Tuco-Tuco Project Report

Bielefeld, 11.04.2021

Written by

Jasmin Matyssek Arne Kramer-Sunderbrink Sinan Harms
Sinan Kuscu Lukas Meinhold

University of Bielefeld

course
31-SW-StiP: 314000 Statistical Consulting

Contents

1 Introduction

Tuco-tucos are small subterranean rodents which belong to the family of the Ctenomyidae. They are mostly found in South America and spend most of their lives in underground cave systems. This makes it difficult for researchers to investigate their behavior in their natural habitat. Most research papers studying their behavior used laboratory data or data from tuco-tucos living in a small outdoor enclosure, such as Jannetti et al. (2019). In that named paper, the researchers were able to show that the tuco-tucos have a circadian rhythm, the downside is though that the data they had at hand was not from animals in the wild.

The research group around Jefferson Silvério and Milene Jannetti from an Argentinean-Brazilian laboratory of Chronobiology is mainly focused on studying the behaviour of this species and thanks to their effort they were able to provide us with the first data gathered from tuco-tucos living in the wild. In order to record the data, the animals were captured and equipped with an accelerometer that was attached via a collar. The device recorded data for several days before the animals were recaptured and the data was retrieved from the collar.

The aim of the following analysis is to identify whether tuco-tucos captured in the wild show a circadian rhythm. The focus will be on answering the question on how the data can be processed, especially downsampled, without losing its underlying structure and how many different behavioral states can be identified by a hidden Markov model such that it is still biologically plausible.

2 Data

The analyzed dataset contains triaxial accelerometer data, collected from four wild tuco-tucos, captured near Anillaco, La Rioja, Argentina. Overall the data contains 13,824,000 observations, sampled at a rate of 10Hz with a sensitivity of $4g$. This amounts to four days of data collected at different times in the year. Data from the days of capture and release was already discarded to avoid any side effects caused by disturbing the animals in their natural habitat, that is why all all datasets start and end exactly at 00:00:00.

Individual	Observation start	Observation end	# observations
JUL16	2019-07-10 00:00:00	2019-07-13 23:59:59	3,456,000
MAR01	2019-03-26 00:00:00	2019-03-29 23:59:59	3,456,000
MAR02	2019-03-26 00:00:00	2019-03-29 23:59:59	3,456,000
OCT09	2019-10-23 00:00:00	2019-10-26 23:59:59	3,456,000

Table 1: Observations per individuals, all times are in Argentinian time (UTC-3).

Using accelerometer data is an established method to assess energy expenditure. For a discussion of how accelerometer data and energy expenditure correlate see Meijer et al. (1989), Bouten et al. (1994) or Gleiss et al. (2011). But raw three dimensional data from these device is hard to analyse directly or even to just visualize, as you can see in figure 1.

Therefore tri-axial data is usually converted to a one dimensional value expressing *dynamic body acceleration* (DBA), using either *overall* DBA (ODBA, as proposed by Wilson et al. (2006)) or *vector* DBA (VeDBA). Both are calculated in a similar fashion:

$$ODBA = |A_x| + |A_y| + |A_z| \tag{1}$$

and:

$$VeDBA = \sqrt{A_x^2 + A_y^2 + A_z^2} \tag{2}$$

where A_x, A_y, A_z are the dynamic accelerations corrected for static acceleration by, for each channel, using the rolling mean over a given time window (2 seconds in our case) and subtracting it from the raw data (Qasem et al., 2012). Both measures only differ in which norm they apply to the three-dimensional acceleration vector. For our purposes we only considered the VeDBA, but it is to be expected that the ODBA would perform similarly. A detailed comparison can be found in Qasem et al. (2012).

Using the VeDBA, the acceleration data can be expressed as a regular time series of a single variable. The VeDBA values are non-negative real values, with a distribution skewed towards zero as can be seen in figure 1.

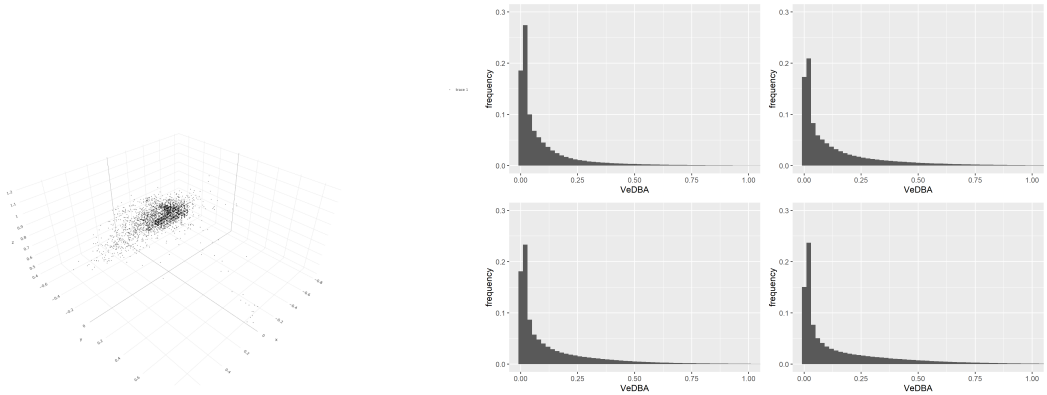

Figure 1: Left: Raw three-dimensional accelerometer data. Right: Histograms of the VeDBA data of the four animals (JUL16, MAR01, MAR02, OCT09 from left to right).

They exhibit a median of 0.05 and mean of 0.15, with the observed maximum being 4.51. There is a strong autocorrelation in the data, as is shown in figure 2 (right). In figure 2 (left) you can see that the animals show different levels of activity over the four days they were recorded with the animal JUL16 being the least active and OCT09 being the most active. In the following report, the colors used here will always be used to distinguish this four animals if not explicitly specified otherwise.

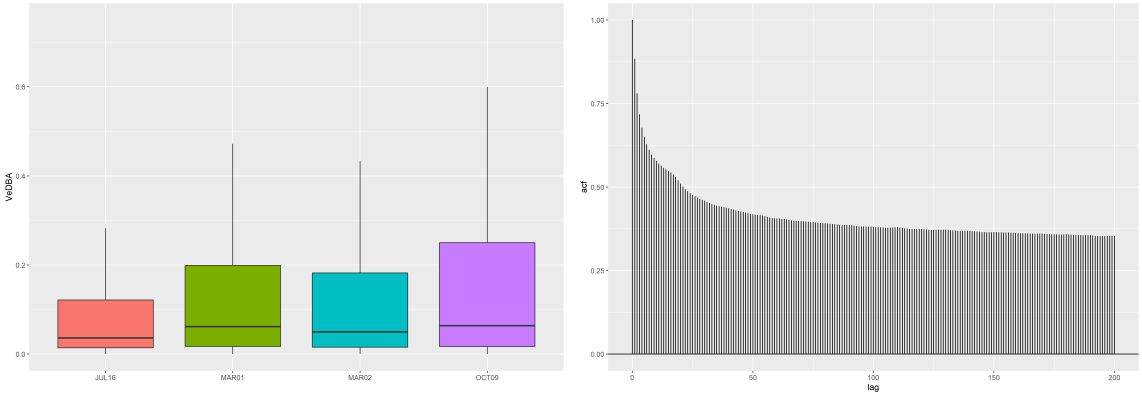

Figure 2: Left: Boxplot of VeDBA values for the different individuals. Right: Autocorrelation of VeDBA values.

3 Model

The behavior of tuco-tucos has already been studied and analysed in several other papers. For example, in Jannetti et al. (2019) the acceleration data of the tuco-tucos were measured as well, but they used a different method for modeling the behavior there. They categorise the activities using thresholds to differentiate between the types of activities. It was discovered that rhythmic behavior can be observed throughout the day.

In our analysis, we want to take not only the VeDBA values into account when classifying the time points in the data, but also their temporal structure. We assume that each animal switches between a finite number of behavioral states during the day (e.g. resting and foraging) with certain probabilities, and these behavioral states in turn determine the distribution of VeDBA values we observe.

Hidden Markov Models (HMMs) are a common tool to model this kind of temporal behavioral patterns. They belong to the dependent mixture models and the theory behind the models will be briefly explained in this chapter, which is largely based on Zucchini et al. (2016).

