Manual Rhythm Analysis App

Dr. Lara S. Burchardt

3/14/2022

Contents

1 Introduction

This is the manual for the shiny application "Rhythm Analysis". The following section will guide you through the first steps and will show you, how to work with the app.

The code for the app lies on my github repository.

Who am I? you can find out more about me here

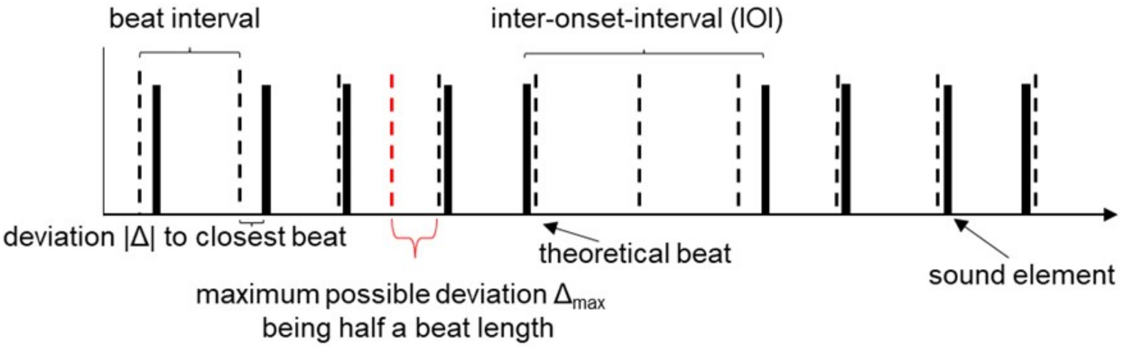
1.1 Why use this app?

1.2 A little background

The temporal structure of animal communication is a potentially crucial parameter, that historically was often overlooked, especially in mammals and birds. With the development of new methods in recent years, research on rhythm analysis in animal communication is an ever-growing field, with yet many knowledge gaps: we want to identify the adaptive function of rhythms and add this piece of the puzzle to ongoing discussions about the evolution of language and music. Apart from that, the temporal structure of vocalizations can inform about species identity, communication context, arousal state, health conditions, and many more, which has wide implications for a lot of research fields ranging from behavioral ecology and cognitive science to conversation biology, and animal welfare. (Anichini et al., 2020; Honing & Ploeger, 2012; Honing et al., 2015; Manser, 2001; Ravignani et al., 2019).

Different parameters can be measured in the context of rhythm analysis, many of which I studied and developed during my doctoral thesis (Burchardt, Briefer, et al., 2021; Burchardt & Knörnschild, 2020). For example, we can calculate simple distributional parameters, such as mean and coefficient of variation of Inter-Onset-Intervals (IOI), the duration between the start of a sound element in a sequence and the start of the next element in that sequence (Figure 1). A possible next step is to describe a single sequence with one particular rhythm or beat in hertz (Hz). A sequence with a rhythm of 5 Hz would have 5 equally distributed sound elements per second, so roughly one sound element every 200 ms. If it is a perfect fit, it is exactly every 200 ms, but in biological systems such 'perfection' seldom happens. Beat precision can then be used to describe how good the fit of the actual elements to the calculated theoretical beat is. I will analyze exact beat frequencies to describe a sequence with the so-called IOI approach (Burchardt & Knörnschild, 2020) and Fourier analysis (Burchardt & Knörnschild, 2020; Burchardt, Picciulin, et al., 2021; Ravignani & Norton, 2017; Saar & Mitra, 2008) and I will use the universal goodness-of-fit value (ugof), which I developed during my doctoral thesis, to determine the beat precision (Figure 1). The striking advantages of the ugof parameter include that it can be calculated independent of the method used to calculate the exact beats; it is independent of the number of elements, the rhythm it is describing, or any other parameter. Other goodness-of-fit values were correlated to i.e., the number of elements or the rhythm they were describing, making them unfit for comparative work or the general description of beat precision. Another important aspect of the ugof parameter is that it can be calculated per sound element and can therefore be investigated on very different levels ranging from single sound elements to long sound sequences. It can be calculated for a single individual, or several individuals of the same species in comparison with another species, making it potentially useful for very different kinds of questions.

