Identification and characterisation of variable oscillatory behaviour using CosinorPy

Miha Moškon*

Faculty of Computer and Information Science, University of Ljubljana, Ljubljana, Slovenia

Abstract

Background and objective: Analysis of rhythmic data has become an important aspect in biological and medical as well as in other fields of science. Several nonparametric methods aimed at such analyses have been reported in recent years. However, parametric methods based on trigonometric regression still reflect several advantages. Different software packages for parametric analysis of rhythmic data have been made available recently. These are mostly based on a single-component cosinor model, which is not able to describe asymmetric oscillations that might reflect multiple peaks per period. Moreover, a basic cosinor model is unable to describe oscillations that are changed (namely forced or damped) with time. Here, we present some important extensions of the recently developed CosinorPy Python package to address these gaps.

Methods: The extended version of CosinorPy provides functionalities to perform a detailed individual or comparative analysis of rhythmic behaviour reflecting (1) multiple asymmetric peaks per period, (2) forced, damped or

*Corresponding author

Email address: `miha.moskon@fri.uni-lj.si` (Miha Moškon)

sustained oscillations, and (3) accumulation or reduction of MESOR with time. In all these cases the package is able to assess the (differential) rhythmicity parameters, evaluate their significance and confidence intervals, and provide a set of publication-ready figures.

Results: We demonstrate the package on some typical scenarios incorporating different types of oscillatory dynamics, such as asymmetric, damped, and forced oscillations. We show that the proposed implementation of a generalised cosinor model is able to reduce the error of estimated rhythmicity parameters in cases that tend to be problematic for alternative models. The implementation of the presented package is available together with the scripts to reproduce the reported results at https://github.com/mmoskon/CosinorPy.

Conclusion: According to our knowledge, CosinorPy currently presents the only implementation that is able to cover the above scenarios using a parametric model with vast applications in biology and medicine as well as in other scientific domains.

Keywords: cosinor, rhythmicity analysis, multi-component cosinor, generalised cosinor, comparative rhythmicity.

---

1. Introduction

Identification and analysis of rhythmic data, for example, circadian data, has become important in different aspects of science, especially in the context of biology and medicine. Several non-parametric approaches aimed at such

analyses have been reported in recent years (for examples, see [1, 2, 3, 4, 5]). However, parametric approaches mostly based on trigonometric regression still reflect several advantages over nonparametric approaches in many cases. Parametric methods can be applied even in cases when we are dealing with unevenly spaced and/or unbalanced data [6, 7, 8, 9, 10]. Rhythmicity parameters, such as oscillation amplitudes and acrophases, can be derived directly from the coefficients of a regression model, which is not the case of many nonparametric methods [2]. Parametric methods can be used to assess the significance of each rhythmicity parameter being different from zero [11] as well as to compare rhythmicity parameters among different measurements and to assess their differential rhythmicity using easy-to-interpret p-values [9, 12, 13, 14]. Parametric methods can be adapted to different situations and can be used to assess the population mean response [9, 11, 12, 15] as well in a combination with different count models for the analysis of count data (for example, see [16]). Nonparametric methods are mostly aimed towards the analysis of omics, e.g., transcriptomics data. On the other hand, parametric approaches have a much wider applicability. They can be applied to different domains even outside biology and medicine (for example, see [17]).

Recent parametric approaches for identification and analysis of rhythmic data are mostly based on a cosinor model [11], and include ECHO (extended circadian harmonic oscillator) [18], COMPARERHYTHMS [19], CircaCompare [14] and CosinorPy [9]. ECHO is based on a generalised cosinor model, which is able to describe sustained (as in the case of a basic cosinor model),

damped or forced oscillations. However, a threshold used for the classification of a measurement in any of these three groups are arbitrarily defined and might depend on the noise in the data [18]. Moreover, ECHO is limited to a single-component cosinor model, which means it can only describe the oscillation reflecting a single and a symmetric peak per period. COMPARERHYTHMS provides a differential rhythmicity analysis pipeline based on the basic single-component cosinor model. CircaCompare presents a method also focused to the assessment of differential rhythmicity and is based on a non-linear regression in a combination with the cosinor model [14]. It is again limited to a single-component model and does not account for attenuation or dampening of oscillation amplitudes over time.

Recently, CosinorPy, a Python package for the identification and characterisation of rhythmic behaviour has been reported [9]. This package has already been used in different studies ranging from a the analysis of 'circadian reprogramming' in cell lines using qPCR data [20] to the prediction of sleep quality metrics using wearable devices data [21]. CosinorPy implements single- as well as multicomponent cosinor models, which can be used in a combination with the population as well as count models. Moreover, the package can be used to assess the differential rhythmicity among pairs of measurements. However, if a user requires evaluation of more accurate parameter statistics, such as the significance of oscillatory amplitudes and the significance of change between acrophases in pairs of measurements, a single-component cosinor had to be used in the basic implementation. In

this model, the rhythmicity parameters and their corresponding p-values can be evaluated analytically using the delta method, as introduced in [22] and adapted in [9]. However, when using a multicomponent cosinor model, the rhythmicity parameters cannot be calculated directly from the regression coefficients, so alternative methods must be used. Their evaluation can be performed numerically from the fitted curves [9]. Here, we describe extensions to the multicomponent cosinor model that allow us to evaluate (1) the p-values of rhythmicity parameters in a multicomponent cosinor model, and (2) the p-values describing the significance of changes in the rhythmicity parameters in pairs of measurements using multicomponent cosinor models. The methodology behind these extensions is based on bootstrapping of confidence intervals and p-values of selected rhythmicity parameters. This increases the computation time required to obtain the results but provides much richer statistics.

Furthermore, we extend the cosinor model to a generalised cosinor model that can account for the accumulation or reduction of the midline statistics of rhythm (MESOR) as well as to describe the damping or amplification of oscillation amplitudes over time. This is especially vital for longer time series (relative to the period of oscillations) during which amplitude and MESOR are likely to change due to different factors [18, 23]. We extend both a single-component and a multicomponent cosinor model to its generalised version. The rhythmicity parameters, their corresponding confidence intervals, and p-values can be estimated directly from the regression coefficients

in a single-component model. In a multicomponent model, bootstrapping is used in a similar way as in a basic multicomponent cosinor model. Contrary to ECHO, our implementation can yield a p-value describing the significance of damping or amplification of an amplitude over time and can be used even when data exhibits multiple asymmetric peaks within a period of oscillation. Moreover, the proposed implementation can be used to assess the comparative rhythmicity of pairs of measurements to evaluate changes in rhythmicity parameters, such as the significance of acrophase shift, amplitude change, change in the amplification coefficient, or change in the linear component.