An HMM is a time series model and consists of two discrete-time stochastic processes. A discrete-time stochastic

process is a random process with countably many (equidistant) time points. The structure of a basic HMM is displayed in the graphic below.

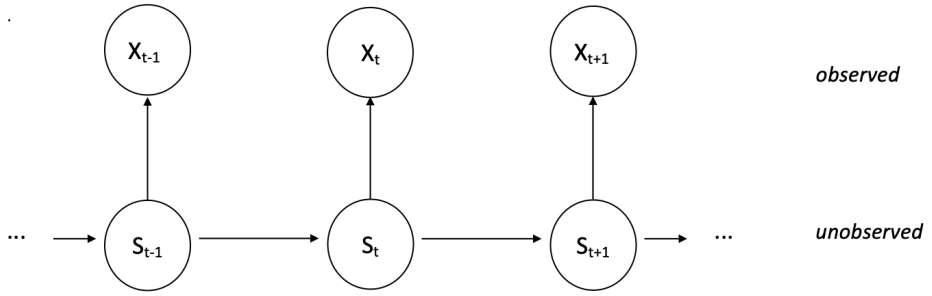

Figure 3: Example of an HMM.

In figure 3, the process $X_1, \ldots, X_T$ is the observed process and $S_1, \ldots, S_T$ the unobserved process. The indices denote the respective time point with $t \in \{1, \ldots, T\}$. The observed process, sometimes called the sequence of emissions, is an observed state-dependent process $X_1, \ldots, X_T$ and the hidden process is an unobservable state process $S_1, \ldots, S_T$ which values are in $\{1, \ldots, N\}$, i.e. the unobserved process has N different states. Each state has a different emission distributions. Here, X_t is the VeDBA data point at time t and S_t indicates which of the N emission distributions is active at the current time t.

An HMM must fulfil two assumptions which have already been hinted at above. The first assumption is the conditional independence assumption:

$$Pr(X_{t+1}|X_1, \ldots, X_t, S_1 \ldots, S_{t+1}) = Pr(X_{t+1}|S_{t+1}) \tag{3}$$

This implies that the distribution of X_{t+1} depends exclusively on state S_{t+1} the individual is in at that time point. The second assumption is the Markov property:

$$Pr(S_{t+1}|S_1, \ldots, S_t) = Pr(S_{t+1}|S_t) \tag{4}$$

The Markov property denotes that a state S_{t+1} depends only on the previous state S_t, hence the state process is completely described by the state transition probabilities. The probabilities of switching the state are the transition probabilities which are defined as $\gamma_{ij} = Pr(S_{t+1} = j|S_t = i)$ for all time points t. These γ_{ij} are the elements of the transition probability matrix

$$\mathbf{\Gamma} = \begin{pmatrix} \gamma_{11} & \cdots & \gamma_{1N} \\ \vdots & \ddots & \vdots \\ \gamma_{N1} & \cdots & \gamma_{NN} \end{pmatrix}. \tag{5}$$

The γ_{ij} denote the probability of changing from state i to state j with $i, j = 1, \ldots, N$. Since the elements γ_{ij} of the matrix Γ represent probabilities, they need to satisfy the constraints $\gamma_{ij} \in [0, 1]$ for $i, j = 1, \ldots, N$ and $\sum_{j=1}^{N} \gamma_{ij} = 1$ for $i = 1, \ldots, N$. Additionally, a start distribution must be specified to determine the distribution of the first state. This so-called initial state distribution is

$$\boldsymbol{\delta}_1 = (Pr(S_1 = 1), \ldots, Pr(S_1 = N)). \tag{6}$$

The distribution of states in an HMM quickly converges to the so called stationary distribution δ – the marginal distribution of states at a certain point in time if we know nothing about previous or subsequent states. This

distribution represents the expected share each state has in the complete state sequence. When fitting an HMM, one usually assumes that we start sampling from the middle of the sequence, not the start, and that we can therefore choose $\delta_1 = \delta$. This is convenient because δ can be calculated from Γ alone, we do not need to learn δ_1 from the data.

For estimating the transition matrix Γ and the emission distributions, we use the so called forward algorithm to calculate the likelihood in the first step. The closed-form expression of the likelihood is

$$\mathcal{L}(\theta) = \boldsymbol{\delta}\boldsymbol{P}(X_1)\boldsymbol{\Gamma}\boldsymbol{P}(X_2)\ \ldots\ \boldsymbol{\Gamma}\boldsymbol{P}(X_T)\mathbf{1}^t. \tag{7}$$

In the expression above, $\boldsymbol{\Gamma}$ is the transition probability matrix, $\boldsymbol{\delta}$ the initial distribution, $\mathbf{1}^t$ is an $N \times 1$-vector which is one in each component and

$$\mathbf{P}(X_t) = \begin{pmatrix} f_1(x_t) & & 0 \\ & \ddots & \\ 0 & & f_N(x_t) \end{pmatrix} \tag{8}$$

where the diagonal elements contain the likelihood of the emissions given the different states defined as $f_i(x_t) = Pr(X_t = x_t | S_t = i)$. In the second step, this likelihood has to be maximised and since there is no analytical solution, numerical methods are used which are already implemented in the statistical computing program R.

In the following analysis, a separate HMM is used for each of the four tuco-tucos, this approach is called "no-pooling". The question of why we decided to look at the individuals separate is briefly described below. Fitting a single HMM for all animals ("complete pooling") assumes that all behaviors of the individual animals are determined by the same parameter values and possible heterogeneity between the individuals is neglected. In this case, the difference would be omitted even though it exists. Since in the given data set only four individuals were observed with a very high frequency we have more than enough data to fit four different models. Also, we have no prior knowledge about the homogeneity of the behavior of the tuco-tucos, which could justify pooling. And even if their behavior was perfectly homogeneous, there could still be heterogeneity in the data due to the differents in the positions of the accelerometers attached to the tuco-tucos. Therefore we decided not to pool over the individuals.

4 Downsampling

While this is obviously a luxury problem, the huge amounts of data that the accelerometers provide (10 samples per second times 4 days makes approximately 3.5 million samples per animal) come with their own set of challenges. On the one hand, it becomes difficult to handle computationally: The fastest system available to us took one to two hours to fit a three state model to the full data of a single animal, which is not convenient and for bigger models with more states even completely infeasible (which is why we only used the three state models for our downsampling experiment described below). Using only every hundredth sample (10 seconds per sample, approximately 35 thousand samples per animal) reduces the training time to a manageable 40 seconds on average (see figure 4).

On the other hand, we are arguably not even interested in structure on that level of resolution: When biologists talk about the circadian rhythm of the animals they study, they are asking at what times the animals are active doing certain things and how long they spend with with these activities. They are not trying to achieve precision on the level of seconds or even tenths of seconds.

Still, it is not clear from the outset, that this resolution does not help with identifying the structure we are interested in. The computational effort could be worth it. In the following we will investigate this questions systematically: Should we downsample? And if so, what is the optimal downsampling factor? And should we simply throw away the data in between the samples or aggregate it using the mean or median function?

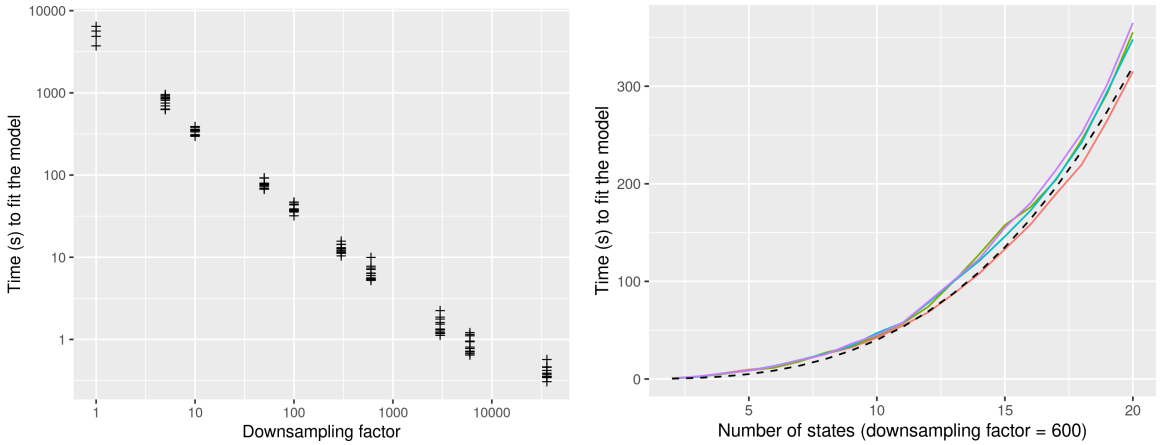

Figure 4: Time (in seconds) it took to train a three state HMM depending on the downsampling factor (left) and depending on the number of states of the model with downsampling factor fixed to 600 with cubic regression line (right).

