$  divided by the total number of hours in the time series, equally likely to be positive or negative. A quadratic trend is defined as  $a(t - t_0)^2$  with

coefficient  $a$  randomly chosen between  $\frac{1}{2}$  and 2 divided by the square of half the total number of hours in the time series, equally likely to be positive or negative, and with vertex at  $t_0$  randomly chosen near the center of the time series (between 10 and 14 times the number of days, where 12 times the number of days gives the center of the time series).

*Packages used by the CIRCADA apps: pracma* (<https://cran.r-project.org/web/packages/pracma/>), *tidyr* (<https://cran.r-project.org/web/packages/tidyr/>), *dplyr* (<https://cran.r-project.org/web/packages/dplyr/>), *data.tree* (<https://cran.r-project.org/web/packages/data.tree/>), *DiagrammeR* (<https://rich-iannone.github.io/DiagrammeR/>), *shinyjs* (<https://deanattali.com/shinyjs/>), *shinythemes* (<https://cran.r-project.org/web/packages/shinythemes/>), *DT* (<https://rstudio.github.io/DT/>), *remotes* (<https://cran.r-project.org/web/packages/remotes/>), *tuneR* (<https://cran.r-project.org/web/packages/tuneR/>), and *shinySky* (<https://github.com/AnalytixWare/ShinySky/wiki>).

## Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

## Acknowledgements

We thank Dominica Cao, Yilin Cui, Hanxiao Lu, and Hannah Wang, for feedback and beta testing of the apps. The undergraduate research of LC, LK, and CL was supported by the Clare Boothe Luce Foundation and the Amherst College Gregory S. Call Undergraduate Research Fund. This work was supported by National Institutes of Health grants GM126545 and P01AG009975.

