Wavelet analysis: Detecting rhythms in time series when rhythms vary across time

Wavelet analysis detects rhythms in time series data. A ‘rhythm’, here, refers to a repeating pattern with a certain amplitude and frequency. The amplitude reflects the size of the signal compared to its baseline, while the frequency describes how often the pattern repeats. Within biology, common interesting frequencies include diurnal and annual rhythms.

Most statistical methods to detect rhythms, such as the Fast Fourier Transform and harmonic regression, can only determine repetition under the assumption that the repeating frequency is similar across the entire time series. Wavelet analysis, in contrast, can detect rhythms not only at a range of frequencies but also at different locations in the time series. This means that it could for example be detected whether diurnal rhythms differ between seasons or change as animals age. Therefore, wavelet analysis is useful when it is expected that the rhythm changes across the time series.

Conceptually, wavelet analysis starts with a mother wavelet, which is a rhythmic signal that reflects a certain form of behavioural repetition. This mother wavelet can be ‘stretched’ into different frequencies, and each such ‘stretch’ is referred to as a daughter wavelet. These daughter wavelets can be wider, for low frequencies (i.e. slow repetitions), or narrower, for high frequencies (i.e. fast repetitions). The daughter wavelets are shifted across the time series, and at each point it is calculated how well the daughter wavelet and the time series correspond; a variable referred to as ‘power’. The power, therefore, reflects the strength of the rhythm at that particular frequency and time point. For more information on the mathematics of wavelet analysis, we refer to Torrence and Compo (1998), Cazelles et al. (2008) and Leise (2015; 2017).

Wavelet analysis can be performed on individual animals, and the range of frequencies to be detected are chosen by the user. By default, wavelet analysis creates visual output, but this output can also be quantified and the significance of the power can be tested. The applications of wavelet analysis are, therefore, quite versatile. In our case, we used it to detect circadian rhythms (i.e. repeating every 24 h) in individual pig feeding behaviour across the growing-finishing phase. Other applications could, for example, be to look at seasonal changes in animal migration across years, or to detect ultradian (i.e. repeating more often than every 24 h) rhythms at different frequencies.

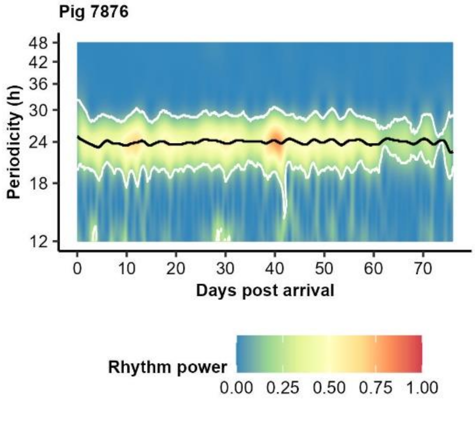


Figure 1. Example of a figure obtained via wavelet analysis.

Things to consider when performing wavelet analysis

There are two main approaches to wavelet analysis: the continuous and the discrete wavelet transform. In the continuous wavelet transform, it is directly measured how frequency and amplitude vary over time by continuously translating the daughter wavelets against the time series. In the discrete wavelet transform, in contrast, the time series is decomposed into a limited number of components that each correspond to a particular frequency. Consequently, the discrete wavelet transform is faster but less precise. In this example, we used the continuous wavelet transform, and we will not discuss the discrete approach further.

Before, it was described that wavelet analysis starts with a mother wavelet, which reflects a pre-chosen form of behavioural repetition. From this it follows that wavelet analysis depends on the type of mother wavelet chosen, as it determines the types of rhythms that can be detected. There are very many types of mother wavelets. In our studies, we used a relatively standard Morlet wavelet, but to determine what works for your dataset, it is good to repeat the wavelet analysis with a few types of mother wavelets (e.g. Morse or Haar wavelets, that have a good theoretical fit to your data.

For more information on these alternative approaches, we refer to Torrence and Compo (1998), Cazelles et al. (2008) and Leise (2015; 2017).

A step-by-step approach, with an example in R for circadian rhythms in pig feeding behaviour

This section presents a step-by-step approach to performing wavelet analysis. It does not dive into the mathematics, but rather aims to point out the steps to take in processing and checking the data, performing the analysis and checking and interpreting the results. The process is illustrated with some of our own data on growing-finishing pig feeding behaviour, an excerpt of which is supplied with this tutorial. Code for walking through this example is also supplied with this tutorial.

Before we start

The example was coded in R, hence an installed version of R and basic knowledge on R coding is required. The tutorial was created in R version 4.3.1 on a Windows laptop, and using RStudio as additional interface. Wavelet analysis is performed, in this tutorial, with the *WaveletComp* package, version 1.1 (Roesch & Schmidbauer, 2018). Other packages used can be seen in the script, make sure to install these.

Step 1 – Determining the research question and data collection

This step was probably taken long before this tutorial, but is good to come back to as it partially determines whether wavelet analysis is suitable to answer the questions. In our case, our research question was to determine whether pigs showed circadian rhythms in their feeding behaviour, and how big the range in circadian rhythm strength was between individual pigs. Ultimately, we wanted to compare this to different indicators of pig welfare, but that is not included within this tutorial.