4.1 Measures

In the following section we will investigate the downsampling question using a grid search, i.e. fit a model for every combination of 9 different downsampling factors, 3 downsampling methods (no aggregation, mean, and median), and the 4 animals plus 4 models fitted to the full resolution data of the 4 animal. This yields $9 \times 3 \times 4 + 4 = 112$ models. Unfortunately we cannot investigate this amount of models by looking at their state sequence predictions and comparing them individually as we would typically do in a model selection problem with a smaller amount of models. Hence, we need to devise a set of measures to quantify the desired properties we aim for in this modelling task.

There are many established methods to compare the quality of models. However, they all assume that the models to be compared were fitted to the same data. This assumption does not hold when we compare models fitted on downsampled data with different downsample factors and methods. For example, a bigger sample size tends to result in a smaller likelihood and hence a bigger AIC (see 4.1.1). Therefore, a new set of evaluation methods had to be developed for this project, or rather: methods used in other contexts had to be adapted for our task. The big challenge was to come up with measures that are not obviously biased towards bigger or smaller sample size as the AIC is. Note that these are by no means established methods for this task, as far as we are aware of, there are no established methods to compare models trained on different data. The whole endeavor is therefore somewhat experimental in nature and results have to be taken with a grain of salt.

The measures we utilized can be roughly divided into those that aim to measure how good the model fits the data and those that assess whether the model is suited to answer the research question at hand - to uncover the structure we are interested in.

4.1.1 Model fit

Normalized log-likelihood When we want to evaluate the fit of a model, the first thing we often think about is the likelihood and a simple way to normalize it would be to divide the logarithm of the likelihood by the number of samples. Assuming our model is able to perfectly capture the dependence structure of the data, i.e. the samples are conditionally independent given the model, this measure amounts to the average log-likelihood of the samples.

$$\frac{1}{T} \log f(x_1, \ldots, x_T | M) = \frac{1}{T} \log \prod_{t=1}^{T} f(x_t | M) = \frac{1}{T} \sum_{t=1}^{T} \log f(x_t | M) \tag{9}$$

However, in general the model will not be able to perfectly capture the dependence structure of the data and

even if this is the case, we can only compare the average-log likelihood if models were trained on data that has the same marginal distribution, which is not the case if we aggregate the data using the mean or median. Hence we will only look at this measure to compare models trained on downsampled data without aggregation and even then we won't be able to base strong conclusions on the results.

As mentioned above, the unnormalized log-likelihood and measures that depend on it are not fit to compare models trained with different sample sizes. However, we will use them later during model selection (section 5.2), and therefore briefly introduce them here. The Akaike information criterion (AIC) and the Bayesian information criterion (BIC) are the most common tools to decide between different models fitted to the same data. Both take the negative log-likelihood and add a term that penalizes the size of the model (the number k of free parameters of the model):

$$AIC = -2\log(f(X|M)) + 2k \tag{10}$$

$$BIC = -2\log f(X|M) + k\log T \tag{11}$$

Residuals Another classic way to assess the fit of a model is to analyze its residuals, or the pseudo residuals in the case of HMMs (see Zucchini et al., 2016, chapter 6.2).

To asses the *vertical* fit of the model (how well the emission distributions of the states capture the data at each time point), we analyse the marginal distribution of the residuals. If the we have a good vertical fit, we expect the residuals to be approximately normally distributed with mean 0 and standard deviation 1. We can measure the normality of an empirical distribution using the Kolmogorov–Smirnov statistic (KSS) which is defined as the maximal difference between the empirical cumulative distribution function (ECDF) of the data (here: the residuals) and the cumulative distribution function (CDF) of the target distribution (here: the standard normal distribution).

Unfortunately the KSS depends somewhat on the amount of data. Because the ECDF is a step function while the CDF of the normal distribution is smooth and approximately diagonal around zero, we expect the KSS to decrease with increasing sample size even if the data is perfectly normally distributed because for bigger amounts of data the steps of the ECDF will tend to get smaller (see figure 5). Therefore, we will complement our plots of the KSS of the residuals of our models with both the expected and minimal KSS we would get if the same amount of data were perfectly normally distributed (both computed via simulation).

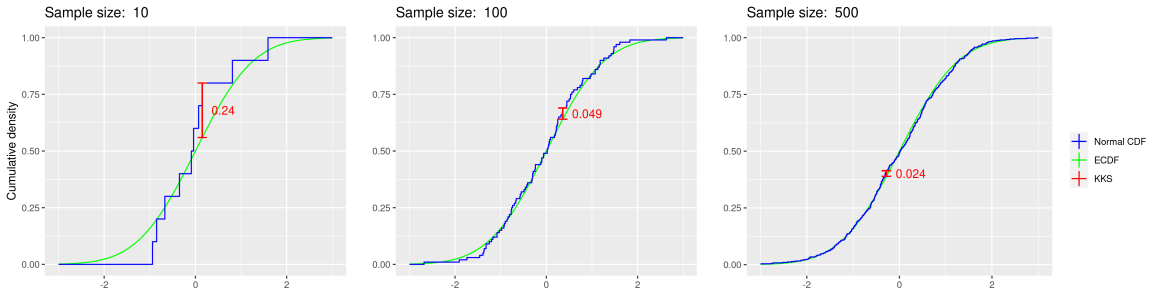

Figure 5: Kolmogorov–Smirnov statistic for different amounts of data sampled from a standard normal distribution.

To assess the *horizontal* fit (how well does the model capture the temporal dependency structure of the data) we take a look at the autocorrelation of the residuals, i.e. the correlation of the residuals with a copy of themselves delayed by one time step. If the HMM would capture the temporal dependence structure of the data perfectly we would expect to find no autocorrelation of the residuals, this is usually not achievable with HMMs, though. Since an HMM only models a limited number of states and assumes conditional independence of emissions given their state, it is not able to capture the local dependence of data within a single state one usually finds in high resolution data. Therefore we expect the autocorrelation to decrease as we reduce the resolution to a level where this local dependence is not in the data anymore.

Entropy A less common tool to measure the fit of an HMM is the entropy of the predicted state sequence given the model and the data.

$$H(S|X,M) = -\sum_{S\in\mathcal{S}} f(S|X,M)\log f(S|X,M) \tag{12}$$

Where $\mathcal{S}$ is set of all N^T possible state sequence, i.e. the set of all values the random variable S can take on, and $0\log 0$ is defined to equal 0. It can be interpreted as a measure of uncertainty of the predicted state sequence: If, given the model, there is only a single sequence of states that could have produced the observed data, i.e. the model is sure about it's predicted state sequence, the entropy will be zero. If, given the model, every possible sequence of states has the exact same probability, i.e. the model has no idea what state sequence could have produced the observed data, the entropy will be maximal (Hernando et al., 2005). If we divide the entropy by it's maximal value ($T\log N$) we get a normalized version, sometimes called the efficiency, that is independent of the number of samples in the sense that it's minimum possible values is always 0 and its maximum possible value is always 1 independent of T.

Here is a simple example to get a feeling for the measure: Assume that we trained a three state model ($N = 3$) that is 100% sure about 90% of the predicted states, but for 10% of the states it can only rule out one of the states, while the other two are equally likely. This yields $2^{0.1T}$ possible state sequences with equal probability $2^{-0.1T}$, hence

$$H(S|X,M) = -2^{0.1T}\times 2^{-0.1T}\log 2^{-0.1T} = 0.1T\log 2 \tag{13}$$

$$\frac{H(S|X,M)}{T\log 3} = 0.1\frac{\log 2}{\log 3} \approx 0.063 \tag{14}$$

Note that the normalized entropy is again independent of the sample size T. However, if we assume that uncertainties occur only at transition points between states and that there is a fixed number of, say, 20 transition points in the data independent of the resolution we get 2^{20} equally likely possible state sequences, hence

$$H(S|X,M) = -2^{20}\times 2^{-20}\log 2^{-20} = 20\log 2 \approx 13.86 \tag{15}$$

$$\frac{H(S|X,M)}{T\log 3} = \frac{20\log 2}{T\log 3} \approx 12.61/T \tag{16}$$

A realistic assumption will arguably lie somewhere between these examples: We expect the number of uncertain state assignments to grow slower than the number of certain state assignments as we increase the sample size, but we don't expect the uncertainty to converge to zero as fast as it does in the second example.