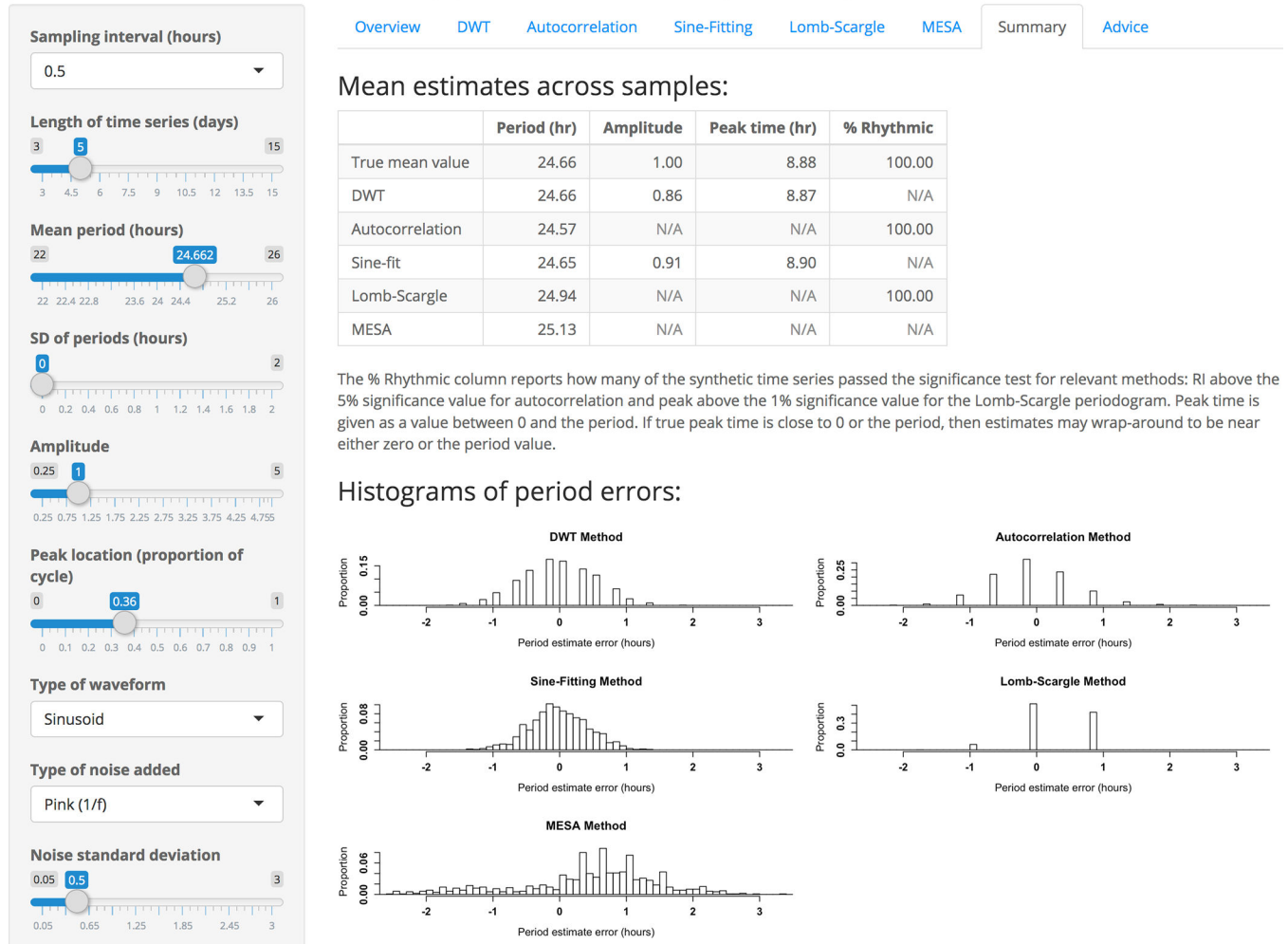
## References

- Abhilash L and Sheeba V (2019) RhythmicAlly: Your R and Shiny-based open-source ally for the analysis of biological rhythms. To appear, *Journal of Biological Rhythms*.
- Chatfield C (2004) *The Analysis of Time Series*, 6th Edition. CRC Press.
- Cohen AL, Leise TL, and Welsh DK (2012) Bayesian statistical analysis of circadian oscillations in fibroblasts. *Journal of Theoretical Biology* 314:182–191. [PubMed: 22982138]
- Cornelissen G (2014) Cosinor-based rhythmometry. *Theoretical Biology and Medical Modelling* 11:16. [PubMed: 24725531]
- Dowse HB (2009) Analyses for physiological and behavioral rhythmicity. *Methods in Enzymology* 454:141–174. [PubMed: 19216926]
- Dowse HB (2013) Maximum entropy spectral analysis for circadian rhythms: theory, history and practice. *Journal of Circadian Rhythms* 11:6. [PubMed: 23844660]
- Gierke CL and Cornelissen G (2016) Chronomics analysis toolkit (CATkit). *Biol Rhythm Res* 47:163–181.
- Geissmann Q, Garcia Rodriguez L, Beckwith EJ, and Gilestro GF (2019) Rethomics: An R framework to analyse high-throughput behavioural data. *PLoS ONE* 14(1): e0209331 10.1371/journal.pone.0209331 [PubMed: 30650089]
- Glynn EF, Chen J, and Mushegian AR (2005) Detecting periodic patterns in unevenly spaced gene expression time series using Lomb–Scargle periodograms. *Bioinformatics* 22:310–6. [PubMed: 16303799]
- Halberg F, Tong YL, and Johnson EA (1967) Circadian system phase, an aspect of temporal morphology: procedures and illustrative examples In: von Mayersbach H, editor. *The Cellular Aspects of Biorhythms*. Berlin: Springer p 20–48.
- Hughes ME, Hogenesch JB, and Kornacker K (2010) JTK\_CYCLE: an efficient nonparametric algorithm for detecting rhythmic components in genome-scale data sets. *Journal of Biological Rhythms* 25(5):372–380. [PubMed: 20876817]