Data was collected on all visits of 110 pigs to an IVOG® electronic feeding station, during one growing-finishing phase. Pigs were individually recognised using radio-frequency identification (RFID), and the feed intake (kg) of each feeder visit was recorded. Visit intakes were summed to hourly feed intakes, which were used as input into the wavelet analysis. For this tutorial, we only use a selection of five pigs with diverse feeding patterns. Data had already been processed, cleaned and checked before it was imported into this tutorial (described in Bus et al., 2023). As sensor data is particularly prone to errors, these steps that safeguard data quality should be performed before moving on with the next steps.

Step 2 – Getting to know the data

Data visualisations help to make appropriate choices and interpret the analysis results correctly. The ranges and trends of individual samples, here pigs, should be known, as analyses will be performed at the individual level. First visualise the general time trends (Figure 2), which in this case means a line plot of the daily feed intake of individual pigs across the growing-finishing phase. As expected, intake shows an increasing trend for all pigs, from about 1-4 kg/day. Pig 8240 has some missing data at the end of the growing-finishing phase, and pig 8281 at two days around d55-59.

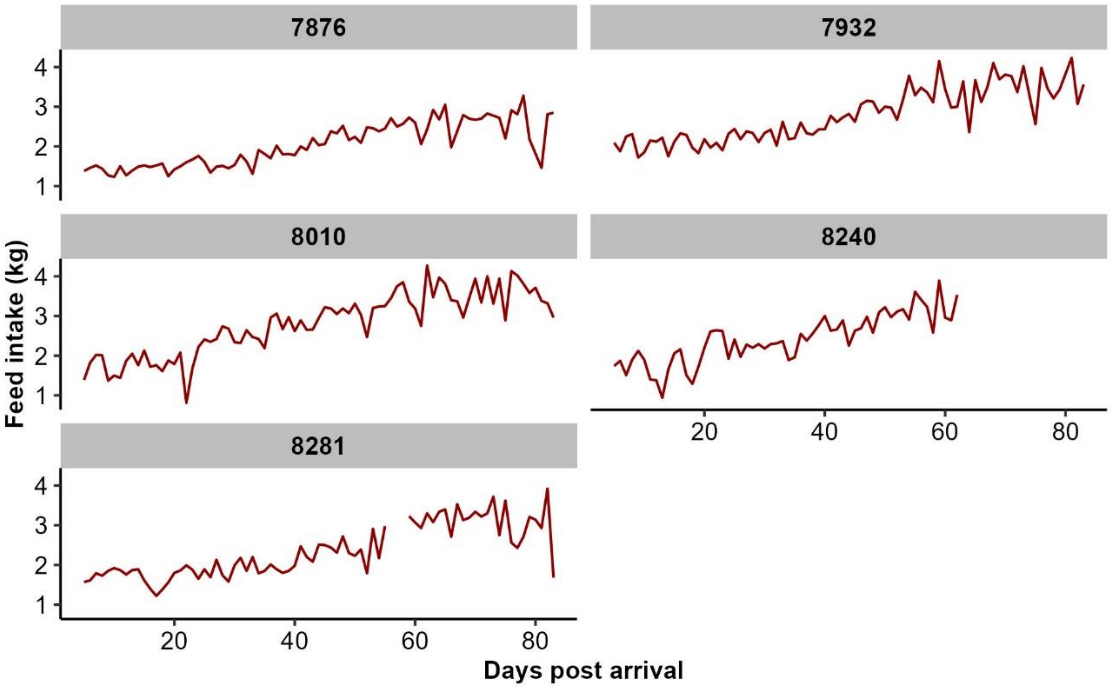


Figure 2. Daily trends in feed intake of individual pigs.

Next, our visualisations should zoom into the level at which we will measure the repetition. In this case, we decided to aggregate the data at an hourly level. The hourly level was chosen as it can reflect diurnal patterns, while avoiding inclusion of too many zeros into the data created by moments on which pigs did not eat. For other types of data, such as from accelerometers, it may be feasible to use a more precise time frame, such as per minute or ten minutes. In contrast, to look at seasonal differences, monthly aggregations may be more appropriate. Making the aggregation choice warrants some visualisations and trial analyses, by simply repeating all steps at different aggregation levels.

Figure 3 presents the hourly feed intakes of our selected pigs across the growing-finishing phase. The darkening colours from top to bottom again show that all pigs increased their feed intake across days. The timing of feeding differed between individuals. For example, pig 7876 mostly ate during the day and little at night from the beginning of the growing-phase until the end, while pig 7932 slowly shifted its feeding to later in the day, pig 8281 mostly ate during the day and evening, and pigs 8010 and 8240 ate throughout the day and night. The changes in timing of feeding across the growing-finishing phase suggests that rhythms may also change over time, confirming wavelet analysis as an interesting analysis.