The extended version of the CosinorPy package thus provides the functionalities to perform an individual or comparative analysis of rhythmic behaviour reflecting (1) multiple asymmetric peaks per period, (2) forced, damped or sustained oscillations, and (3) accumulation or reduction of MESOR with time. According to our knowledge, it presents the only implementation that can cover all these scenarios using a parametric model with vast applications in biology and medicine as well as in other scientific domains. The functionalities reported here are available in the extended CosinorPy package (version 2.1), which can be installed by `pip install CosinorPy==2.1`. All the main functionalities of the package can be accessed through easy-to-use wrapper functions as demonstrated in a series of interactive Python notebooks (IPYNB) available at https://github.com/mmoskon/CosinorPy.

2. Methods

2.1. From a single- to a multi-component cosinor model

A multicomponent cosinor model can be described by the following equation:

$$y(t) = A + \sum_{i=1}^{N} \left(\beta_{i,1} \cdot \sin\left(\frac{t}{\tau/i} \cdot 2\pi\right) + \beta_{i,2} \cdot \cos\left(\frac{t}{\tau/i} \cdot 2\pi\right)\right) + E(t), \quad (1)$$

where t corresponds to the observed time points in the time series, $E(t)$ is the error term, τ is the assumed period of the oscillations, N is the number of components in the model, and A, $\beta_{1,1}$, $\beta_{1,2}$,..., $\beta_{N,1}$, $\beta_{N,2}$ are the regression coefficients. In these, A describes the MESOR (midline statistics of rhythm) and $\beta_{i,j}$ the amplitude of a trigonometric component i, j [11].

If the period is known, this model can be transformed into a linear form, which simplifies the regression process [9, 11]. The linear regression behind the cosinor model is used to determine the values of the regression coefficients. If we assume that these parameters are distributed according to a t-distribution, their confidence intervals as well as their p-values can be estimated using the t-test. Using the values of regression coefficients, the rhythmicity parameters can be calculated analytically in a one-component cosinor model, which is not possible when the number of components is increased. On the other hand, the rhythmicity parameters of a multicomponent cosinor model can be evaluated from the fitted curve, as described in [9]. This can be combined with bootstrap resampling to evaluate the confidence intervals

and p-values of rhythmicity parameters in a multicomponent cosinor model.

CosinorPy2 implements bootstrapping as a means of assessing the confidence intervals as well as the significance of a rhythmicity parameter. The observed measurements are resampled with repetitions (bootstrap samples), and a cosinor model is constructed from each of the bootstrap samples. The rhythmicity parameters are evaluated against these models to construct their bootstrap distributions. These are than used to evaluate the significance and confidence intervals of each of the observed rhythmicity parameters.

In addition, bootstrapping can also be applied to the evaluation of the significance of differential rhythmicity between two datasets. In a multicomponent cosinor model, this can be performed by evaluating confidence intervals of mean differences (for example, amplitudes) and assessing their corresponding p-values, with the null hypothesis being that the means are equal.

Another benefit of using bootstrapping to assess the distribution of rhythmicity parameters is that an arbitrary metric, that can be numerically evaluated on the fitted curve, can be analysed. For example, one could also opt to analyse the asymmetry of peaks, centre of gravity of the fitted curve, etc. Even though this would require manipulation of the source code of the package, the extensions would be straightforward, since only the function used to evaluate the rhythmicity parameters on a fitted curve would require a minor modification.

2.2. From a cosinor to a generalised cosinor model

A generalised single-component cosinor model can be described as follows:

$$y(t) = A + \beta \cdot e^{C \cdot t} \cdot \cos\left(\frac{2\pi}{\tau} \cdot t - \phi\right) + D \cdot t + E(t), \tag{2}$$

where A is MESOR, β is the amplitude of oscillation, C is the amplification coefficient, D is the linear component, *phi* is the acrophase, τ is the period, t is the time, and $E(t)$ is the error term. The amplification coefficient defines whether the oscillations are sustained ($C = 0$), forced ($C > 0$), or damped with time ($C < 0$). This model cannot be expressed in a linear form. To estimate its coefficients, a nonlinear regression must be performed [9, 14]. The p-values of the individual coefficients and their corresponding confidence intervals can be evaluated using the covariance matrix obtained within the regression process.

A generalised cosinor model can be used to describe oscillations that are forced or damped over time in a manner similar to ECHO (extended circadian harmonic oscillator) [18]. In addition, the proposed generalised cosinor model can also describe linear trends within the data, namely the accumulation or reduction of MESOR. Furthermore, the proposed model can be extended to perform a comparative analysis of the rhythmicity parameters between two

datasets:

$$
\begin{aligned}
y(t) = &(A + G \cdot A_0) + \\
&\beta \cdot e^{(C+G\cdot C_0)\cdot t} \cdot \cos\left(\frac{2\pi}{\tau} \cdot t - (\phi + G \cdot \phi_0)\right) + \\
&(D + G \cdot D_0) \cdot t + E(t),
\end{aligned} \tag{3}
$$

where G defines the dataset and is equal to 0 if a measurement belongs to the first dataset and 1 if it belongs to the second dataset. The coefficients A_0, B_0, C_0, D_0 and ϕ_0 define the differences between the datasets. Their corresponding p-values can be used to assess whether the differences between the datasets are significant for each rhythmicity parameter.

A generalised cosinor model can be extended to a multicomponent generalised cosinor model:

$$
y(t) = A + e^{C\cdot t} \cdot \sum_{i=1}^{N} \left(\beta_i \cdot \cos\left(\frac{2\pi}{\tau/i} \cdot t - \phi_i\right)\right) + D \cdot t + E(t), \tag{4}
$$

where β_i represents the amplitude of component i and ϕ_i its acrophase. As in the case of a basic multicomponent cosinor model, bootstrap resampling can be applied to evaluate the confidence intervals and p-values of the rhythmicity parameters in a generalised multicomponent cosinor model (see Section 3.1).