This app calculates the rhythm of a sequence of time points using exactly these methods and parameters.


beat interval
inter-onset-interval (IOI)
deviation |Δ| to closest beat
theoretical beat
sound element
maximum possible deviation Δmax
being half a beat length

$$ugof = \frac{|\Delta|}{\Delta_{max}}$$

Figure 1: Theoretical Explanation of the universal goodness of fit value. If applicable, please cite Burchardt et al, 2021

1.2.1 Literature

The methods were described and used by me for example in the following peer-reviewed publications:

Methods:

Paper 1: "Comparison of methods for rhythm analysis of complex animals' acoustic signals" by Burchardt & Knörnschild, 2020

Paper 2: "Novel ideas to further expand the applicability of rhythm analysis" by Burchardt, Briefer and Knörnschild, 2021

The code in action:

Paper 3: "Assessing acoustic competition between sibling frog species using rhythm analysis" by Filer, Burchardt, van Rensburg, 2021

Paper 4: "A primer on rhythm quantification for fish sounds: a Mediterranean case study" by Burchardt, Picciulin, Parmentier and Bolgan, 2021

1.3 Preparations

1.3.1 Packages

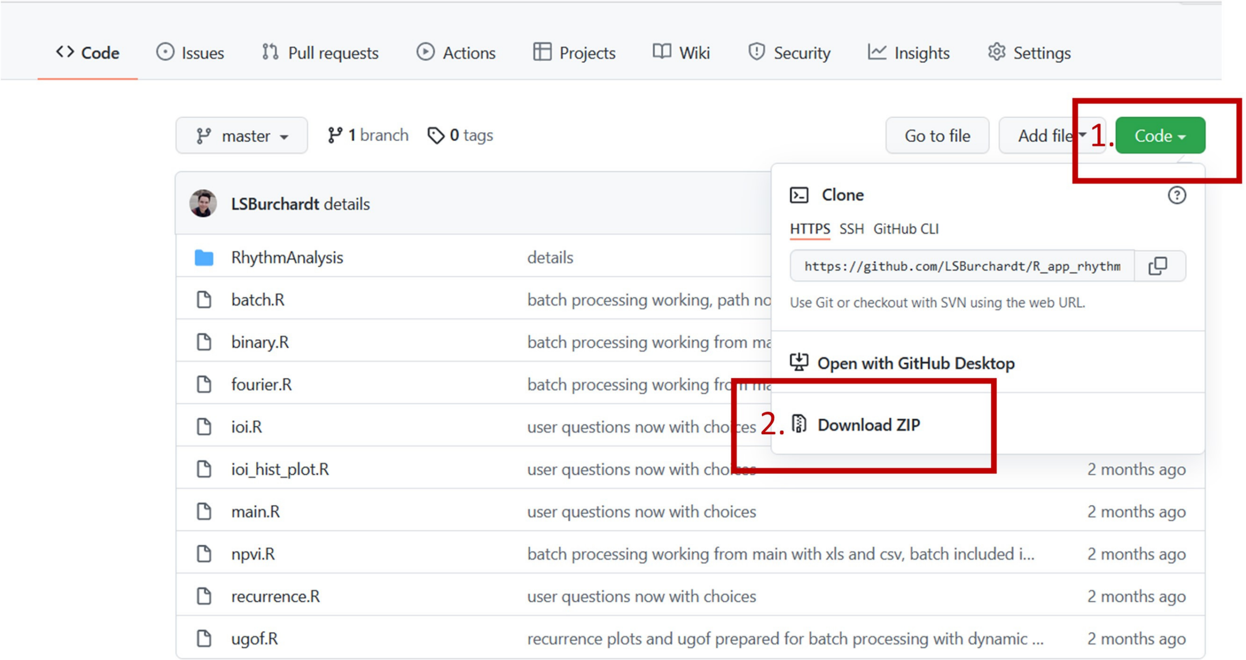
For the app to work a bunch of packages need to be installed. The app will automatically check, whether all necessary packages are installed on your local machine, and if not, will install them This is an overview of the needed packages, just for your information.

```
install.load
shiny
shiny.Files
shinybusy
shinyjs
shinyWidgets
tidyverse
readxl
openxlsx
svDialogs
pracma
SynchWave
vegan
corrplot
plotly
DT
```

1.3.2 Necessary Code on your machine

In the RStudio app, set up new project (`file -> New project`). If you have never used R projects before, see here for a short explanation of the use and advantageous of R Projects.

You now need to download the code from github into this project folder. This will ensure, that RStudio can access all necessary files and data without any further steps.



`https://github.com/LSBurchardt/R_app_rhythm`

Use Git or checkout with SVN using the web URL.

LSBurchardt details

RhythmAnalysis	details	
batch.R	batch processing working, path no	
binary.R	batch processing working from ma	
fourier.R	batch processing working fro	
ioi.R	user questions now with cho ces	
ioi_hist_plot.R	user questions now with cho	2 months ago
main.R	user questions now with choices	2 months ago
npvi.R	batch processing working from main with xls and csv, batch included i...	2 months ago
recurrence.R	user questions now with choices	2 months ago
ugof.R	recurrence plots and ugof prepared for batch processing with dynamic ...	2 months ago

Figure 2: Screenshot of github page to guide code download.