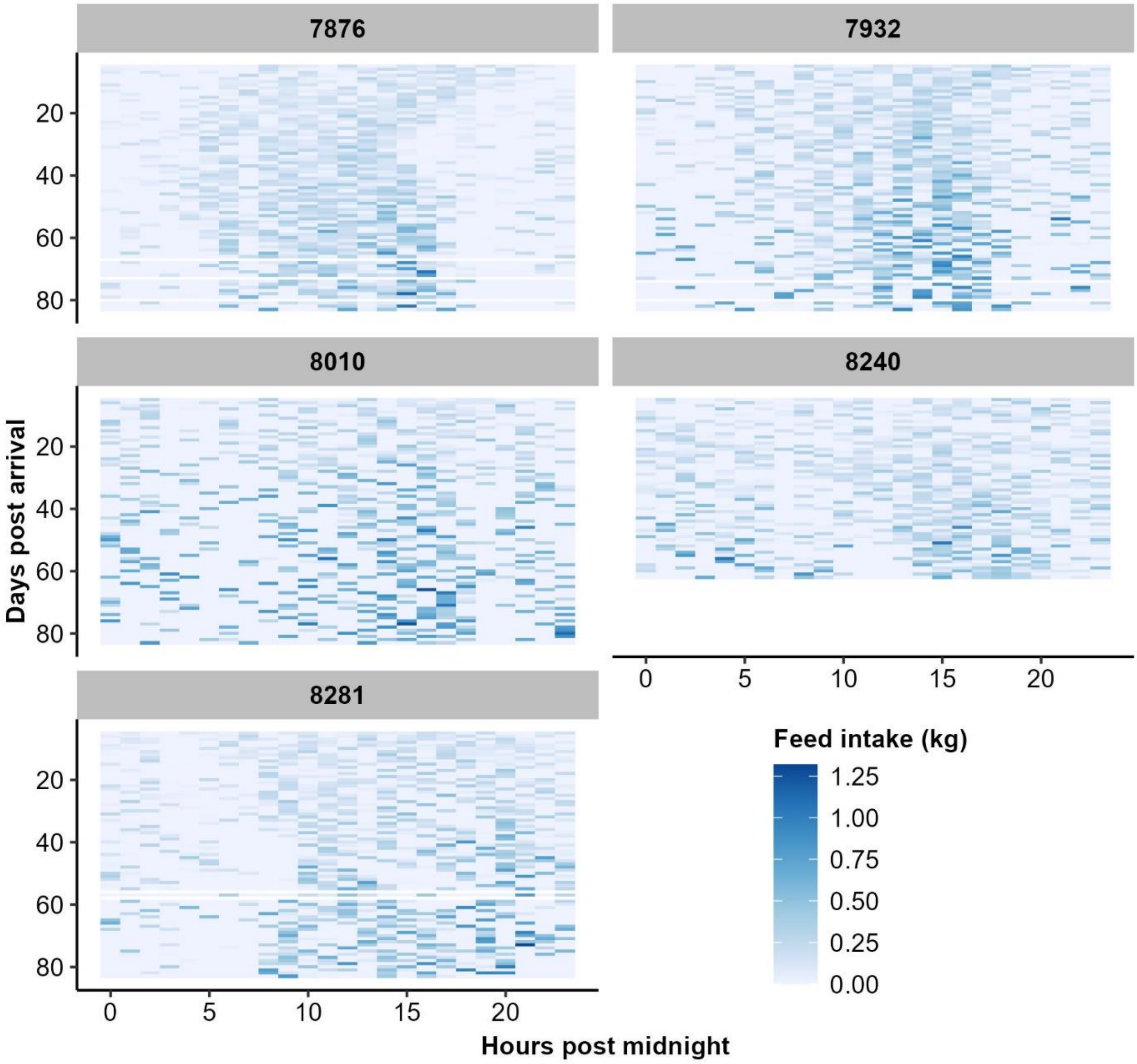
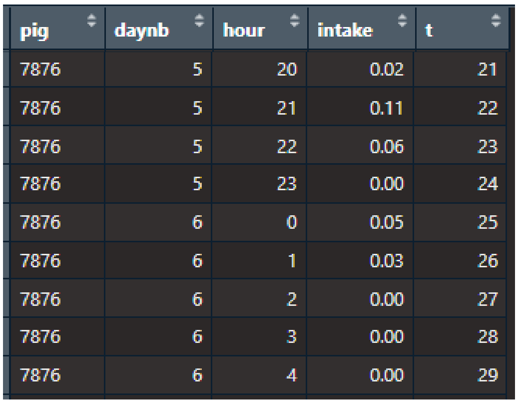


Figure 3. Heat maps of the hourly feed intake of individual pigs across the growing-finishing phase.

Step 3 – Preparing the data for analysis

For analysis, the data must be sorted temporally in steps of equal distance, but no information is required on how this relates to external time keeping like hours or days. Currently, our data is described with a combination of hours (hour: 0-23) and days (daynb: 5-83), which can be simplified to an integer, 't', that simply counts upwards from the first (i.e. hour 0 at d5) to the last (i.e. hour 23 at d83) observation, see also the screenshot on the right.



pig	daynb	hour	intake	t
7876	5	20	0.02	21
7876	5	21	0.11	22
7876	5	22	0.06	23
7876	5	23	0.00	24
7876	6	0	0.05	25
7876	6	1	0.03	26
7876	6	2	0.00	27
7876	6	3	0.00	28
7876	6	4	0.00	29

Like most analyses, wavelet analyses make some assumptions about the data. There are three main things to consider: 1) the assumption of stationarity; 2) the assumption of equal amplitude; and 3) the inability to handle missing data.

Assumption of stationarity

The assumption of stationarity means that the data may not contain any trends over time. From Figure 2, it is already clear that in our data this assumption is violated: feed intake increases across time. This can be mediated by de-trending the data. We fitted a local regression curve (LOESS) and took the residuals of the fitted line for further analysis. These residuals reflect the difference between the hourly measured value and the general trend across time, hence representing the variation in hourly intake surrounding the trend.