In addition, the differential rhythmicity statistic between two datasets can be evaluated similarly as in a basic model, namely using bootstrap for the evaluation of amplitude changes and acrophase shifts. However, signif-

icance in change of MESOR (A), amplification coefficient (C) and linear component (D) can be assessed directly from the regression coefficients and their covariance matrices (see Section 3.2).

3. Results

3.1. Analysing different types of rhythmic data

We used synthetically generated data to demonstrate the functionalities of the extended CosinorPy package on some typical scenarios and their combinations. Namely, we generated data reflecting (1) symmetric and asymmetric oscillations with single or multiple peaks per period, (2) damped, sustained, and forced oscillations, and (3) with or without the accumulation of a linear component around which oscillations occur (MESOR). The generation process is summarised in Table 1 and available together with the presented data analysis pipeline as interactive Python Jupyter notebook at https://github.com/mmoskon/CosinorPy/blob/master/demo_independent_nonlin.ipynb.

First, we analysed the generated data using a basic multi-component cosinor model as described in [9]. CosinorPy was used to identify the optimal number of components in a model to describe the given data as accurately as possible. The results of the fitting process are visualised in Figure 1 and included as supplementary Table 1. A more detailed analysis of the results obtained with multi-component cosinor models was performed using bootstrap as described in Section 2 (see Supplementary Table 2). However,

Table 1: Parameters for synthetically generated data. Rhythmicity periods of the generated data were set to 24 hours in all scenarios. Sampling period was set to 2 hours and noise level to 30%. Each scenario had three replicates. Abbreviations and symbols: N – number of components in a cosinor model, ϕ – acrophase; amplif. – amplification; lin. – linear component

test	N	**sampling time**	ϕ	**amplif.**	**lin.**
sym	1	48 h	π	0	0
sym_lin_comp	1	48 h	0	0	0.1
asym	3	48 h	π	0	0
asym_lin_comp	3	48 h	0	0	0.1
sym_damped	1	72 h	0	-0.04	0
sym_forced	1	72 h	π	0.04	0
asym_damped	3	72 h	0	-0.04	0
asym_forced	3	72 h	π	0.04	0

as seen from Figure 1, this model was unable to accurately describe forced or damped oscillations (see sym_damped, sym_forced, asym_damped, asym_forced) and/or the accumulation of a linear component around which oscillations occur (see sym_lin_comp, asym_lin_comp).

We can generalise a single-component cosinor model to obtain a model in the Equation 2. We proceeded with our analysis using this model. Since it is based on non-linear regression, it is less robust than the basic cosinor model. However, the generalised model can account for additional dynamical aspects in the observed data, such as damped or forced oscillatory behaviour and accumulation of the MESOR. The results of the fitting process using a single-component generalised cosinor model are visualised in Figure 3 and summarised in Table 2 (a complete statistic of the evaluated models is available as Supplementary Table 3).

This model was able to adequately describe damped and sustained os-

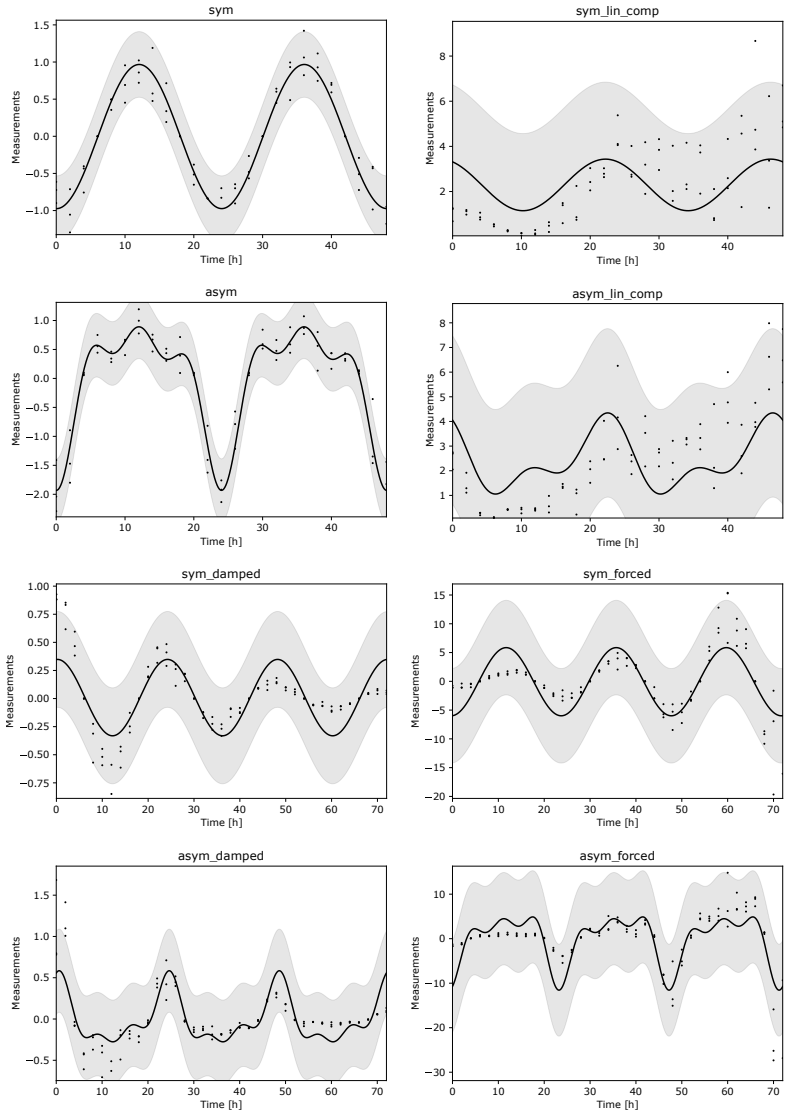

Figure 1: Results of the fitting process performed with a multi-component cosinor model. CosinorPy was used to identify the model that achieves the best fit for each scenario. The model is not able to account for forced/damped oscillations (sym_damped, sym_forced, asym_damped, asym_forced) and/or the accumulation of a linear component around which oscillations occur (sym_lin_comp, asym_lin_comp)

cillations as well as the accumulation of the MESOR. However, since it is based on a single-component cosinor, the model was unable to accurately de-