On the project webpage, you click the green button “Code” and then choose “Download ZIP” (Figure 1) Actually needed are the following items in the folder:

```
global.R
ui.R
server.R
manual.pdf
```

and the folders `www` and `images_manual` with all its content.

Everything, that you save within the app (i.e., data) will also be saved into the project folder.

2 How to use the app?

2.1 How should the input data look like?

The input data needs to include the relevant event time points you want to calculate rhythms for. This could be starting points of sound elements, end points of sound elements, or timepoint with peak energy of sound elements, depending on your data and approach. These timepoints need to be in the first column of a datatable input, that could be in one of the three formats:

1. csv
2. xls
3. xlsx

You select the input format later in the app through a dropdown menu. The data table can include respective “endpoints” of sound elements, if applicable or element types in a second and third column, but that is not necessary. A possible example of an input data table is shown below.

Table 1: Exemplary input data table with timepoints of interest in the first column (here start points), the respective end points and the element type.

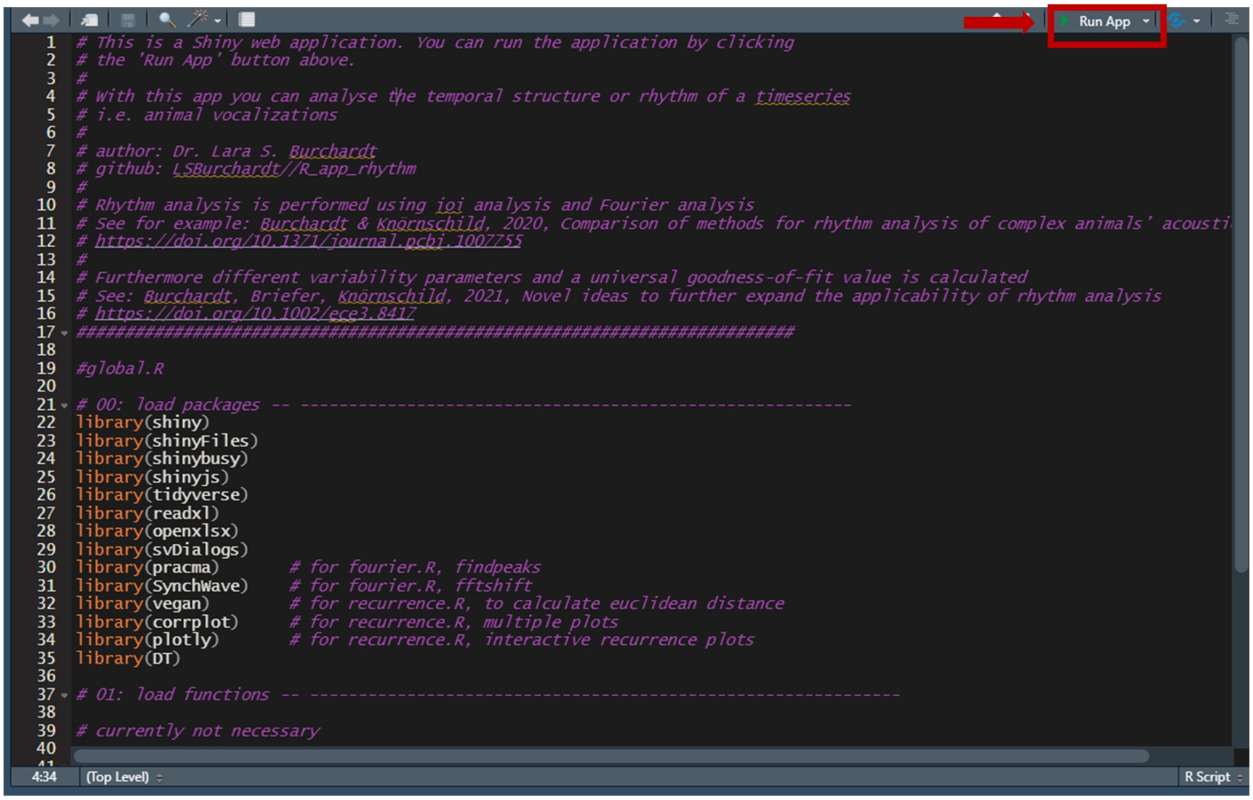
start	end	element type
0.1	0.12	a
0.2	0.23	a
0.3	0.31	a
0.4	0.42	a
0.5	0.51	a
0.6	0.62	a
0.7	0.71	a
0.8	0.82	a
0.9	0.91	a
1	1.01	a

2.2 Starting the app

Before you start the app, your environment should be empty, no datasets should be loaded or values be stored. To make sure it is, run the following code `rm(list = ls)` in the console. WARNING: Only do this, when the respective project is opened! Otherwise you might loose data from other projects, you do not want to remove.