4.1.2 Relevance

The preceding methods all measure the fit of the model in some way, but to optimize the fit of the model is not the only thing we are aiming for when selecting a model (or in this special case: the data). Arguably, it is not even the most important aim in our case. More importantly, our results (here primarily the predicted state sequence) should be relevant for the research subject at hand: The tuco-tucos and their daily behavior.

We are not trying to select the data that fits any specific hypothesis about the circadian rhythm of the animals but we do assume that there is some structure to their behavior, that they don't select some action completely at random at every moment of the day, and we try to select the downsampling factor that allows us to see that structure most clearly.

State changes In a perfect world, the animals would switch between clearly separated behavioral states a finite number of times throughout the day and our model would be able to identify these time points. Hence, as long as the resolution does not get so coarse, meaning that we summarize a time interval with more than

one state change to a single sample, the number of state changes in the predicted state sequence should be constant. Obviously, we do not live in a perfect world – The behavioral states are not clearly separated and our predictions of them will be noisy and hence we expect the number of state changes to grow with the sample size.

When comparing two models trained on the same amount of data, a smaller number of state changes could be a sign that the model was more successful in distilling the real behavioral states from the noisy data. A state sequence with a smaller amount of state changes is also easier to read by the naked eye without any further aggregation methods (e.g. state density plots like in figure 27), as figure 6 demonstrates.

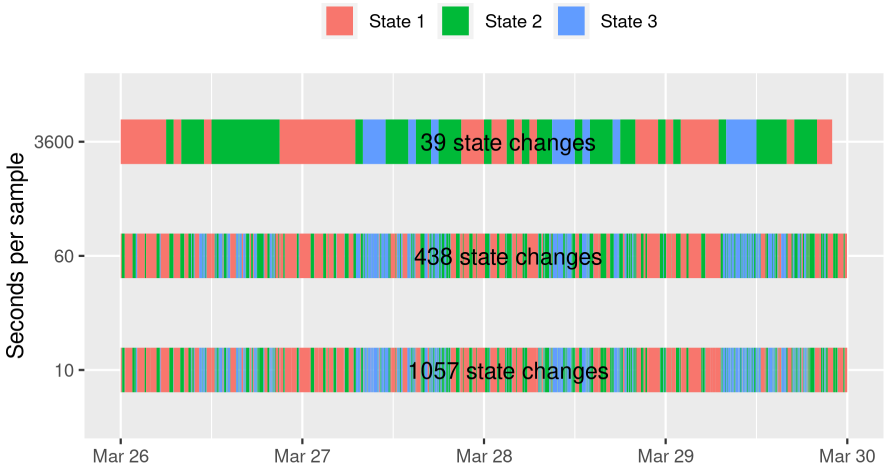

Figure 6: Predicted state sequence of models fitted to data from the same animal with different resolutions.

Temporal dissimilarity of states In figure 7 we see the emission distributions of a three state model fitted to data with a resolution of 30 seconds per sample. Clearly, state 3 is the most active state, but the role of state 1 and 2 are not immediately clear. If we look at the plot of the data with data points colored according to the predicted state sequence, we see that state 2 covers the bulk of the intervals of lower activity and state 3 merely *mops up* extreme values (very close to zero and bigger than 0.1) during that intervals. There seems to be no temporal structure to it, state 2 and 3 merely divide the intervals of low activity *horizontally* among each other. This gives the model a better fit than if it had to cover all these values with a single gamma distribution, but for our research question, this horizontal distribution has no relevance, it only adds unnecessary complexity to the predicted state sequence: The fitted states 2 and 3 arguably represent only a single behavioral state. The fact that this behavioral state can be fitted better by two gamma distributions than a single one is of little interest to the circadian rhythm of the tuco-tucos.

Here is how can we diagnose this defect without taking a close look at the colored scatterplot of every single one of 112 fitted models: We estimate the distribution of each state over time from the predicted states using kernel density estimation with an Epanechnikov kernel with a bandwidth (standard deviation of the kernel) of ten minutes. Then, we compute the *dissimilarity index* for each pair of distributions. This measure equals 1 if there is no overlap at all between the estimated temporal distributions of states i and j, and 0 if they share the exact same distributions:

$$\frac{1}{2}\int_{-\infty}^{\infty} |\hat{f}_i(t) - \hat{f}_j(t)|\, dt \tag{17}$$

Figure 8 visually demonstrates how this works: In each plot, we see the temporal distribution of two states (for clarity only a single day is plotted). The shaded area between these distributions (divided by two) is the dissimilarity index of the two states. We can see how our observation from figure 7 directly results in the low dissimilarity index of states 2 and 3: The states occupy the same intervals without clear temporal structure

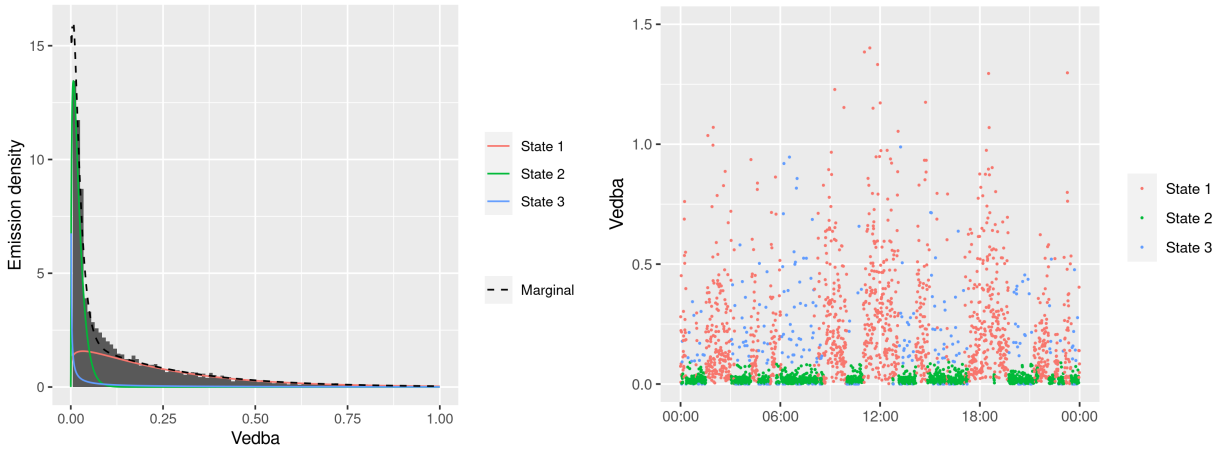

Figure 7: Example for a model with three states that are not temporally clearly separated but divide the same time intervals horizontally among each other.

within that intervals, hence their temporal distributions look very similar and and hence their low dissimilarity index.

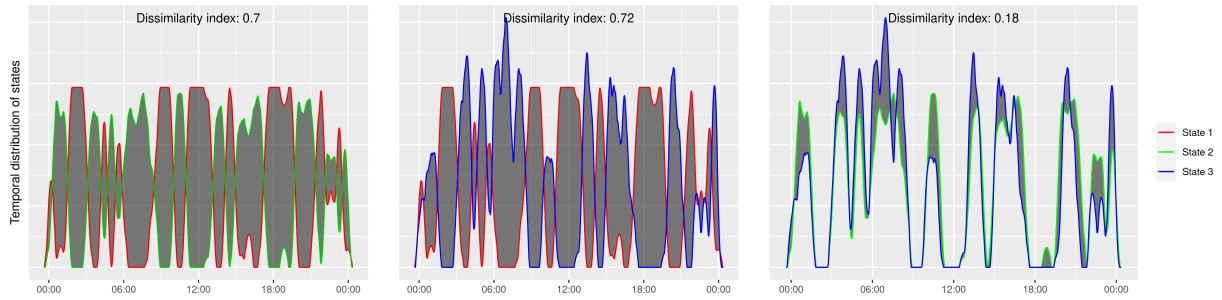

Figure 8: Difference between temporal density plots of the states from the model described in figure 7.