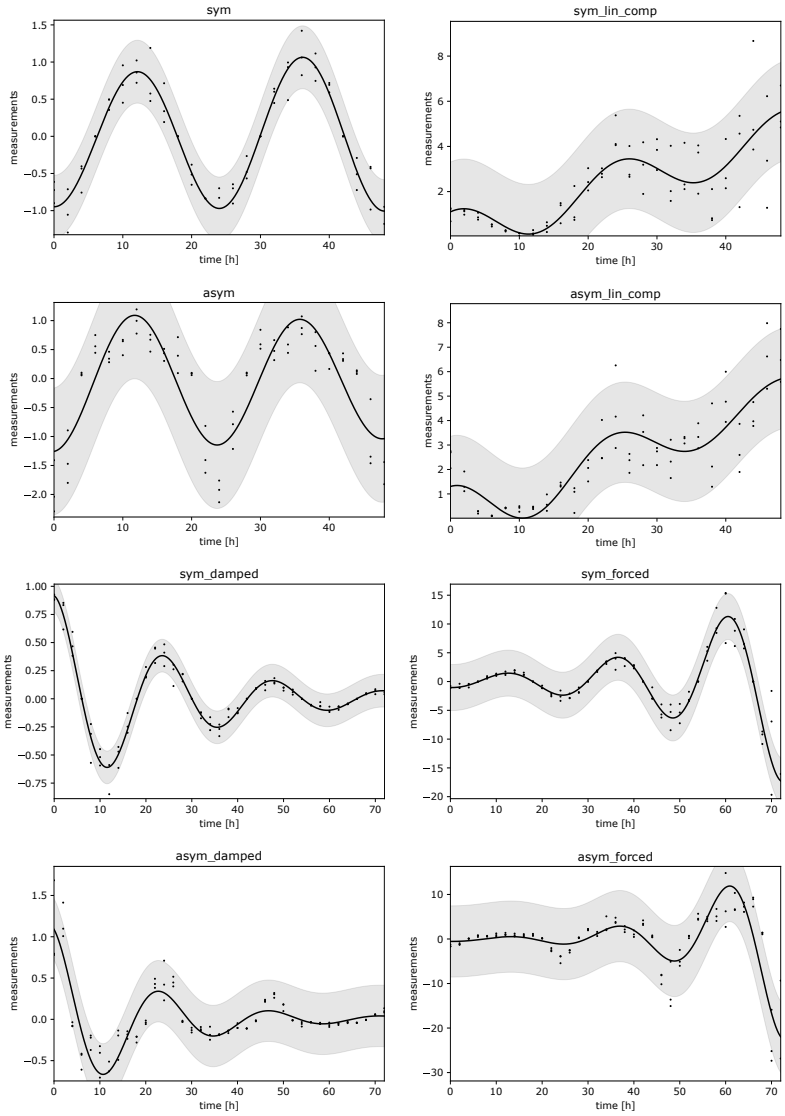

Figure 2: Results of the fitting process performed with a single-component generalised cosinor model. The model is not able to accurately describe the data reflecting asymmetric oscillations and/or multiple peaks per period (asym, asym_lin_comp, asym_damped, asym_forced). The model is able to account for forced/damped oscillations and/or the accumulation of a linear component around which oscillations occur

scribe the rhythmicity in the data reflecting asymmetric oscillations and/or multiple peaks per period (see Figure 2 and Figures asym, asym_lin_comp,

asym_damped, asym_forced). Moreover, even though the data in the test asym_forced reflects oscillatory behaviour, the test of rhythmicity period being different from zero was not significant after the adjustment for multiple testing using false discovery rate ($q = 0.062$, see Table 2).

To address this problem, we applied a generalised multi-component cosinor model as described in Equation 4. CosinorPy was used to assess the optimal number of components in a cosinor model for each scenario. The optimal number of components equalled 1 for all symmetric, and 3 for all asymmetric cases. The results of the fitting process are visualised in Figure 3, and the full statistics of the evaluated models are available as Supplementary Table 4. To identify the significance of the rhythmicity parameters, we ran the wrapper function that performs the bootstrap analysis of the obtained models. The results of this analysis are summarised in Table 3. As expected, oscillation amplitudes were significant in all cases. Acrophase values were significant in all the cases for which acrophases were set to π, namely sym, asym_lin_comp, sym_forced, and asym_forced. A complete statistics of the bootstrap analysis are available as Supplementary Table 5.

3.2. Comparative analysis of rhythmic data

Using a basic or a generalised single- or multicomponent cosinor model, CosinorPy is able to evaluate the differential rhythmicity for an arbitrary pair of measurements. We applied a generalised cosinor model to analyse the differences between the pairs (sym, sym_lin_comp), (asym, asym_lin_comp),