Open one of the following scripts: `global.R`, `ui.R` or `server.R`. All three scripts together form the app. You can start the app from any of the three files.

To avoid confusion, it might be easiest to open the `global.R` file, as it is the shortest. Once you open it, on the top right corner you will see the option `Run App` with a green triangle in front of it. If you click on that option, the app will start.

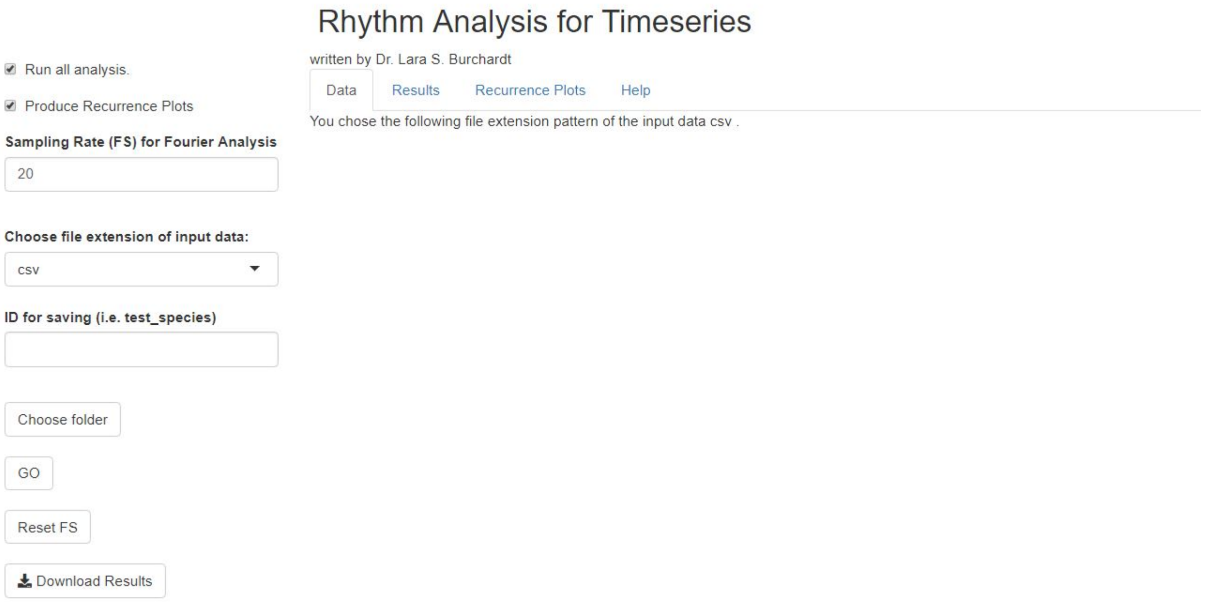


```
# This is a Shiny web application. You can run the application by clicking
# the 'Run App' button above.
#
# With this app you can analyse the temporal structure or rhythm of a timeseries
# i.e. animal vocalizations
#
# author: Dr. Lara S. Burchardt
# github: LSBurchardt//R_app_rhythm
#
# Rhythm analysis is performed using ioi analysis and Fourier analysis
# See for example: Burchardt & Knörnschild, 2020, Comparison of methods for rhythm analysis of complex animals' acousti
# https://doi.org/10.1371/journal.pcbi.1007755
#
# Furthermore different variability parameters and a universal goodness-of-fit value is calculated
# See: Burchardt, Briefer, Knörnschild, 2021, Novel ideas to further expand the applicability of rhythm analysis
# https://doi.org/10.1002/ece3.8417
##########################################################################

#global.R

# 00: load packages -- ---------------------------------------------------------
library(shiny)
library(shinyFiles)
library(shinybusy)
library(shinyjs)
library(tidyverse)
library(readxl)
library(openxlsx)
library(svDialogs)
library(pracma)       # for fourier.R, findpeaks
library(SynchWave)    # for fourier.R, fftshift
library(vegan)        # for recurrence.R, to calculate euclidean distance
library(corrplot)     # for recurrence.R, multiple plots
library(plotly)       # for recurrence.R, interactive recurrence plots
library(DT)

# 01: load functions -- ------------------------------------------------------------

# currently not necessary

```

Figure 3: Where to find the "Run App button" when, i.e. the `global.R` script is opened.

2.3 Input parameters, Options

When you start the app, a new window will open, that looks like this:



Rhythm Analysis for Timeseries

written by Dr. Lara S. Burchardt

You chose the following file extension pattern of the input data csv .