Two states that are distinct but not clearly separated with noisy transition periods would result in a medium dissimilarity index. Even when the states are perfectly separated, we won't get a perfect dissimilarity index of 1 – Due to the smoothing introduced by the density estimation, there will always be some overlap. An exception is the most extreme downsampling case where we only have a single data point per hour. For an Epanechnikov kernel with a standard deviation of ten minutes the width of the kernel window (its support) is $10 \times 2 \times \sqrt{5} \approx 45$ minutes. Hence for that resolution the kernel window will be completely covered by a single data point, there is no smoothing taking place and the dissimilarity index will always be exactly 1.

4.2 Evaluation

In the last section we introduced a number of measures to compare both the fit of models trained on different amounts of data and their relevance for studying the circadian rhythm of the tuco-tucos. We have seen that the interpretation of these values is not always straight forward, even for those measures that look independent of the sample size on first sight, it can still have indirect effects. Hence, we need to proceed with caution when we are now going to look at the results of our experiment and try to make inferences about the best downsampling factor and method for our data.

First, we need to describe our method more clearly: For every downsampling factor d we summarize every complete block of d samples to a single sample by either taking the mean or median of all samples in the block or by simply using the first sample and discarding the rest without aggregation. We have chosen values for d from an approximately logarithmic scale to get a wide range of different factors (see table 2). Therefore, the x-axis

of the plots that compare models with different downsampling factors will always be scaled logarithmically.

For the four animal data sets, preprocessed in that way, we fit an HMM with three states with gamma emission distributions. We don't do any pooling. Besides the reasons mentioned in section 3, having four independent models for every downsampling factor and method allows us to asses if the trends we observe are robust among different data sets.

Downsampling factor	1	5	10	50	100	300	600	3000	6000	36000
Time per sample	0.1s	0.5s	1s	5s	10s	30s	1min	5min	10min	1h
Samples per time	$10\frac{1}{\text{s}}$	$2\frac{1}{\text{s}}$	$1\frac{1}{\text{s}}$	$12\frac{1}{\text{min}}$	$6\frac{1}{\text{min}}$	$2\frac{1}{\text{min}}$	$1\frac{1}{\text{min}}$	$12\frac{1}{\text{h}}$	$6\frac{1}{\text{h}}$	$1\frac{1}{\text{h}}$

Table 2: Downsampling factors used in our experiment and translations into a "human readable" format.

4.2.1 Downsampling without aggregation

We will start by comparing different downsampling factors for the simple downsampling method without aggregation.

As expected, the number of state changes does not stay constant but decreases as the downsampling factor increases. This behavior can be described by the (double logarithmic) regression line defined by $54000d^{-0.73}$, i.e. if we reduce the sample size by a factor of 10, we reduce the number of state changes approximately by a factor of $10^{0.73} \approx 5$. A simple explanation is that, as we reduce the sample size, we reduce the opportunity for noise in the state sequence. This does not necessarily mean that the model learned a clearer temporal structure, it does mean that plots of the model will be cleaner and easier to read with a naked eye, though.

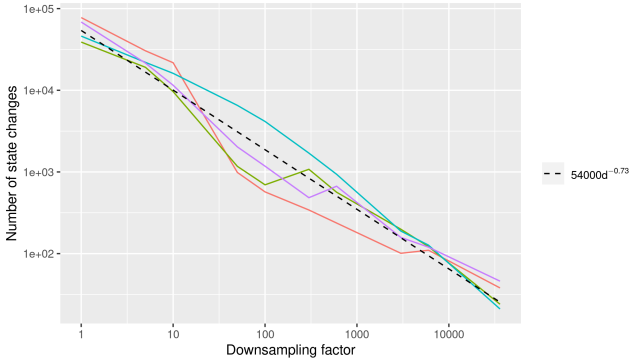

Figure 9: Number of state changes as a function of the downsampling factor, grouped and colored by individual, with regression line.

The plot of the dissimilarity indices (figure 10) is unfortunately quite convoluted. Here are some observations: For all downsampling factors, but the two highest ones, the temporal distribution of the state with the biggest share of the state sequence (state 1) is relatively dissimilar from that of states 2 and 3 while state 2 and 3 have similar temporal distributions. This is because the model typically uses state 1 as a resting state and states 2 and 3 for medium and high activity which happen at similar times during the day. Sometimes these roles are changed which causes the dissimilarity of some pairs of states to go up while that of other pairs goes downs (see the animal corresponding to the red lines at $d = 10$ for example). To get a clearer picture of the overall dissimilarity, we computed the average of the three dissimilarity indices weighted by the share of their corresponding states of the state sequence:

$$\frac{1}{2}\sum_{i=1}^{3}\sum_{j=i+1}^{3}(\delta_i + \delta_j) \times \text{dissimilarity index of } (i,j) \tag{18}$$

This gives a much cleaner picture, showing that the overall temporal separation of the states increases as we increase the downsampling factor. Only some of this can be attributed to the model learning a clearer temporal

structure, though. As we explained in section 4.1.2, the dissimilarity index necessarily goes to 1 as the time per sample approaches 45 minutes. Never the less, both, the number of state changes and the dissimilarity indices, consistently indicate that the temporal structure of the state sequence gets clearer as we increase the downsampling factor.

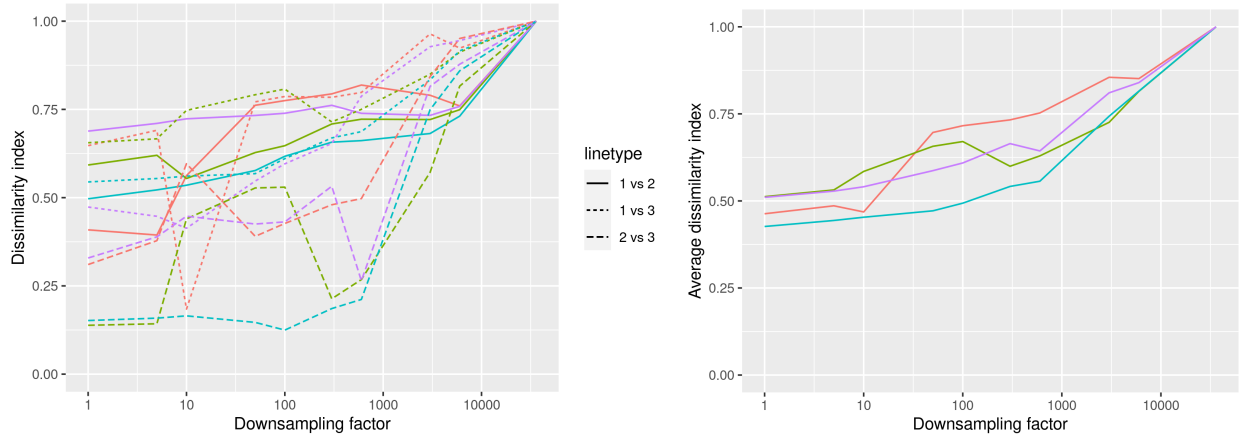

Figure 10: Dissimilarity index of pairs of states ordered by their share of the state sequence (δ) (left) and as an average of all pairs weighted by δ (right), colored by individual.

On the other hand, we are throwing away huge amounts of data, 99.9% for a downsampling factor of 1000, and at some point the model won't be able to reliably identify any structure in the data. We can see this in the normalized log-likelihood decreasing as we increase the downsampling factor (figure 11).

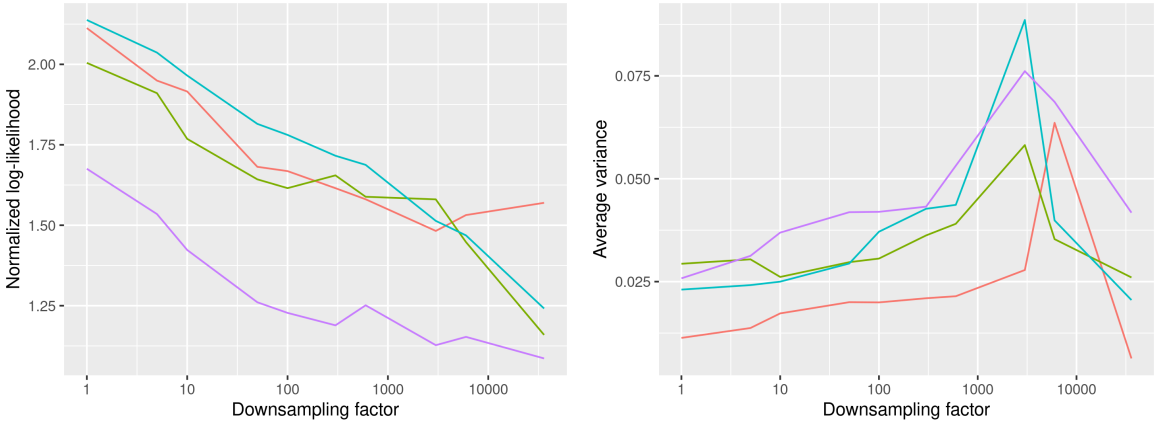

Figure 11: Left: Normalized log-likelihood as a function of the downsampling factor. Right: Average variance of emission distributions weighted by δ.