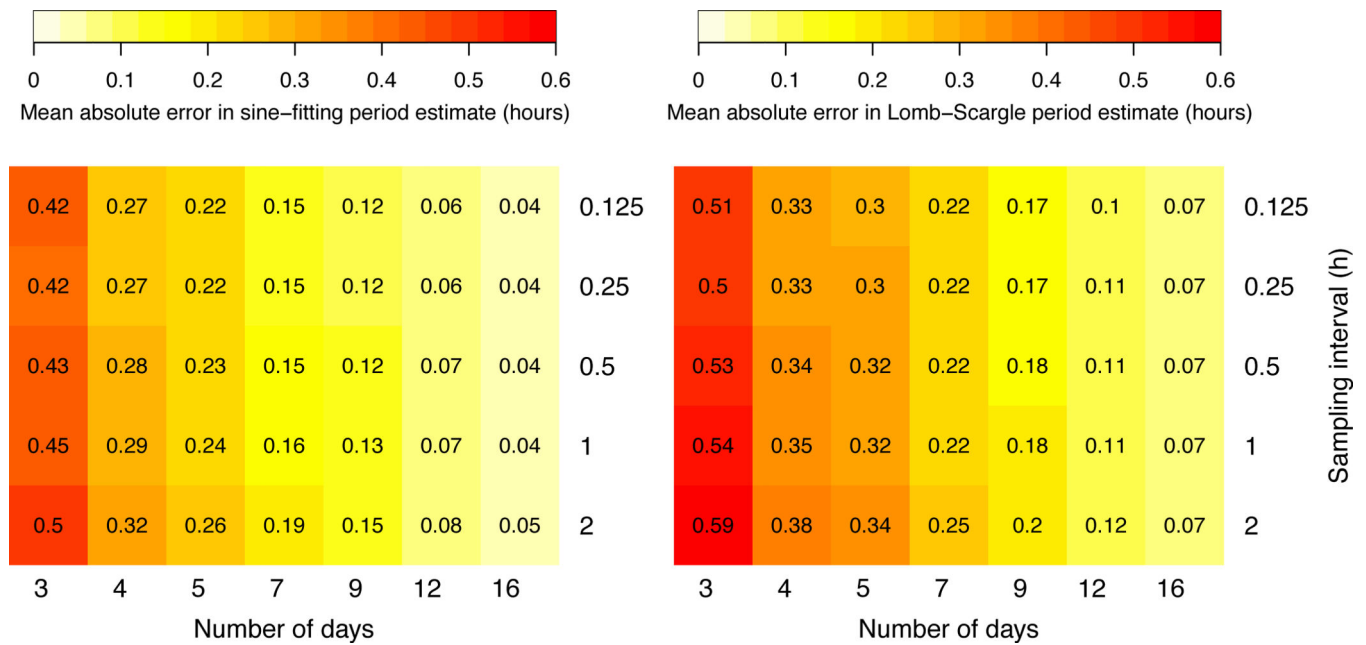
- Hurley JM, Loros JJ, and Dunlap JC (2016) Circadian oscillators: Around the transcription–translation feedback loop and on to output. *Trends in Biochemical Sciences* 41(10):834–846. [PubMed: 27498225]
- Leise TL (2015) Wavelet-based analysis of circadian behavioral rhythms In: Sehgal A, editor. *Methods in Enzymology*. San Diego, CA: Academic Press p 95–119.
- Leise TL and Harrington ME (2011) Wavelet-based time series analysis of circadian rhythms. *Journal of Biological Rhythms* 26:454–463. [PubMed: 21921299]
- Leise TL, Goldberg A, Michael J, Montoya G, Solow S, Molyneux P, Vetrivelan R, and Harrington ME (2018) Recurring circadian disruption alters circadian clock sensitivity to resetting. To appear, *European Journal of Neuroscience*.
- Levine JD, Funes P, Dowse HB, and Hall JC (2002) Signal analysis of behavioral and molecular cycles. *BMC Neuroscience* 3:1. [PubMed: 11825337]
- Lomb N (1976) Least-squares frequency analysis of unequally spaced data. *Astrophys Space Sci* 39: 447–462.
- Refinetti R, Cornelissen G, and Halberg F (2007) Procedures for numerical analysis of circadian rhythms. *Biological Rhythm Research* 38:275–325. [PubMed: 23710111]
- Ruf T (1999) The Lomb-Scargle periodogram in biological rhythm research: analysis of incomplete and unequally spaced time-series. *Biological Rhythm Research* 30(2):178–201.
- Samar VJ, Bopardikar A, Rao R, and Swartz K (1999) Wavelet analysis of neuroelectric waveforms: a conceptual tutorial. *Brain and Language* 66:7–60. [PubMed: 10080864]
- Singer JM and Hughey JJ (2019) LimoRhyde: a flexible approach for differential analysis of rhythmic transcriptome data. *Journal of Biological Rhythms* 34:5–18. [PubMed: 30472909]
- Thaben PF and Westermark PO (2014) Detecting rhythms in time series with RAIN. *Journal of Biological Rhythms* 29:391–400. [PubMed: 25326247]
- Torrence C and Compo GP (1998) A practical guide to wavelet analysis. *Bulletin of the American Meteorological Society* 79(1):61–78.
- Ulrych T and Bishop T (1975) Maximum entropy spectral analysis and autoregressive decomposition. *Rev. Geophys. Space Phys.* 13:183–300.
- VanderPlas JT (2018) Understanding the Lomb–Scargle periodogram. *The Astrophysical Journal Supplement Series* 236:16.
- Wu G, Anafi RC, Hughes ME, Kornacker K, and Hogenesch JB (2016) MetaCycle: an integrated R package to evaluate periodicity in large scale data. *Bioinformatics* 32(21):3351–3353. [PubMed: 27378304]
- Zielinski T, Moore AM, Troup E, Halliday KJ, and Millar AJ (2014) Strengths and limitations of period estimation methods for circadian data. *PloS One* 9(5): e96462. [PubMed: 24809473]