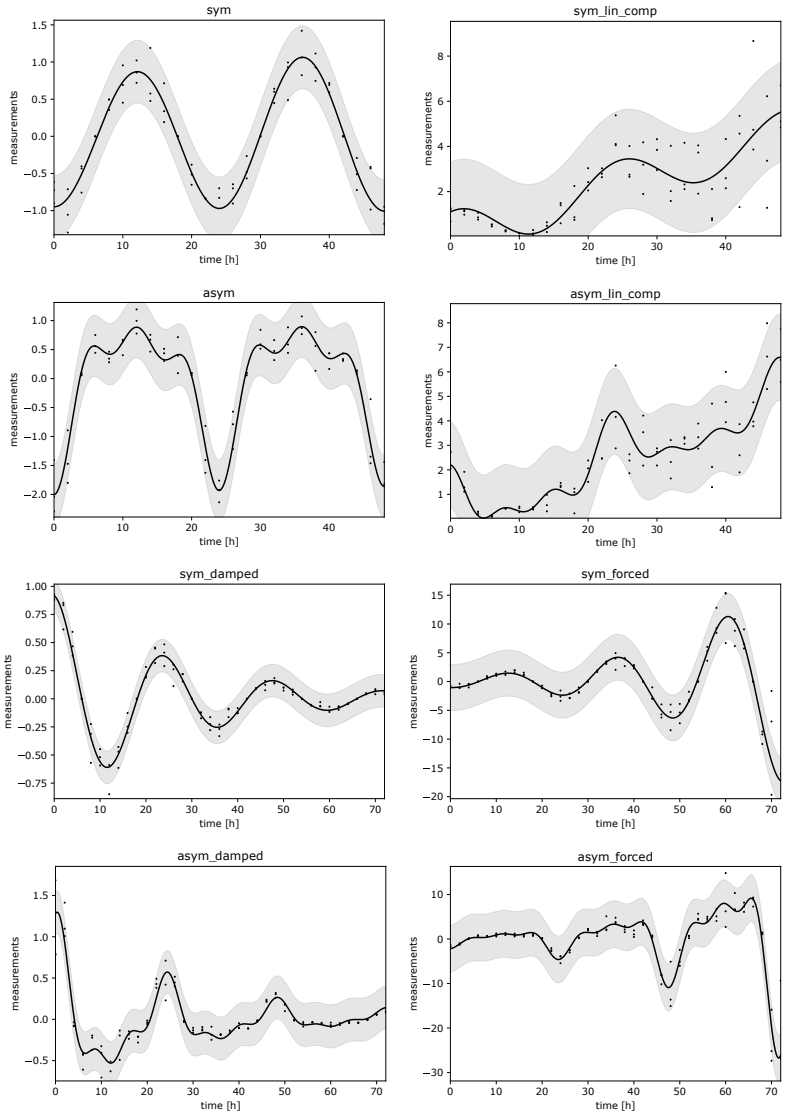

Figure 3: Results of the fitting process performed with a multi-component generalised cosinor model. CosinorPy was used to identify the model that achieves the best fit for each scenario. The model is able to describe the data reflecting asymmetric oscillation, multiple peaks per period, and is able to account for forced/damped oscillations and/or the accumulation of a linear component around which oscillations occur

(sym_damped, sym_forced), and (asym_damped, asym_forced). Comparisons are visualised in Figure 4. Within each pair, differential rhythmicity was

assessed using a wrapper function that performs a comparative bootstrap analysis. Summary of the results of this analysis are presented in Table 4 (a complete statistics is available as Supplementary Table 6).

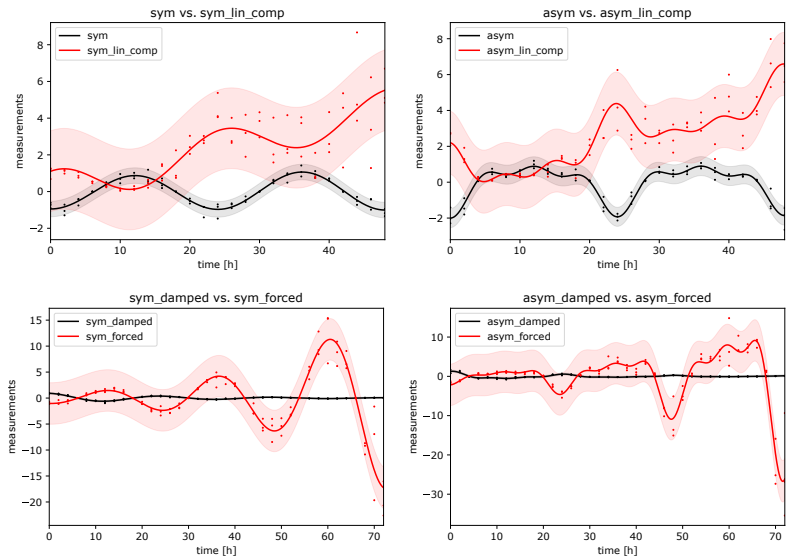

Figure 4: Results of the comparison analysis using a multi-component generalised cosinor model. The models that achieve the best fit for each scenario and identified in the previous steps were used in the analysis

3.3. Evaluating errors of different cosinor models

To rigorously assess the errors produced by different cosinor models, we generated a set of rhythmic data reflecting different types of oscillatory behaviour. Namely, we generated samples reflecting (1) symmetric and asymmetric oscillations with single or multiple peaks per period, and (2) samples reflecting damped, sustained, and forced oscillations. For each of the combinations of (1) and (2) 100 samples were generated. Moreover, each of these samples was combined with different levels of noise, which were set to 10%,

Table 2: Results of the analysis of a single-component generalised cosinor model. Abbreviations and symbols: A – amplitude; ϕ – acrophase; amplif. – amplification; lin. – linear component; $q(\cdot)$ – false-discovery rate adjusted p-value

test	A	**q(A)**	ϕ	**q(ϕ)**	**amplif.**	**q(amplif.)**	**lin.**	**q(lin.)**
sym	0.863	**0.0**	3.142	**0.0**	0.005	0.088	0.003	0.207
sym_lin_comp	1.071	**0.004**	-0.174	0.483	-0.001	0.910	0.093	**0.0**
asym	1.194	**0.0**	-3.079	**0.0**	-0.003	0.624	0.001	0.861
asym_lin_comp	1.306	**0.001**	0.016	0.932	-0.012	0.433	0.103	**0.0**
sym_damped	0.927	**0.0**	-0.017	0.599	-0.037	**0.0**	0.0	0.675
sym_forced	0.897	**0.0**	-3.118	**0.0**	0.041	**0.0**	0.009	0.675
asym_damped	1.127	**0.0**	0.148	0.044	-0.050	**0.0**	0.0	0.675
asym_forced	0.282	0.062	-3.142	**0.0**	0.061	**0.0**	0.015	0.675

test	N	A	**q(A)**	ϕ	**q(ϕ)**	**amplif.**	**q(amplif.)**	**lin.**	**q(lin.)**
sym	1	0.863	**0.0**	3.138	**0.0**	0.005	0.088	0.003	0.155
sym_lin_comp	1	1.071	**0.0**	-0.176	0.415	-0.001	0.910	0.093	**0.0**
asym	3	1.442	**0.0**	3.138	**0.0**	-0.001	0.768	0.001	0.631
asym_lin_comp	3	1.588	**0.0**	0.094	0.357	-0.005	0.609	0.102	**0.0**
sym_damped	1	0.927	**0.0**	-0.013	0.487	-0.037	**0.0**	0.000	0.631
sym_forced	1	0.897	**0.0**	-3.120	**0.0**	0.041	**0.0**	0.009	0.473
asym_damped	3	1.054	**0.0**	-0.138	0.128	-0.034	**0.0**	0.001	0.155
asym_forced	3	1.408	**0.0**	-2.981	**0.0**	0.037	**0.0**	0.017	0.318