The analysis of the pseudo residuals (figure 12) does not match the results of the normalized log-likelihood: We see that, as we increase the downsampling factor, the KSS gets closer to the expected KSS of normally distributed data of the same sample size (the dashed line) – The residuals become more normal, indicating that the model becomes better at fitting the data "vertically". This contradiction with the log-likelihood can possibly be resolved by taking into account that, as we increase the downsampling factor, the average variance of the three emission distributions (weighted by δ) increases (see figure 11 right): A more spread out distribution will yield a smaller likelihood even though it's shape fits the data better. In the most extreme cases, where the variance is reduced again due to the small sample size, even though the likelihood keeps decreasing, the KSS is not a reliable measure of normality due to the small sample size. Despite this explanation attempt we need to admit that we don't have a clear result about the effect the downsampling factor has on the vertical fit of the model.

Fortunately, the autocorrelation of the residuals allows for a nice explanation: When we have VeDBA data of a single individual at an extreme resolution like 10 data points per second the autocorrelation of the data will be very high. As we see in figure 12, our model is not particularly good at learning this dependency on the micro level. The fact that the autocorrelation is closest to zero when the time between two consecutive data points is 10 to 60 seconds suggests that there is a temporal structure at that resolution the HMM is able to learn. If we increase the downsampling factor further, the temporal fit of the model gets worse again. It is hard to say whether this is significant due to the low sample size, though.

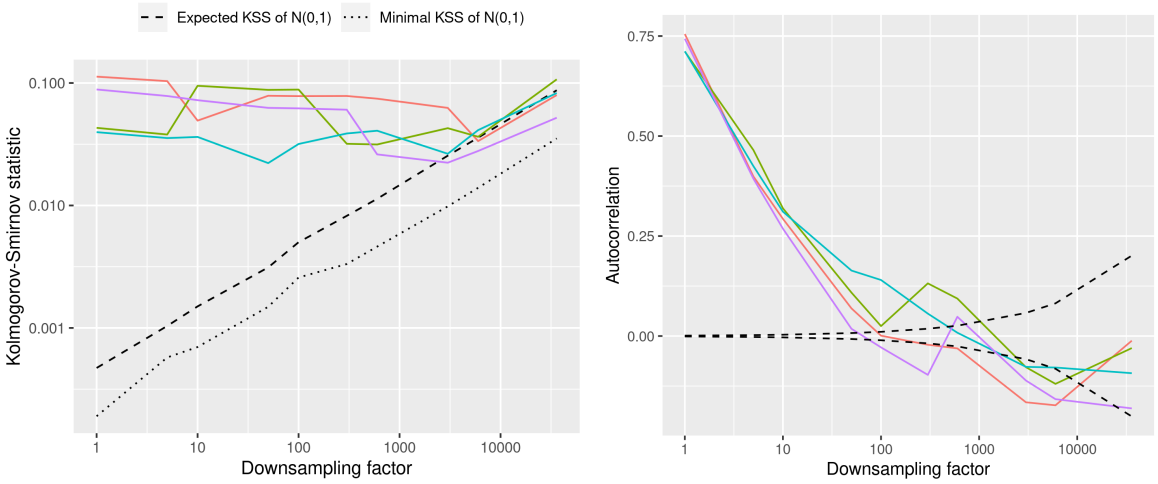

Figure 12: Left: Kolmogorov-Smirnov statistic of pseudo residuals, compared to the expected and minimal Kolmogorov-Smirnov statistic of standard normally distributed data. Right: Autocorrelation of residuals with lag 1. The black dashed lines mark the 95% confidence interval for the autocorrelation of noise of that sample size, values outside this interval signal statistically significant autocorrelation.

The third measure of fit we introduced in the last section is the (normalized) entropy of the state sequence given the data and the model, which can be interpreted as a measure of uncertainty about the models predicted state sequence. Remember that if there is a learnable temporal structure to the data, we would expect that the amount of time points that are not clearly identifiable would grow slower than the amount of clearly identifiable time points if we increase the sample size, hence we would expect the normalized entropy to rise as we increase the downsampling factor. In light of this knowledge, it would actually be more worrying if the entropy would turn out to be constant with respect to the downsampling factors in figure 13.

There is more structure to the plot however: The fact that the entropy decreases again for the biggest downsampling factor is likely due to the decreasing variance of the states we have seen in figure 11 – States that produce a smaller range of emissions are easier to identify. The local optimum around 5 to 10 seconds per data point could possibly support the hypothesis raised in connection with the similar local optimum of the autocorrelation, that there is a sweet spot where micro level correlation of the data a HMM is not able to learn is removed by the downsampling, but there is still enough data to learn the lower resolution temporal structure.

To summarize the result of our analysis of models trained on data downsampled without aggregation: The higher the downsampling factor, the easier it is to interpret the model with our research question in mind and up to a certain point we are able to fit a model decently well on downsampled data, arguably even better than on the full resolution data and certainly much faster. If we reduce the number of samples further, we loose to much data to get a good fit, though.

We could decide for a sweet spot somewhere between 5 and 30 seconds per sample... or maybe we can reduce the sample size further without the drawbacks identified above if we don't throw away the data between the samples but aggregate the data in each interval using the mean or median function.

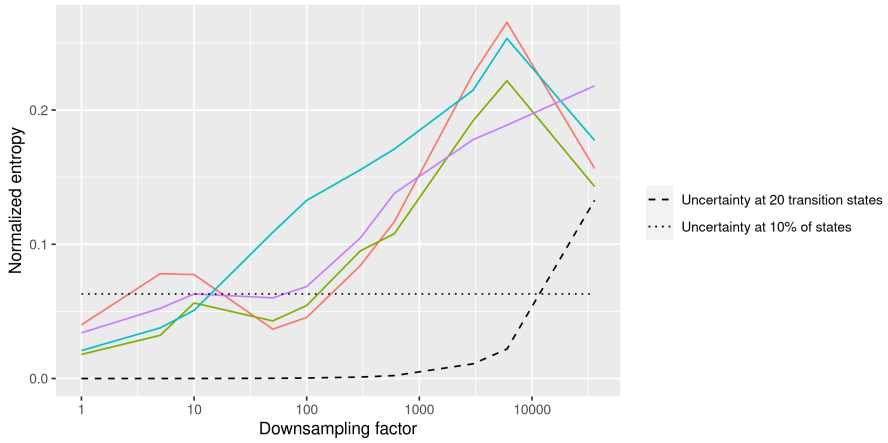

Figure 13: Normalized entropy of the state sequence with the normalized entropy of the examples described in equations 14 and 16 for reference.

4.2.2 Downsampling with aggregation

In figure 14 we plotted two characteristics of the fitted models as we increase the downsampling factor: The means of the emission distributions of the three states and the stationary distribution δ of the states. Note that, if we fitted a model to the full resolution data and assumed that we found the true data generating model, we could simply adjust the model to the downsampled data by raising its transition matrix Γ to the power of the downsampling factor. This would leave both the means of the emission distribution and δ unchanged.

As we can see in the first column of figure 14, the models trained on the downsampled data are not just the same model adjusted for the bigger time difference between samples but very different models. That does not necessarily mean that the models contradict each other in a sense that is relevant to our research question: As we see in figure 6, HMMs sometimes simply use their states differently to describe essentially the same structure. In figure 6 we see three models describing roughly the same circadian rhythm (low activity in the night and in the morning, high activity in the afternoon and evening), yet, because the low resolution model utilizes state 3 more sparsely for very high activity, it will have very different emission means and a different stationary distribution. Nevertheless, the increased oscillation of the values for very high downsampling factors in figure 14 is a clear sign that throwing away 99.9% of the data (unsurprisingly) makes the original structure of the data unidentifiable.