## CIRCADA-S: Circadian App for Data Analysis - Synthetic Time Series

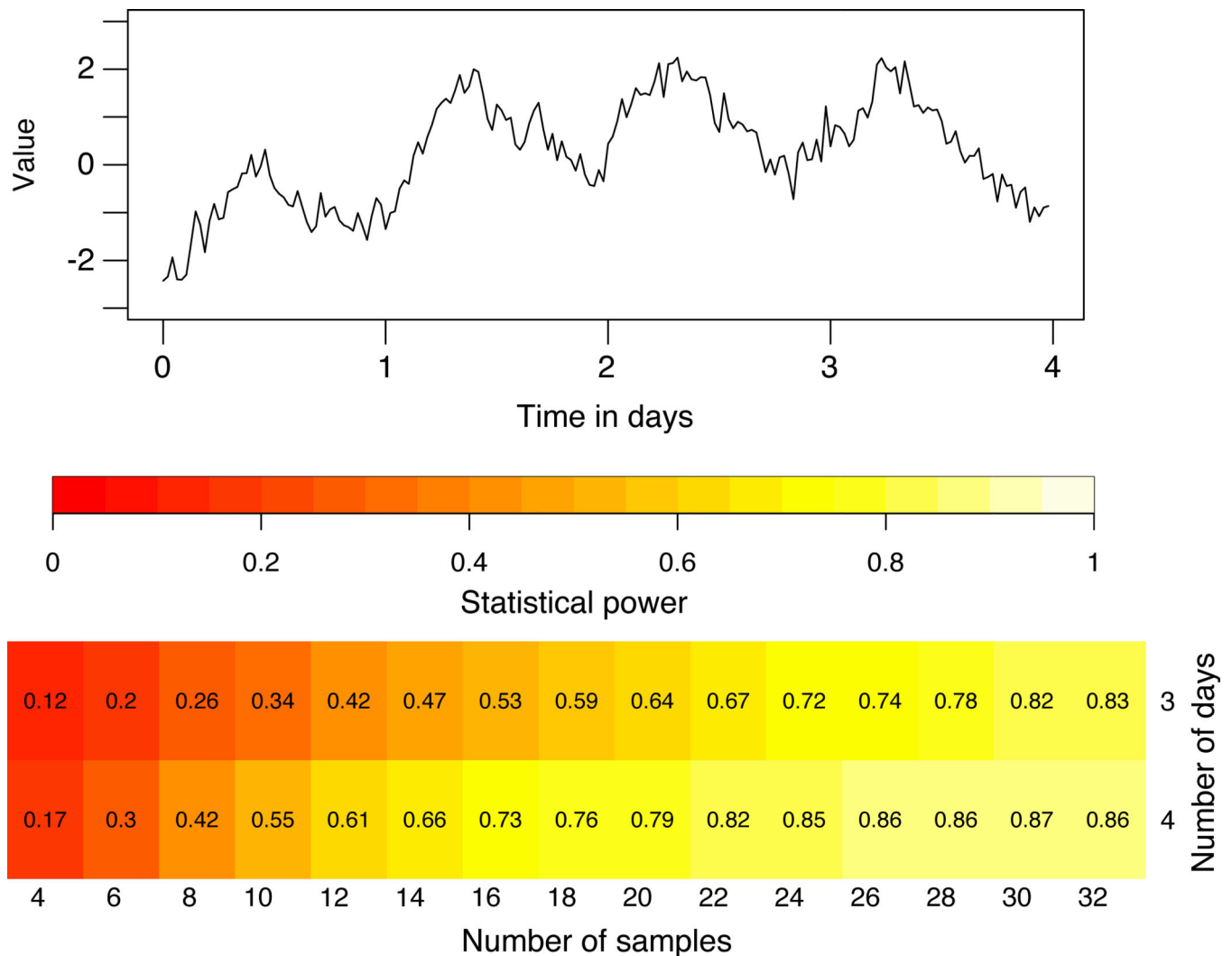


**Figure 1.** Partial screenshot of CIRCADA-S Summary tab, comparing results from the five methods for a set of 1,000 generated time series with the user-selected characteristics. The histograms of period errors demonstrate the differences between the methods. The DWT, autocorrelation, and Lomb-Scargle produce discrete estimates with possible values determined by the sampling interval and number of cycles, while sine-fitting and MESA produce finer-grained estimates.