Figure 4 shows the resulting data of the individual pigs. The height of the black vertical lines now reflect the hourly data that is used for further analysis. The red lines reflect any remaining trend in the data, and as these lines are all straight lines at 0 kg, there is no trend left. The data was successfully de-trended and the assumption of stationarity is now met.

Assumption of equal amplitude

The power of the rhythm estimated by the wavelet analysis is influenced not only by the moment of repetition but also by how large the amplitude of the rhythm is. The larger the amplitude, the higher the power tends to be. Although this is relevant, it becomes a problem if the amplitudes are consistently larger in some periods of the time series than in others. For example, if amplitudes are larger later in the time series compared to earlier, then the power of the rhythm will be consistently estimated as higher at later than at earlier moments.

Figure 4 can be used to assess whether the assumption of equal amplitude is violated, because the black lines reflect the amplitude of the signal. For all pigs, it can be seen that the amplitude is indeed larger at later ages than at earlier ages, as the black lines generally and more frequently reach higher values. This indicates that pigs ate relatively more in fewer hours at older ages, compared to relatively lower quantities during more hours at younger ages.

Correction for unequal amplitudes can be done using an amplitude envelope. The amplitude envelope represents the largest difference in amplitude within a certain time series, and is calculated by deducting the lowest value from the highest value (i.e. here the highest hourly intake minus the lowest hourly intake). Subsequently, all values are divided by this value, which standardises all values to the same range. For optimal results, this process is performed for small subsets of the time series. In this example, we used periods of seven days. Too long periods lead to too severe corrections, while too short periods do not fully remove the amplitude trends. It is advised to try out some different periods for amplitude envelope calculation and correction, and visualise its effects.

Figure 5 visualises the result. The black lines now reach similar amplitudes across the growing-finishing phase, and the trend line is still set at zero; the assumptions for both stationarity and equal amplitude are now met.

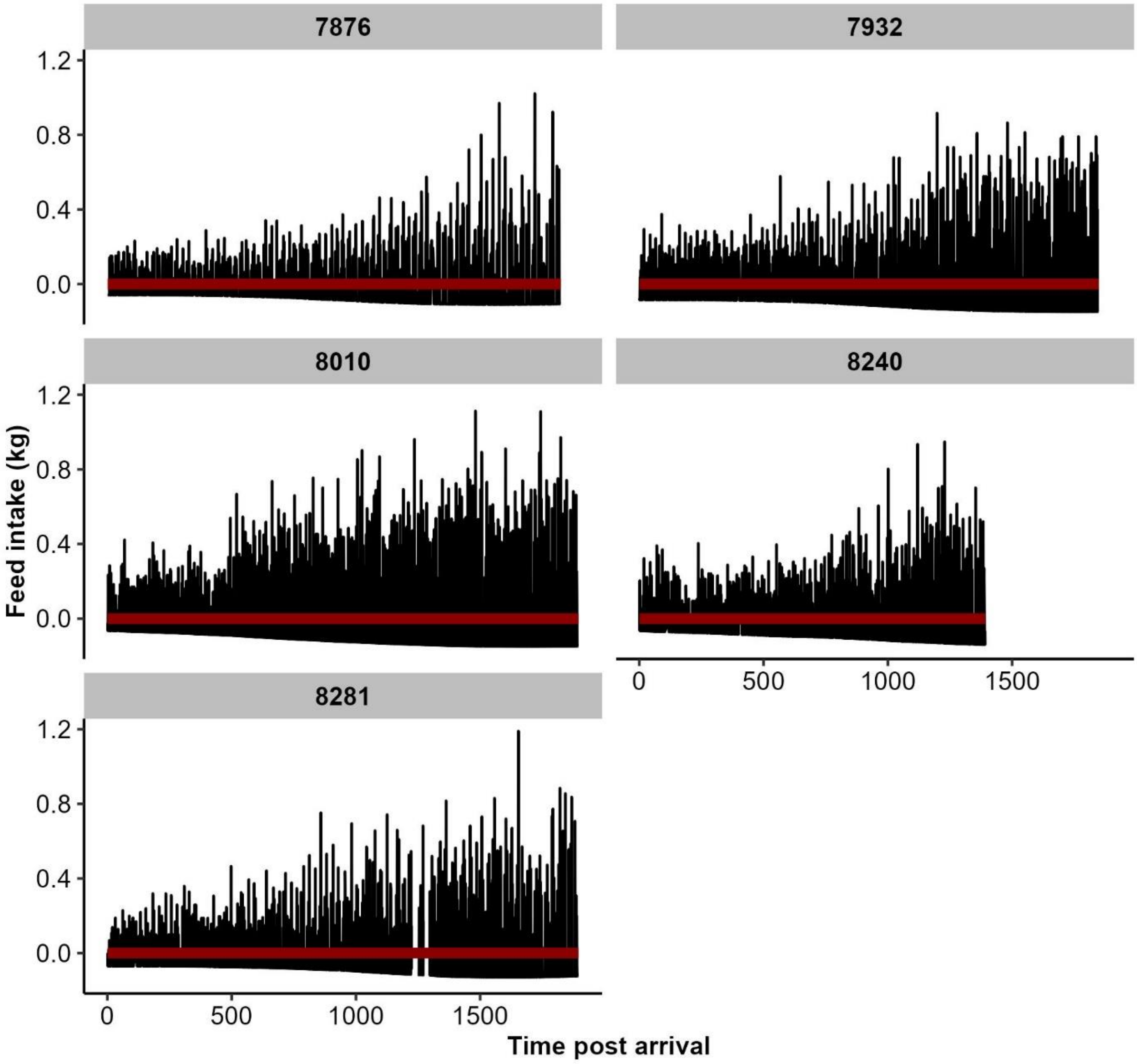


Figure 4. Hourly feed intake of individual pigs (black lines) after de-trending. Time values are increasing integers from arrival at the farm onwards, and the red line reflects the remaining trend in the data.