We see in the second and third column of figure 6 that aggregating this data to a mean or median value yield models which are very robust for a wide range of downsampling factors. Note that aggregating the data using the mean or median removes extreme values from the data. The fact that the emission means get closer together as we increase the downsampling factor is therefore most likely just due to the model adjusting to the smaller range of the downsampled data and not changing in any relevant way.

In figure 15 we can see that the number of state changes as well as the average dissimilarity improve for higher downsampling factors just as we have seen in the last section for non aggregated data. So aggregation yields the same benefits as we increase the downsampling factors, but does it help with the fit of the model?

While using the normalized log-likelihood to compare model fit was at least dubious in the non aggregation case, we really cannot justify to use the likelihood in this case because the mean and median function actually change the distribution of the data.

The plots of the KSS (figure 16) have generally the same shape as we seen for models trained on data without aggregation. The values for mean aggregation are more concentrated around the lower range of values for no aggregation, the values for median aggregation are more concentrated at the upper range for moderate downsampling factors. The mean method seems to produce data that is easier to fit with the gamma emission distributions.[1]

[1]Note that the mean of independent gamma distributed variables is again gamma distributed, while the median is not. Since

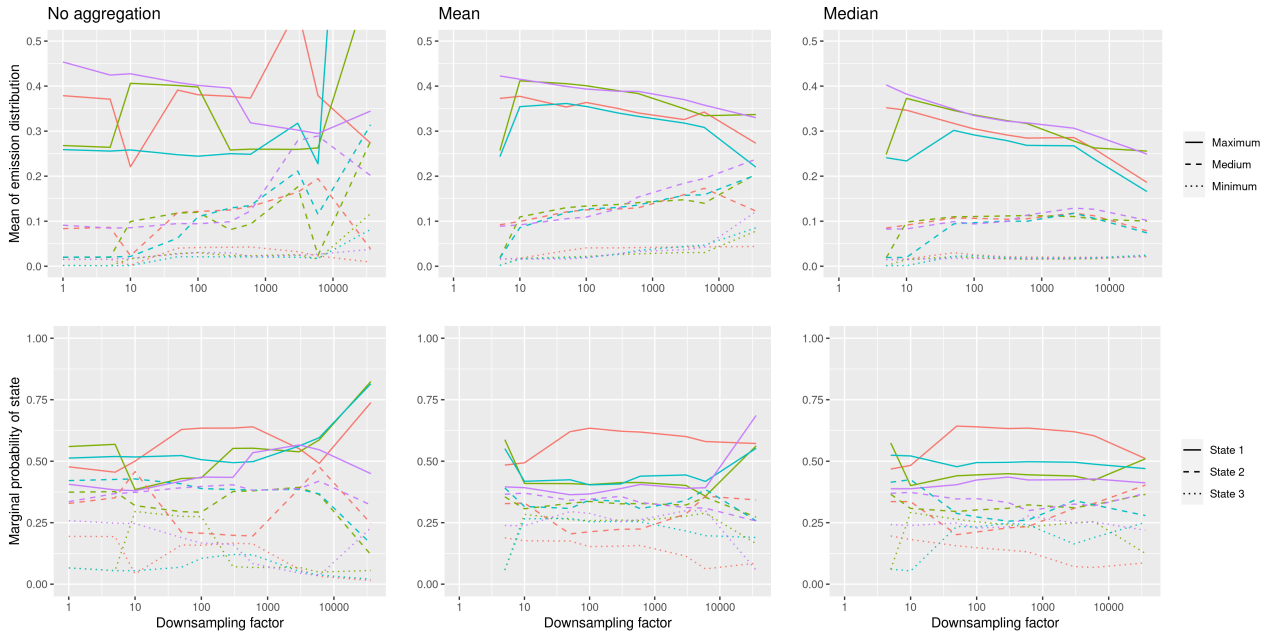

Figure 14: First row: Biggest, middle and smallest mean values of the emission distributions of the states. Second row: Marginal probability of the states δ.

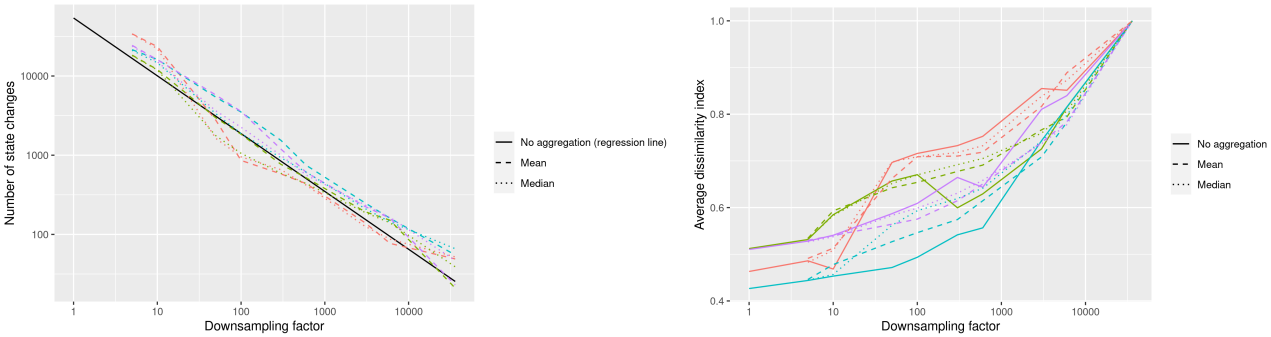

Figure 15: Left: State changes for methods trained on aggregated data with the regression line from figure 9 for reference. Right: Average dissimilarity index of all pairs weighted by their combined marginal probability.

For a downsampling factor up to 50 the autocorrelation of the residuals is consistently better for the models without aggregation. This is possibly due to the downsampling removing dependencies of samples on the micro level that the HMM does not like to learn while the aggregation methods preserve some of these. For higher downsampling factors the mean aggregation results are very similar to the no aggregation results, the local optimum of autocorrelation around zero is extended to downsampling factors of 6000. The median aggregation yields autocorrelation values that are slightly higher than that of the other methods.

With the normalized entropy of the state sequence (figure 17) we finally see a clear advantage of the aggregation, allowing the local optimum of the no aggregation results at a downsampling factor of 60 to extend to much higher values – For factors of 300 to 6000 the model is clearly more confident about its predicted state sequence. Interestingly, if we add up the reference entropies from equation 14 and 16, we get a pretty good fit of the entropy of the models with aggregation, i.e. the model is as sure about it's predictions as a model that is undecided between 2 of 3 states for 10% of the time points and a constant 20 additional time points.

the aggregated values are clearly not independent this fact can only hint at an explanation of the better performance of the mean method with respect to the KSS, though.

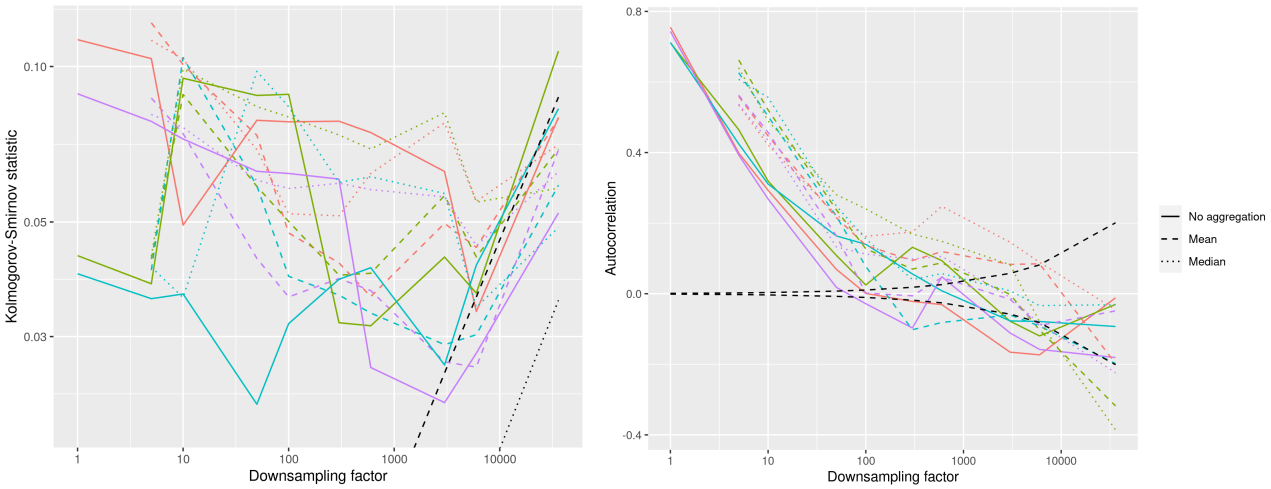

Figure 16: Left: Kolmogorov-Smirnov statistic of pseudo residuals for different methods, expected and minimal Kolmogorov-Smirnov statistic of standard normally distributed data from figure 12 in black for reference. Right: Autocorrelation of residuals with lag 1 for different methods. The black dashed lines mark the 95% confidence interval for the autocorrelation of noise of that sample size, values outside this interval signal statistically significant autocorrelation.