☑ Run all analysis.

☑ Produce Recurrence Plots

Sampling Rate (FS) for Fourier Analysis: 20

Choose file extension of input data: csv

ID for saving (i.e. test_species):

Figure 1: Alt text

You can change several inputs. 1. First of all, you can decide, whether Recurrence Plots should be produced, this is a slightly time consuming task, which is why the option is given not to render them. 2. The second input is the sampling rate (FS) you want to choose for the Fourier analysis calculations. The default is 20. Note, that frequencies can be decomposed up to half of the sampling rate. So for the default FS of 20 you would analyse the rhythm in a window between 0 and 10 Hz with the Fourier analysis. 3. As mentioned before, you need to specify the format of your input data. Only one input format can be analysed at a time. 4. You can select a savename for your data, that will be included in the filename, of you download the results. A savename is not necessary for the app to work properly.

2.4 Choose Data to analyse

After choosing all necessary input arguments, you can click the `Choose folder` button, which will open an explorer window, where you select the folder containing your data in the chosen file format such as `csv`. If there is other files in the same folder, with the same file extension, this will lead to errors, so make sure, to have a designated folder for your data.

After you chose the folder, the data to be analysed will appear in list form in the `Data` Tab (Figure 4).

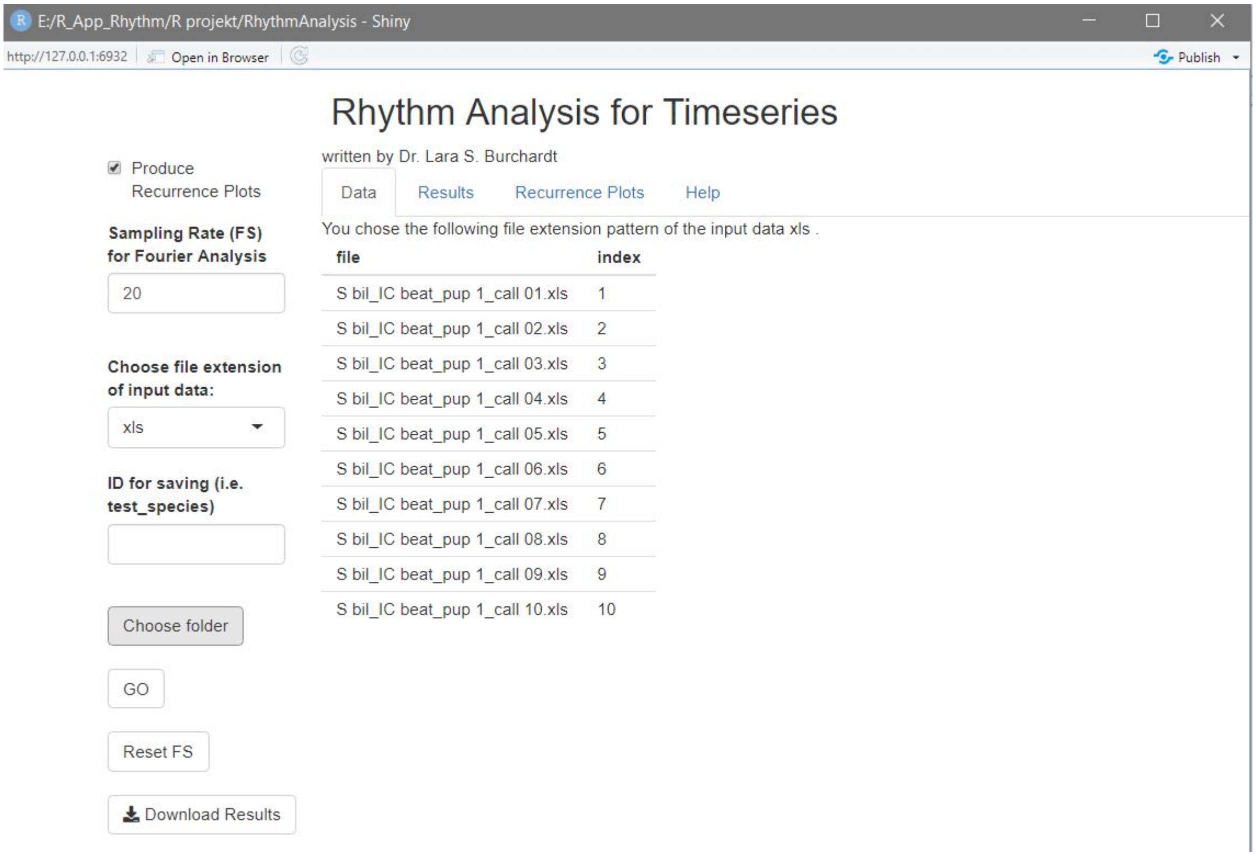
click the “GO” Button to run the analyses on the chosen data. If you want to calculate the results with another sampling rate, change the sampling rate and click “Reset FS” to run the calculations again with the new parameter.