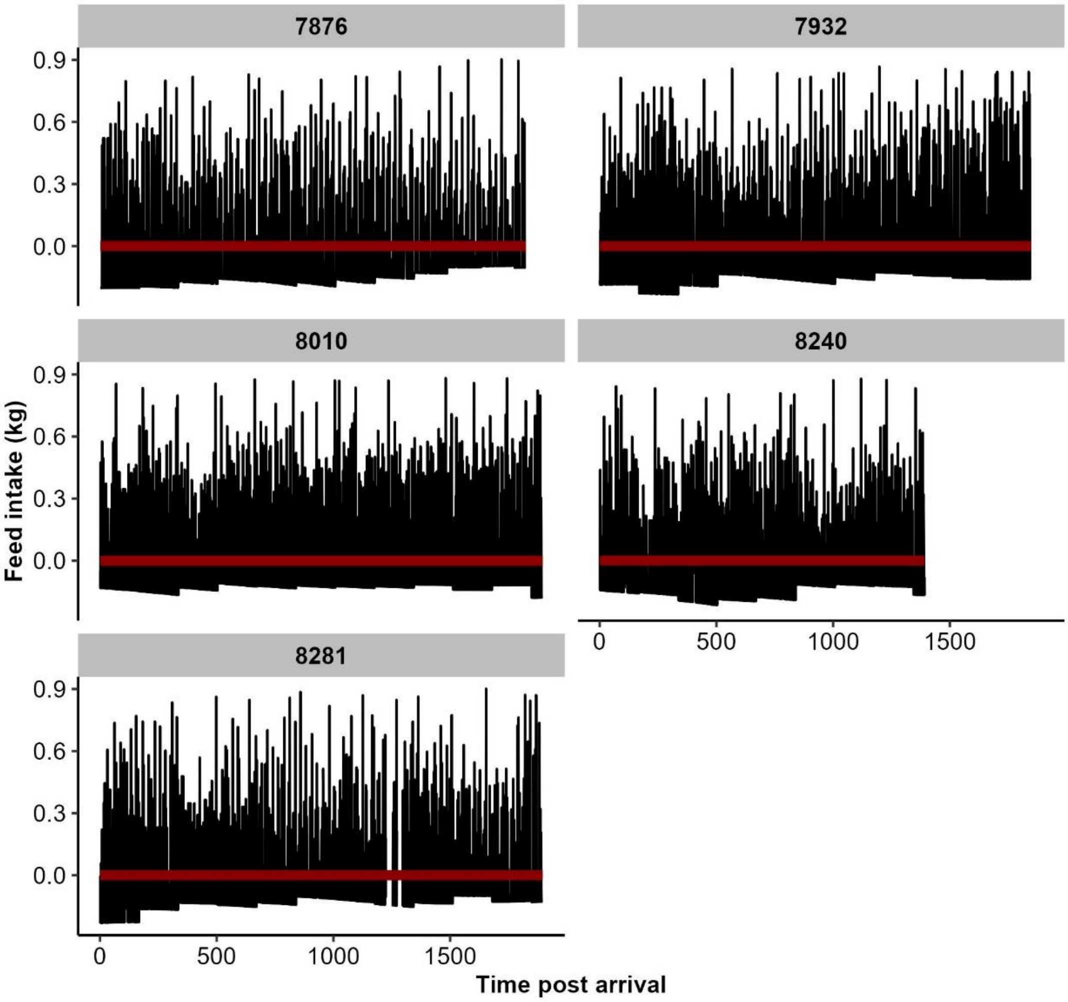


Figure 5. Hourly feed intake of individual pigs (black lines) after de-trending and amplitude correction. Time values are increasing integers from arrival at the farm onwards, and the red line reflects the trend in the data.

Removing missing data

Wavelet analysis cannot handle missing data, and the R function will give an error if an incomplete dataset is entered. This problem can be solved in different ways, depending on why the data are missing and the structure of the data. The main options are interpolation of missing data points, replacement of these points with a fixed value, or removal of all individuals that have any missing data. In our case, missing data were very common for the full dataset, though this is less visible in our selection of five example pigs. Removing all these pigs would have depleted our dataset, while interpolation would have artificially introduced structure into the data. Therefore, we chose to replace all missing values with zero's. To decide this for your own dataset, a good start is to plot the missing data, such as done in Figure 6.