Table 3: Results of the bootstrap analysis of best-fitted generalised cosinor models. Abbreviations and symbols: N – number of components in a cosinor model, A – amplitude; ϕ – acrophase; amplif. – amplification; lin. – linear component; $q(\cdot)$ – false-discovery rate adjusted p-value

test	N_1	N_2	**d**(A)	**q(d**(A))	**d**(ϕ)	**q(d**(ϕ))	**d(amplif.)**	**q(d(amplif.))**	**d(lin.)**	**q(d(lin.))**
sym vs. sym_lin_comp	1.0	1.0	0.208	0.769	2.969	**0.0**	-0.006	0.616	0.090	**0.000**
asym vs. asym_lin_comp	3.0	3.0	0.146	0.872	-3.044	**0.0**	-0.004	0.616	0.100	**0.000**
sym_damped vs. sym_forced	1.0	1.0	-0.030	0.872	-3.107	**0.0**	0.078	**0.000**	0.009	0.365
asym_damped vs. asym_forced	3.0	3.0	0.354	0.769	-2.843	**0.0**	0.070	**0.000**	0.016	0.309

Table 4: Results of the comparative analysis. Abbreviations and symbols: N_1, N_2 – number of components in a cosinor model; A – amplitude; ϕ – acrophase; amplif. – amplification; lin. – linear component; $d(\cdot)$ – difference of parameter; $q(\cdot)$ – false-discovery rate adjusted p-value

20%, 50%, and 100% of the generated data. We used the generated samples in a combination with different cosinor models, namely a single-component cosinor model (C1), a single-component generalised cosinor model (C1G), a 3-component cosinor model (C3), and a 3-component generalised cosinor model (C3G). We evaluated the mean square error (MSE) of evaluated oscillation amplitudes and acrophases for each of these models. MSE of evaluated oscillation amplitudes are presented in Table 5, and MSE of evaluated oscillation acrophases in Table 6. The distribution of errors for noise levels of 10%, 20% and 50% are visualised in Figures 5 and 6. The whole analysis is available as an interactive Python Jupyter notebook at https://github.com/mmoskon/CosinorPy/blob/master/analyse_errors.ipynb.

The results indicate that the proposed generalised cosinor models can accurately reproduce the oscillatory signal even when oscillations are damped or forced. As already shown in our previous analyses, asymmetric oscillations are better reproduced with a multi-component cosinor model. However, when the noise is increased above 50% of the signal, it is better to use a model with a smaller number of components to reduce the error, especially if oscillations are damped. This derives from the fact that a model with a larger number of components can be easily overfitted to the noisy data. When noise becomes prevalent, simpler models tend to perform better and are more robust. This problem can be controlled by the selection of the most suitable model for each dataset separately. Namely, we should opt to select the model that is able to describe the dataset with the smallest number of components. This can be

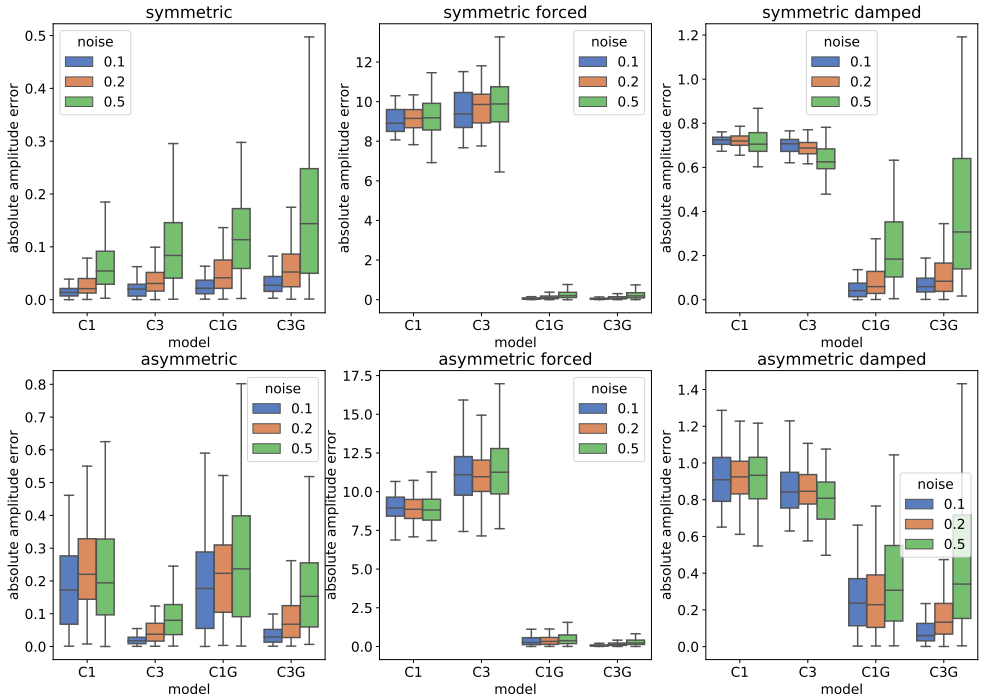

Figure 5: Distributions of amplitude errors for different cosinor models using different types of oscillations and different levels of noise. Abbreviations: C1 – single-component cosinor model; C3 – 3-component cosinor model; C1G – generalised single-component cosinor model; C3G – generalised 3-component cosinor model

performed automatically using the functionalities as provided by CosinorPy [9].