If you want to analyse a different data set with a different file extension, you need to restart the app. The app will crash, when you change the file extension after having run calculations already.

2.5 Analysis and Output

All analyses will run automatically and results will appear in the `Results` Tab. Results will be presented in a table. Visualizations will also be generated for the relevant results parameters.

It will look something like this:



Rhythm Analysis for Timeseries

written by Dr. Lara S. Burchardt

☑ Produce Recurrence Plots

Sampling Rate (FS) for Fourier Analysis

20

Choose file extension of input data:

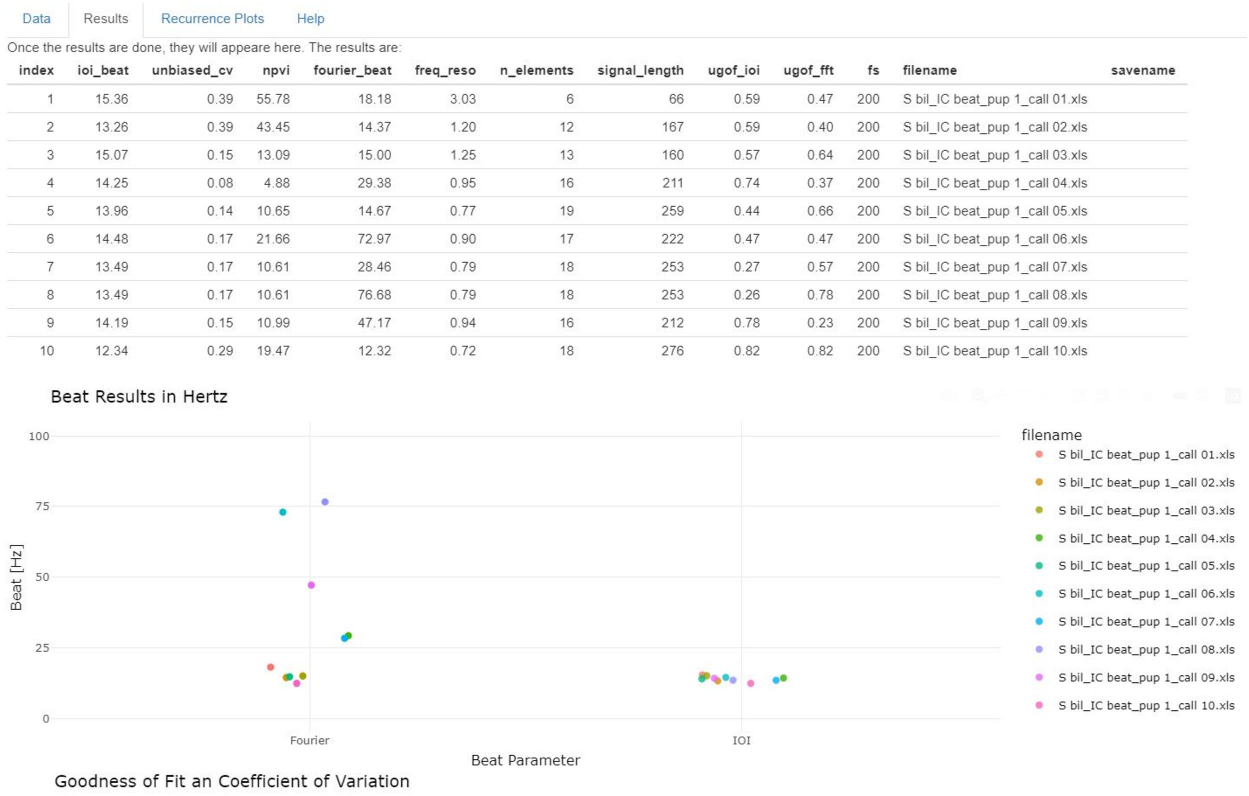
xls

ID for saving (i.e. test_species)

You chose the following file extension pattern of the input data xls .

file	index
S bil_IC beat_pup 1_call 01.xls	1
S bil_IC beat_pup 1_call 02.xls	2
S bil_IC beat_pup 1_call 03.xls	3
S bil_IC beat_pup 1_call 04.xls	4
S bil_IC beat_pup 1_call 05.xls	5
S bil_IC beat_pup 1_call 06.xls	6
S bil_IC beat_pup 1_call 07.xls	7
S bil_IC beat_pup 1_call 08.xls	8
S bil_IC beat_pup 1_call 09.xls	9
S bil_IC beat_pup 1_call 10.xls	10