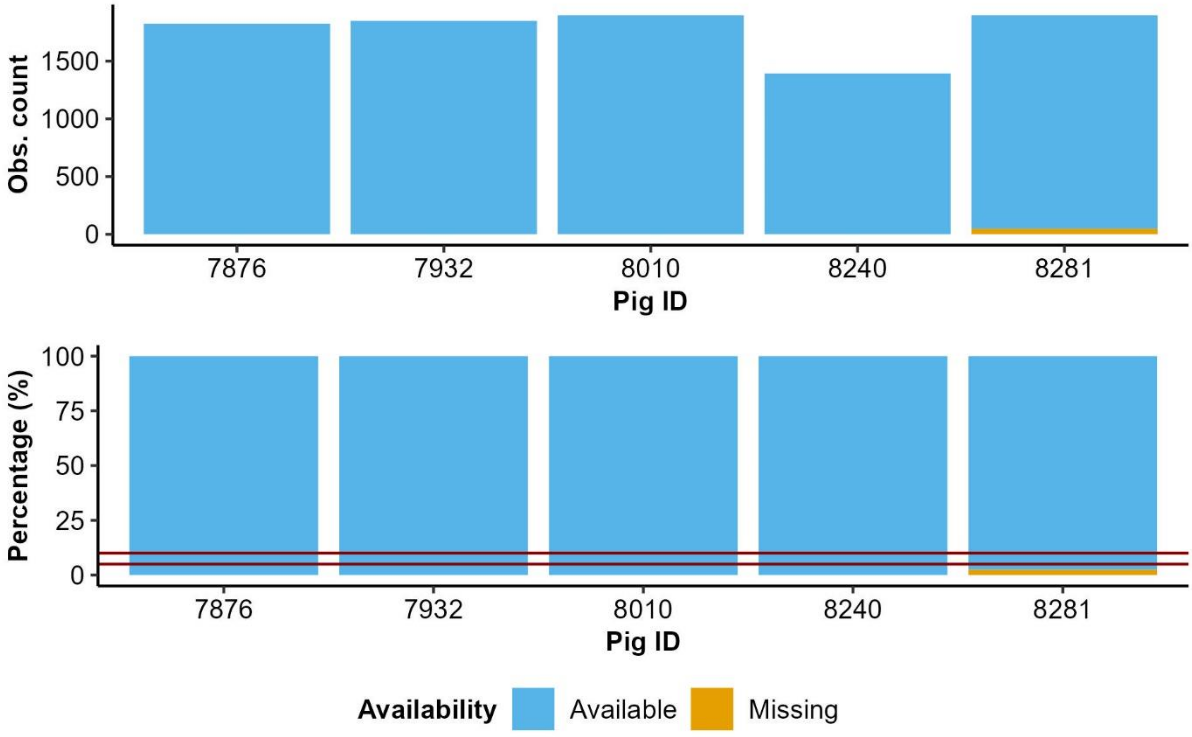


Figure 6. Missing and available observations per individual pig, in absolute number of observations (top) or relative percentages (bottom). The two red lines represent 5% and 10% of missing observations.

To check whether and how addition of these zeros would influence our analysis results, we performed a simulation in which we analysed time series with and without many zeros. We found that high presence of zeros in the data led to noise in the wavelet power spectra (the final output, see Step 5) at low frequencies, in this case between frequencies at approximately 4-10 h, but not at higher frequencies. As our main interest was on circadian rhythms, which repeat at 24 h, this was not considered an issue. In addition, our data already contained many zeros even before missing values were replaced, because there were many hours during which a pig did not eat. The important thing is that an approach to remove missing data should be chosen based on relevant arguments, and this will depend on the study aims, the dataset studied and the frequency of missing data.

Step 4 – Performing the analysis

The data have now been prepared for analysis. During the analysis, the daughter wavelets are translated across the time series at different frequencies, and the agreement between the data and each daughter wavelet is calculated as a variable called 'power'. This power hence reflects the strength of the rhythm at a certain frequency and moment.

The power can be interpreted relatively, with low values indicating no real rhythm and high values indicating a strong rhythm. In addition, significance of this power can be estimated by comparing it to simulated white noise. White noise contains all periodicities at about an equal level and hence estimating its power represents the power that could be expected at random. By repeatedly comparing the time series' powers to those of many white noise simulations (Monte Carlo simulation), a P-value can be calculated that reflects how likely it is that the time series' powers were stringer than random. By placing a threshold on the P-value (e.g. P < 0.05), significance of the power can be classified. Note that this is not an exact test but rather an approximation of significance.

In this tutorial, wavelet analysis was performed using the *analyze.wavelet*() function of the *WaveletComp* package. It was performed for each individual pig using a *for*-loop, the results of which are saved in a list ('wavelet.results'). In the script, first a simple analysis example is given. In Step 6, this example is enlarged with quantitative extractions of results. When using the code from Step 6, the code from Step 4 can be skipped.