4. Conclusions

The extended version of the CosinorPy package provides a suite of methods that can be used in the analysis of different types of rhythmic data. Most importantly, it can be used to assess the statistics of the rhythmicity parameters even when the observed data reflect multiple asymmetric peaks per one period of oscillations. Moreover, the extended package provides an

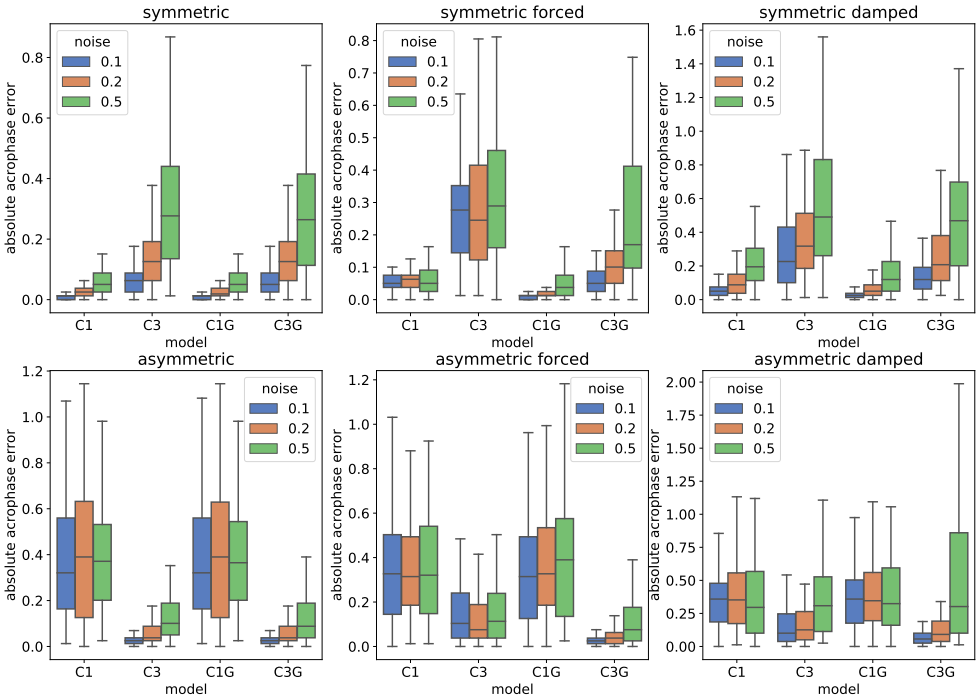

Figure 6: Distributions of acrophase errors for different cosinor models using different types of oscillations and different levels of noise. Abbreviations: C1 – single-component cosinor model; C3 – 3-component cosinor model; C1G – generalised single-component cosinor model; C3G – generalised 3-component cosinor model

implementation of generalised single- and multicomponent cosinor models, which can describe damped or forced oscillations as well as the accumulation of the MESOR. The statistics of assessed fits can be evaluated even with these relatively complex models using bootstrapping of confidence intervals and p-values. We believe that the proposed extensions of CosinorPy will find even wider applicability than its original implementation.

Conflict of interest statement

The author declares that he has no conflict of interest.

oscillations	amplification	noise	C1	C1G	C3	C3G
symmetric	none	0.1	**0.0**	**0.001**	**0.001**	**0.002**
		0.2	**0.001**	**0.004**	**0.002**	**0.005**
		0.5	**0.008**	0.033	0.015	0.060
		1.0	**0.022**	0.132	0.058	1.577
	forced	0.1	82.088	**0.006**	92.712	**0.005**
		0.2	83.629	**0.025**	95.586	**0.016**
		0.5	85.149	**0.122**	101.800	**0.088**
		1.0	85.130	**0.441**	113.664	**0.418**
	damped	0.1	0.521	**0.004**	0.495	**0.009**
		0.2	0.521	**0.015**	0.476	**0.034**
		0.5	0.514	**0.094**	0.403	0.995
		1.0	**0.495**	1.482	**0.279**	8.727
asymmetric	none	0.1	0.051	0.057	**0.001**	**0.002**
		0.2	0.074	0.081	**0.003**	0.012
		0.5	0.069	0.105	**0.017**	**0.059**
		1.0	0.101	0.173	**0.048**	0.218
	forced	0.1	81.771	0.234	127.804	**0.008**
		0.2	79.448	0.225	126.010	**0.035**
		0.5	80.331	0.385	136.439	**0.154**
		1.0	77.913	**0.722**	142.018	**0.690**
	damped	0.1	0.876	0.105	0.752	**0.011**
		0.2	0.875	0.099	0.742	**0.041**
		0.5	0.874	**0.371**	0.652	1.704
		1.0	0.753	1.140	**0.414**	8.885

Table 5: Mean square errors of evaluated oscillation amplitudes. Abbreviations: C1 – single-component cosinor model; C3 – 3-component cosinor model; C1G – generalised single-component cosinor model; C3G – generalised 3-component cosinor model

Funding

This work has been partially supported by the scientific-research program P2-0359, and by the basic research projects J1-9176 and J5-1798, all financed by the Slovenian Research Agency. The funding body had no role in the design of the study and collection, analysis, and interpretation of data nor in writing the manuscript.

oscillations	amplification	noise	C1	C1G	C3	C3G
symmetric	none	0.1	**0.0**	**0.0**	0.006	0.006
		0.2	**0.001**	**0.001**	0.027	0.028
		0.5	**0.006**	**0.006**	0.132	0.126
		1.0	**0.031**	**0.030**	0.248	0.228
	forced	0.1	**0.004**	**0.0**	0.120	**0.004**
		0.2	**0.004**	**0.001**	0.120	0.017
		0.5	**0.007**	**0.004**	0.147	0.111
		1.0	**0.013**	**0.015**	0.218	0.271
	damped	0.1	**0.005**	**0.001**	0.122	0.031
		0.2	0.017	**0.007**	0.187	0.121
		0.5	0.079	**0.046**	0.457	0.384
		1.0	**0.354**	**0.361**	0.798	0.863
asymmetric	none	0.1	0.227	0.228	**0.043**	**0.042**
		0.2	0.274	0.275	**0.122**	**0.154**
		0.5	0.226	0.225	**0.182**	**0.181**
		1.0	**0.287**	**0.287**	0.508	0.522
	forced	0.1	0.214	0.209	0.320	**0.043**
		0.2	0.200	0.211	0.253	**0.032**
		0.5	**0.223**	0.241	0.283	**0.208**
		1.0	**0.241**	**0.265**	0.324	0.541
	damped	0.1	0.209	0.215	0.277	**0.088**
		0.2	0.252	0.257	**0.204**	**0.220**
		0.5	**0.240**	**0.258**	0.386	0.579
		1.0	**0.582**	**0.549**	1.055	1.071

Table 6: Mean square errors of evaluated oscillation acrophases. Abbreviations: C1 – single-component cosinor model; C3 – 3-component cosinor model; C1G – generalised single-component cosinor model; C3G – generalised 3-component cosinor model

Acknowledgements

I would like to thank all the past, current and future users of CosinorPy for using the package and for providing the critical feedback, which allows the package to progress further.