Figure 2: Alt text



Once the results are done, they will appeare here. The results are:

index	ioi_beat	unbiased_cv	npvi	fourier_beat	freq_reso	n_elements	signal_length	ugof_ioi	ugof_fft	fs	filename	savename
1	15.36	0.39	55.78	18.18	3.03	6	66	0.59	0.47	200	S bil_IC beat_pup 1_call 01.xls	
2	13.26	0.39	43.45	14.37	1.20	12	167	0.59	0.40	200	S bil_IC beat_pup 1_call 02.xls	
3	15.07	0.15	13.09	15.00	1.25	13	160	0.57	0.64	200	S bil_IC beat_pup 1_call 03.xls	
4	14.25	0.08	4.88	29.38	0.95	16	211	0.74	0.37	200	S bil_IC beat_pup 1_call 04.xls	
5	13.96	0.14	10.65	14.67	0.77	19	259	0.44	0.66	200	S bil_IC beat_pup 1_call 05.xls	
6	14.48	0.17	21.66	72.97	0.90	17	222	0.47	0.47	200	S bil_IC beat_pup 1_call 06.xls	
7	13.49	0.17	10.61	28.46	0.79	18	253	0.27	0.57	200	S bil_IC beat_pup 1_call 07.xls	
8	13.49	0.17	10.61	76.68	0.79	18	253	0.26	0.78	200	S bil_IC beat_pup 1_call 08.xls	
9	14.19	0.15	10.99	47.17	0.94	16	212	0.78	0.23	200	S bil_IC beat_pup 1_call 09.xls	
10	12.34	0.29	19.47	12.32	0.72	18	276	0.82	0.82	200	S bil_IC beat_pup 1_call 10.xls	

Beat Results in Hertz
100
75
50
25
0
Beat [Hz]
Fourier
IOI
Beat Parameter
filename
● S bil_IC beat_pup 1_call 01.xls
● S bil_IC beat_pup 1_call 02.xls
● S bil_IC beat_pup 1_call 03.xls
● S bil_IC beat_pup 1_call 04.xls
● S bil_IC beat_pup 1_call 05.xls
● S bil_IC beat_pup 1_call 06.xls
● S bil_IC beat_pup 1_call 07.xls
● S bil_IC beat_pup 1_call 08.xls
● S bil_IC beat_pup 1_call 09.xls
● S bil_IC beat_pup 1_call 10.xls
Goodness of Fit an Coefficient of Variation

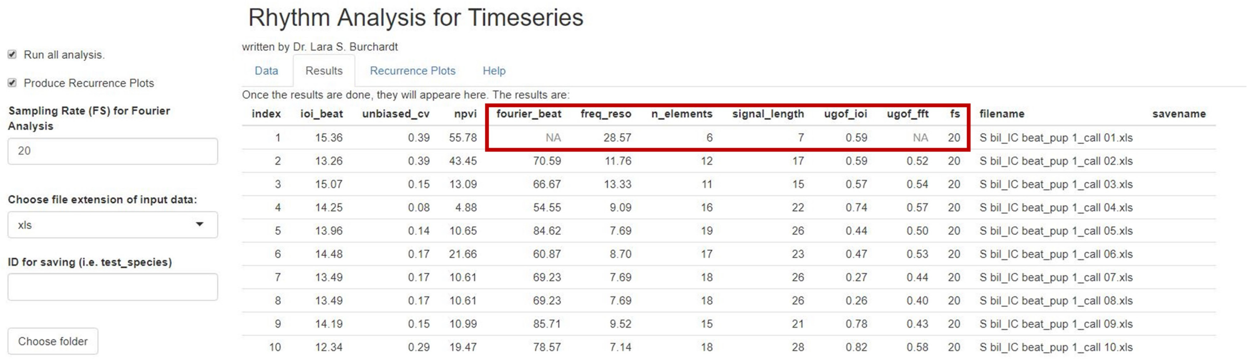
Results and Results Table

The following parameters and results are calculated:

1. the IOI beat That is the beat frequency in Hertz, that describes the element sequence best, when using the average IOI duration as the basis for the rhythm calculation. This method is suitable for many datasets. It is suitable for datasets with a low variability in IOI duration and a general consistency, i.e. because the sequence is not that long, consists of only the same element type and the temporal structure is very consistent.