**Figure 2.**

Errors in estimating period using sine-fitting (left) and Lomb-Scargle (right) methods for 1,000 noisy sinusoids with normally distributed periods (mean 24 h, SD 0.5 h), amplitude 1, noise standard deviation 0.5 with pink noise, and quadratic trend, for indicated number of days and sampling interval.



**Figure 3.**

Power study using generated time series. Top: example of a 4-day noisy sinusoid. Bottom: Probability of detecting a significant difference in sine-fit period between two groups for the indicated number of samples and length of time series (using t.test in R), where the true means are 23.35 and 23.85 h. Periods within each group are normally distributed with standard deviation 0.25h, amplitude is 1, noise is pink with standard deviation is 0.5, trend is quadratic, and a sampling interval is 0.5h. Around 30 samples would be needed to obtain a statistical power of 0.80 using a 3-day duration, while increasing the duration to 4 days substantially decreases the required sample size to just over 20.

**Table 1.**

Summary of methods available in CIRCADA apps.

Method	Purpose	References	Notes
Discrete wavelet transform (DWT)	Removal of noise and/or trend; estimates of peak times and heights	(Leise and Harrington, 2011) (Leise, 2015) (Samar et al., 1999) (Torrence and Compo, 1998)	May need to discard first and last day due to edge effects; sampling rate determines period bands; best if time series has at least 5 cycles
Autocorrelation	Evidence of rhythmicity via RI; period estimate	(Chatfield, 2006, p22–28) (Dowse, 2009) (Levine et al., 2002)	Works well for any waveform; should detrend first; good initial test to apply in examining a possibly rhythmic time series; best if time series has at least 4 cycles
Sine-fitting	Estimates of period, phase, and amplitude	(Cornelissen, 2014) (Cohen et al., 2012) (Refinetti et al., 2007) (Halberg et al., 1967)	Can be used on non-uniformly sampled data; works best for sinusoidal waveforms; typically good to detrend before applying
Lomb-Scargle periodogram	Estimate of period; statistical test for significance	(Glynn et al., 2005) (Ruf, 1999) (VanderPlas, 2018) (Lomb, 1976)	Can be used on non-uniformly sampled data; can use randomization to generate p-values if assumptions aren't met for theoretical formula
MESA	Estimate of period	(Dowse, 2009) (Dowse, 2013) (Ulrych and Bishop, 1975)	No significance test; higher resolution estimates than Fourier periodogram

**Table 2.**

Summary of what each method is capable of estimating, as implemented in the CIRCADA apps.

	<b>DWT</b>	<b>Autocor.</b>	<b>Sine-fit</b>	<b>Lomb-S.</b>	<b>MESA</b>
Period	✓	✓	✓	✓	✓
Phase	✓		✓		
Amplitude	✓		✓		
p-value for rhythmicity		✓		✓	
Strength of rhythmicity		✓			

Author Manuscript

Author Manuscript

Author Manuscript

Author Manuscript

**Table 3.**

User choices for generating synthetic data in CIRCADA-S. To run with all replications having identical period (and other characteristics), set SD of periods to zero; a non-zero value results in a normally distributed set of periods with indicated mean and SD. Noise “scaled by the signal” is generated by taking Gaussian noise and multiplying pointwise by the signal, to imitate experimental data in which noise increases in proportion with the signal value.

Parameter	Range of values
Sampling interval	0.125, 0.25, 0.5, 1, or 2 h
Mean period	20–28 hours
Standard deviation (SD) of periods	0–2 hours
Peak location	0–1 (any proportion of a cycle)
Amplitude	0–3 (arbitrary units)
Waveform	Sinusoid, square wave, or asymmetric triangular wave
Noise standard deviation	0–3 (arbitrary units)
Noise type	Gaussian, pink, brown, or scaled by the signal
Trend	None, linear, or quadratic
# of samples	1–1000