The only input required for analysis is the hourly intake values. They should be sorted correctly, using the variable 't' we made previously. Other than that, a few choices should be made:

- lowerPeriod. The lowest frequency of interest. We were interested in circadian rhythms (i.e. 24 h), but you should always take a range around that as the analysis cannot achieve such precision. As we were also interested to see other rhythms, we chose a minimum frequency of 8 h.
- upperPeriod. The highest frequency of interest. We chose 48 h.
- dt. The time resolution. This determines how many steps each daughter wavelet is shifted each time it is translated against the time series. By default, dt is one, which is suitable for data that contains one observation per time moment (i.e. the daughter wavelet shifts one 't' each time). If this is not the case (i.e. t=1 has more or fewer than one measurement), dt should be changed to have 1/dt reflect the number of observations per time unit.
- dj. The frequency resolution. This determines by what steps between lowerPeriod and upperPeriod the frequency is varied when the daughter wavelets are translated against the time series. The larger the value (i.e. the smaller 1/dj), the more precise the analysis, though this takes more time and can also lead to overfitting. We used dj = 1/100.
- make.pval. This parameter determines whether statistical testing of the P-value should be performed ('TRUE', the default) or not ('FALSE'). Turning statistical testing off makes the analysis considerably faster, so if only the power values are required that is a good option.
- n.sim. The number of white noise simulations that the P-value calculation is based on. More simulations take more time, but also gives a more reliable estimate of the P-value. In our final analysis, we used n.sim = 1000, which took about 24 h to run for 110 pigs with 3 months of data (about 2160 datapoints per pig). For the purpose of this tutorial, we chose a smaller value of n.sim = 100, to speed up the analysis (took about 3 min for the 5 pigs).
- loess.span. The *analyze.wavelet*() function has a built-in method of smoothing and de-trending data, similar to what we did manually previously. By default, this is turned on, so in our case we should turn it off by setting it to zero.
- verbose. By default, visualisations of the analysis results are produced and printed on-screen. This is nice if you only analyse one animal, but as we included the analysis in a *for*-loop this output is not useful, so we turn it off.

Depending on the choices, the analysis will be faster or slower. It For larger datasets, frequency ranges, time series and significance simulations, running overnight will be necessary. Progress is shown in the R Console for each repetition of the *for*-loop.

Wavelet analysis does not often fail, but when it does this is inconvenient in a loop. In our example, the *tryCatch*() function is used to force the *for*-loop to continue even when the analysis of a pig has failed. To record whether this occurred, within the loop data frames are created that make notes of analyses that errored or produced a warning. The analyses of pigs with warnings or errors are skipped and are included 'wavelet.results' list as missing values.

Step 5 – Visualising and interpreting the results

WaveletComp provides the *wt.image*() function for creating visualisations of wavelet analysis results. This function produces a so-called wavelet spectrum, which visualises the strength of the rhythm at different frequencies and moments in the time series. The images can be personalised by using a range of settings, which can all be found in the documentation of the *wt.image*() function. In this example, we changed 1) the colours of the powers, using 'color.key', 'n.levels.' and 'maximum.level', and 2) the plot, axes and legend titles, using 'main', 'periodlab', 'timelab', 'spec.time.axis' and 'legend.params'.

Figure 7 provides the wavelet spectra of the performed analyses. The colours, ranging from dark blue to red, reflect the power (like in a heatmap) for a range of frequencies (i.e. y-axis, from 8-48 h, ‘Periods’) across the growing-finishing phase (i.e. x-axis, days after arrival at the farm). The closer the colour comes towards red, the stronger the repetition of that specific rhythm at that moment. The black line reflects the most likely rhythm detected. In these images, therefore, that means that there is a most likely rhythm detected for all pigs at approximately 24h, but the colours suggest that the strength (i.e. power) of this rhythm strongly differs between pigs. In addition, the white lines envelop periods during which the power was significantly stronger than white noise simulations. For some pigs, such as 7876, this hence means that they showed a strong circadian rhythm throughout their time in the barn, while the other pigs showed sequential periods with and without weak rhythms.