References

[1] M. E. Hughes, J. B. Hogenesch, K. Kornacker, JTK_CYCLE: an efficient nonparametric algorithm for detecting rhythmic components in genome-scale data sets, Journal of biological rhythms 25 (5) (2010) 372–380.

[2] P. F. Thaben, P. O. Westermark, Detecting rhythms in time series with RAIN, Journal of biological rhythms 29 (6) (2014) 391–400.

[3] A. L. Hutchison, M. Maienschein-Cline, A. H. Chiang, S. A. Tabei, H. Gudjonson, N. Bahroos, R. Allada, A. R. Dinner, Improved statistical methods enable greater sensitivity in rhythm detection for genome-wide data, PLoS Comput Biol 11 (3) (2015) e1004094.

[4] A. L. Hutchison, R. Allada, A. R. Dinner, Bootstrapping and empirical bayes methods improve rhythm detection in sparsely sampled data, Journal of biological rhythms 33 (4) (2018) 339–349.

[5] E. Ness-Cohn, R. Braun, TimeCycle: Topology Inspired MEthod for the Detection of Cycling Transcripts in Circadian Time-Series Data, Bioinformatics (2021).

[6] R. C. Anafi, L. J. Francey, J. B. Hogenesch, J. Kim, Cyclops reveals human transcriptional rhythms in health and disease, Proceedings of the National Academy of Sciences 114 (20) (2017) 5312–5317.

[7] M. D. Ruben, G. Wu, D. F. Smith, R. E. Schmidt, L. J. Francey, Y. Y.

Lee, R. C. Anafi, J. B. Hogenesch, A database of tissue-specific rhythmically expressed human genes has potential applications in circadian medicine, Science Translational Medicine 10 (458) (2018).

[8] M. D. Ruben, L. J. Francey, Y. Guo, G. Wu, E. B. Cooper, A. S. Shah, J. B. Hogenesch, D. F. Smith, A large-scale study reveals 24-h operational rhythms in hospital treatment, Proceedings of the National Academy of Sciences 116 (42) (2019) 20953–20958.

[9] M. Moškon, CosinorPy: a Python package for cosinor-based rhythmometry, BMC bioinformatics 21 (1) (2020) 485.

[10] M. M. Doyle, T. E. Murphy, M. A. Pisani, H. K. Yaggi, S. Jeon, N. S. Redeker, M. P. Knauert, A SAS macro for modelling periodic data using cosinor analysis, Computer Methods and Programs in Biomedicine (2021) 106292.

[11] G. Cornelissen, Cosinor-based rhythmometry, Theoretical Biology and Medical Modelling 11 (1) (2014) 16.

[12] C. Bingham, B. Arbogast, G. C. Guillaume, J. K. Lee, F. Halberg, Inferential statistical methods for estimating and comparing cosinor parameters., Chronobiologia 9 (4) (1982) 397–439.

[13] J. M. Singer, J. J. Hughey, LimoRhyde: a flexible approach for differential analysis of rhythmic transcriptome data, Journal of biological rhythms 34 (1) (2019) 5–18.

[14] R. Parsons, R. Parsons, N. Garner, H. Oster, O. Rawashdeh, Circacompare: a method to estimate and statistically support differences in mesor, amplitude and phase, between circadian rhythms, Bioinformatics 36 (4) (2020) 1208–1212.

[15] Y. L. Tong, Parameter estimation in studying circadian rhythms, Biometrics (1976) 85–94.

[16] M. A. Luque-Fernandez, C. V. Ananth, S. E. Sanchez, C.-F. Qiu, S. Hernandez-Diaz, U. Valdimarsdottir, B. Gelaye, M. A. Williams, Absence of circadian rhythms of preterm premature rupture of membranes and preterm placental abruption, Annals of epidemiology 24 (12) (2014) 882–887.

[17] Š. Verovšek, T. Zupančič, M. Juvančič, S. Petrovčič, M. Svetina, M. Janež, Ž. Pušnik, I. L. Bajec, M. Moškon, The aspect of mobility and connectivity while assessing the neighbourhood sustainability, Academic Journal of Interdisciplinary Studies 10 (3) (2021) 37–48.

[18] H. De los Santos, E. J. Collins, C. Mann, A. W. Sagan, M. S. Jankowski, K. P. Bennett, J. M. Hurley, ECHO: an application for detection and analysis of oscillators identifies metabolic regulation on genome-wide circadian output, Bioinformatics 36 (3) (2020) 773–781.

[19] A. Pelikan, H. Herzel, A. Kramer, B. Ananthasubramaniam, Venn dia-

gram analysis overestimates the extent of circadian rhythm reprogramming, The FEBS Journal (2021).

[20] U. Kovač, Z. Žužek, L. Raspor Dall'Olio, K. Pohar, A. Ihan, M. Moškon, D. Rozman, M. Starčič Erjavec, Escherichia coli affects expression of circadian clock genes in human hepatoma cells, Microorganisms 9 (4) (2021) 869.

[21] J. Palotti, I. Perez-Pozuelo, A. Al Homaid, R. Mall, M. Posa, H. Berlin, D. Jin, F. Farooq, P. Szolovits, Predicting several sleep quality metrics based on same day physical activity, in: AI for Public Health Workshop at ICLR'21, 2021, pp. 1–8.

[22] M. Sachs, cosinor: Tools for estimating and predicting the cosinor model, R package version 1.1 (2014).
URL https://CRAN.R-project.org/package=cosinor

[23] H. De los Santos, K. P. Bennett, J. M. Hurley, MOSAIC: a joint modeling methodology for combined circadian and non-circadian analysis of multi-omics data, Bioinformatics 37 (6) (2021) 767–774.