2. the unbiased Coefficient of Variation This parameter indicates the variability of IOI durations independent of the sample size and the actual mean it is describing. The CV sets mean and standard deviation into relation and can be interpreted as a percentage. The higher it is, the higher is the variability.
3. the normalized Pairwise Variability Index (nPVI) Another variability parameter, originating in linguistics. It indicates how well you can predict one IOI by the preceding IOI. A value of 0 would indicate, that every IOI in a sequence is exactly equal.
4. the Fourier beat That is the beat frequency in Hertz, that describes the element sequence best, when using Fourier analysis to calculate rhythm. For that, the sequence is first transformed into a binary sequence with th chosen sampling rate. This binary sequence, where the points of interest (i.e., start point of a sound element) are coded as 1 and everything is 0, is decomposed into its sinusoidal components with a fast Fourier transformation. Note, that this calculation is strongly influenced by the frequency resolution of the Fourier analysis. The temporal resolution, i.e., the sampling rate divided by the sample length gives the frequency resolution. As a rule of thumb: the longer the signal, the smaller is the problem of a coarse frequency resolution. In case the frequency resolution is higher, than the actual frequency window, that can be depicted (i.e., sampling rate of 20 Hz, resulting in a frequency window up to 10 Hz, but a frequency resolution of 11 HZ) this value will be given as NA (Figure 7)

5. the frequency resolution That is the frequency resolution of the Fourier Beat. This is especially important, of you want to interpret the suitability of the Fourier approach or compare the IOI beat and the Fourier beat. The IOI beat is a continuous value, the Fourier beat result can only take multiplies of the frequency resolution. Fourier beats of a sequence with a low (i.e. large number) frequency resolution are to be treated



Rhythm Analysis for Timeseries

written by Dr. Lara S. Burchardt

☑ Run all analysis.

☑ Produce Recurrence Plots

Sampling Rate (FS) for Fourier Analysis: 20

Choose file extension of input data: xls

ID for saving (i.e. test_species)

Once the results are done, they will appeare here. The results are:

index	ioi_beat	unbiased_cv	npvi	fourier_beat	freq_reso	n_elements	signal_length	ugof_ioi	ugof_fft	fs	filename	savename
1	15.36	0.39	55.78	NA	28.57	6	7	0.59	NA	20	S bil_IC beat_pup 1_call 01.xls	
2	13.26	0.39	43.45	70.59	11.76	12	17	0.59	0.52	20	S bil_IC beat_pup 1_call 02.xls	
3	15.07	0.15	13.09	66.67	13.33	11	15	0.57	0.54	20	S bil_IC beat_pup 1_call 03.xls	
4	14.25	0.08	4.88	54.55	9.09	16	22	0.74	0.57	20	S bil_IC beat_pup 1_call 04.xls	
5	13.96	0.14	10.65	84.62	7.69	19	26	0.44	0.50	20	S bil_IC beat_pup 1_call 05.xls	
6	14.48	0.17	21.66	60.87	8.70	17	23	0.47	0.53	20	S bil_IC beat_pup 1_call 06.xls	
7	13.49	0.17	10.61	69.23	7.69	18	26	0.27	0.44	20	S bil_IC beat_pup 1_call 07.xls	
8	13.49	0.17	10.61	69.23	7.69	18	26	0.26	0.40	20	S bil_IC beat_pup 1_call 08.xls	
9	14.19	0.15	10.99	85.71	9.52	15	21	0.78	0.43	20	S bil_IC beat_pup 1_call 09.xls	
10	12.34	0.29	19.47	78.57	7.14	18	28	0.82	0.58	20	S bil_IC beat_pup 1_call 10.xls	

Figure 3: Alt text

and interpreted carefully. Ideally, the frequency resolution should be <0.5 Hz.

6. the number of elements The number of elements in each sequence is also depicted in the results table.
7. the signal length That is the signal length of the binary sequence necessary to calculate the Fourier analysis. It depends on the sampling rate and the total signal duration and influences the frequency resolution.
8. the universal goodness of fit of the IOI beat That is the value, indicating, how well the IOI beat describes the sequence. It can lie between 0 and 1 and can be interpreted as a percentage. The lower it is, the better the fit between theoretical beat and actual sound elements.
9. the universal goodness of fit of the Fourier beat That is the same parameter, just calculated for the Fourier beat

For detailed explanations of the parameters and methods please see: 1. Burchardt & Knörnschild, 2020 2. Burchardt et al, 2021

Furthermore shown in the results table are:

10. the sampling rate The chosen sampling rate is given in the results table.
11. the file name The file name of the analysed sequence
12. a possible save name
    If you chose a "savename" in the interface, that will be depicted here as well, for the easier discrimination in possible subsequent analyses.

The very first column of the results table just gives an index number, that can be used for subsequent analyses and can be used in this app, to re-analyze certain parts of a certain sequence.

2.5.1 Results and Visualizations

3 The different Tabs in detail

3.1 Data

3.2 Results

3.3 Recurrence Plots

3.4 Re-run analysis on Section

3.5 Beat precision details

3.6 Help