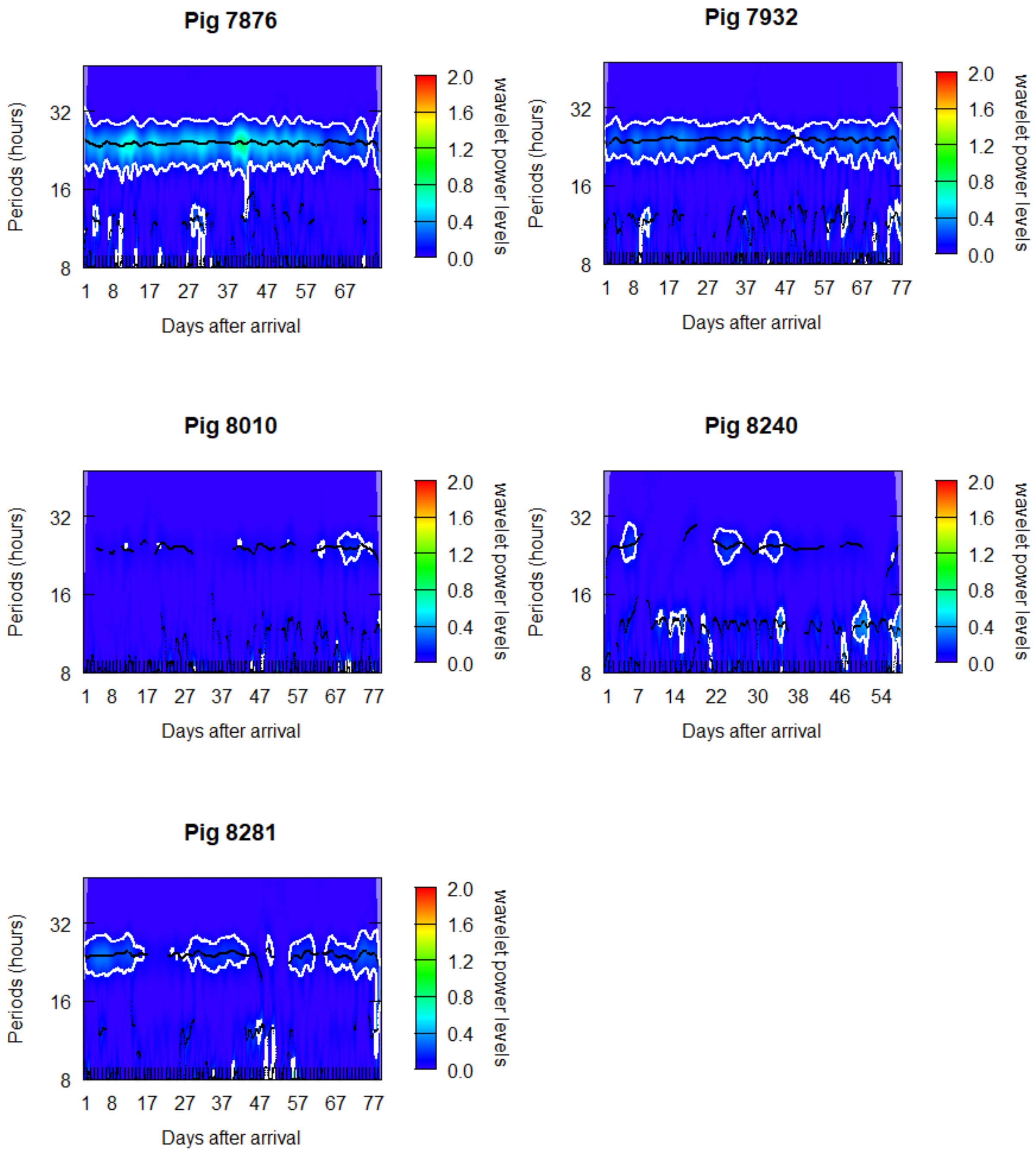


Figure 7. Wavelet spectra of individual pigs with white lines enveloping repetition detected as significantly stronger than white noise. Colours reflect the power of the repetition at a certain frequency (Periods) and certain moment of the growing-finishing phase (Days after arrival).

At the edges of the plot, quite small in this example, the colours fade out. These areas are called the 'cone of influence', and they indicate that the reliability of the power estimates for these locations in the plot is low. This is because daughter wavelets placed at the edges can be compared to less data. Power estimates within this cone of influence should therefore be interpreted with caution.

There were also some rhythms detected at around 12h, but these are a bit jumpy and generally not significant except for some specific, narrow locations in the time series. Most likely, these are artefacts created by the large number of zeros in the data, as described previously.

Step 6 (optional) – Extracting results for further study and visualisation

The visualisations of the power estimates and their significance are the main results of wavelet analysis. In some cases, it may be additionally desirable to obtain the quantitative results, for example for further analysis or to summarise results across individual animals. In our case, we quantified the number of days per pig in which a significant circadian rhythm was detected to compare this between pigs and months. In addition, we extracted the power values to use these as inputs for further analysis, such as to study whether pigs with weaker rhythms behaved differently than pigs with stronger rhythms, or whether pigs changed their rhythm strength during welfare issues. In the script, code is provided to extract 1) the power values at all frequencies and time points, and 2) the P-values that belong to each of these power values, followed by 3) on which days of the growing-finishing phase a circadian rhythm (with a bit of an error margin: 23.5-24.5 h) could be detected for individual pigs. Especially 3) is merely an example of what you can do with our dataset. The steps taken should be clear from the annotations in the script.

Wavelet spectra were previously created using the built-in function of the *WaveletComp* package. For *ggplot2* users, however, these may be inconvenient to combine with your other, *ggplot2*-based plots. The wavelet spectra can be recreated in *ggplot2* by extracting the relevant variables from 'wavelet.results'. Code to extract these variables and create *ggplot*'s out of them is also given in this step, after the quantifications.

Output

Scientific output

Bus, J.D., Boumans, I.J.M.M., Engel, J., Te Beest, D.E., Webb, L.E., Bokkers, E.A.M., 2023. Circadian rhythms and diurnal patterns in the feed intake behaviour of growing-finishing pigs. Scientific Reports 13, 16021. DOI: 10.1038/s41598-023-42612-1

References

Roesch, A., Schmidbauer, H., 2018. WaveletComp: Computational Wavelet Analysis, version 1.1. https://CRAN.R-project .org/package=WaveletComp

Torrence, C., Compo, G.P., 1998. A practical guide to wavelet analysis. Bulletin of the American Meteorological Society 79 (1):61-78. DOI: 10.1175/1520-0477(1998)079<0061:APGTWA>2.0.CO;2

Leise, T.L., 2017. Analysis of nonstationary time series for biological rhythms research . Journal of Biological Rhythms 32 (3):187-194. DOI: 10.1177/0748730417709105

Leise, T.L., 2015. Wavelet-based analysis of circadian behavioral rhythms. In: Methods in Enzymology, pages 95-119. Academic Press Inc. DOI: 10.1016/bs.mie.2014.10.011

Cazelles, B., Chavez, M., Berteaux, D., Ménard, F., Vik, J.O., Jenouvrier, S., Stenseth, N.C., 2008. Wavelet analysis of ecological time series. Oecologia 156 (2):287-304. DOI: 10.1007/s00442-008-0993-2