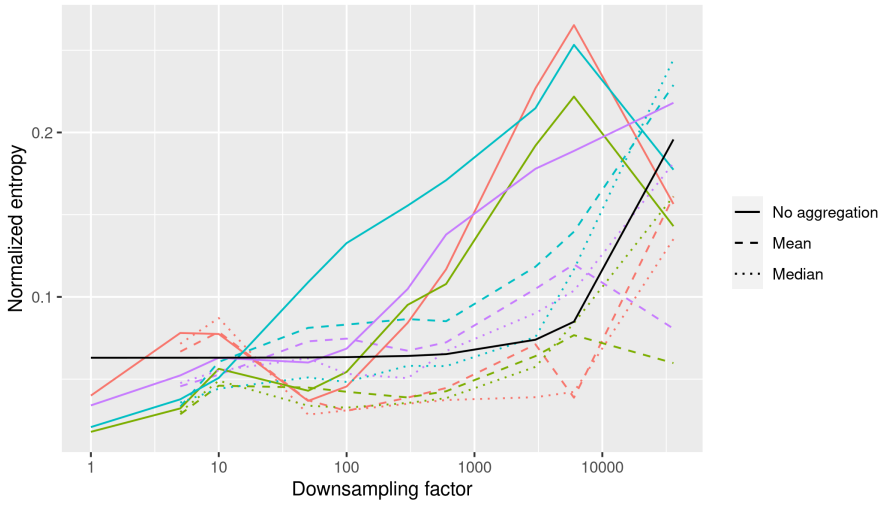

Figure 17: Normalized entropy of the state sequence with the sum of the reference entropies from figure 13 for reference in black.

4.3 Summary of the results

The most important takeaway from this chapter should be: Systematically evaluating preprocessing methods in an unsupervised context (i.e. where there is no ground truth we can compare our results to) is very hard. There are no established methods and constructing a measure that is not in some way biased towards lower or higher sample sizes seems impossible. Therefore, results cannot be used for simple decision rules (like the AIC) but have to be interpreted carefully. In the last section we tried to do that to the best of our abilities, but in the end, we have to reduce all this complexity to a single decision: What is the best downsampling factor for our purpose, and what is the best downsampling method.

The extremes can be ruled out: Full resolution data is just impractical, waiting 2 hours for a three state model to fit is bad enough but since computation time is cubic with respect to the number of states (see figure 4), fitting bigger models becomes completely infeasible. And we are not interested in structures on that

micro level nor would a "vanilla" HMM be able to learn it anyway (see figure 16).[2] Extreme downsampling on the other hand does not produce data well suited for HMMs either. Even using aggregation methods, especially the normalized entropy becomes unacceptably high for high downsampling factors. A sweet spot between convenient computation times and readability on the one hand and model fit and confidence about the predicted state sequence on the other could be chosen at one minute per sample (downsampling factor 600).

Aggregation using mean and median yields similar results. The states seem to be easier to separate in the data produced by median aggregation judging from the slightly lower number of state changes and the slightly higher confidence about the predicted state sequence. The data produced by the mean aggregation seems to be easier to fit judging from the better residual values.

Figure 18 compares the different downsampling methods using a single day of one of the animals as an example. Having seen how similar the HMMs perform on data downsampled using the different methods it is surprising that the downsampled data looks so different to the naked eye. The aggregation using the median vertically separates three clusters of VeDBA values much more clearly and the model fits three corresponding emission distributions with much less overlap. While on first sight this looks like the HMM trained on the median data should have a clear advantage over the HMM trained on the mean data, all three models classify the time points almost identically. Still, while HMMs are obviously capable to identify structures in messy data where the human eye struggles to see it, it is nice to be able to understand the classification decisions of the model when looking at the data, which is why we favour the median as a downsampling method.

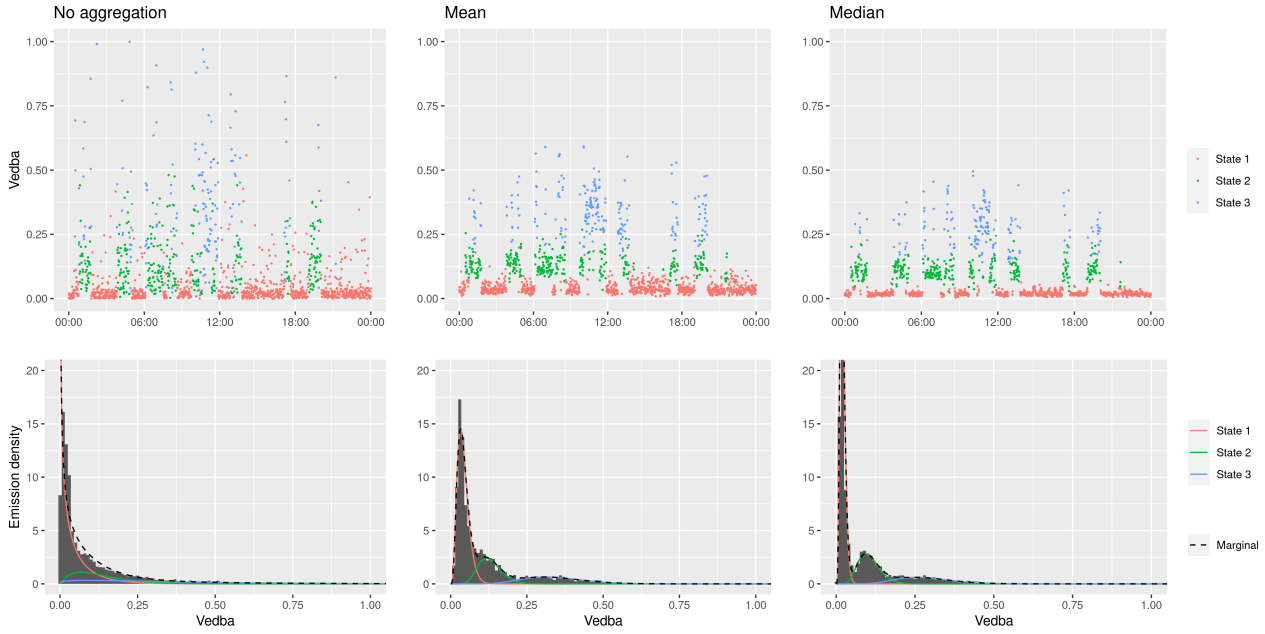

Figure 18: First row: Data from one day of one animal downsampled to one sample per minute using different methods. Second row: Emission distributions of the states fitted on the data with histogram of the data.

5 Model Selection

5.1 Do we need to model temporal dependence?

HMMs are not the simplest tool that allows us to cluster activity values. Jannetti et al. (2019) simply chose two thresholds to cluster data into three different states: Times where the activity measure (ODBA in their case, VeDBA for us) is lower than the mean are considered episodes of low body movement, times where the activity measure is higher than one half of the maximal value are considered episodes of high activity. If we prefer to have an algorithm choose these threshold for us automatically, we can use the k-means algorithm with $k = 3$.

[2] If one really wants to model the direct dependency of close data points, one could try to fit a Markov-switching model that does not assume conditional independence of observed values (Zucchini et al., 2016, p. 150). For our purpose that is not necessary.

Both methods yield clustering that look very similar to the one we get by fitting an HMM with three states on first sight and demand very little effort in comparison to fitting an HMM. The advantage of an HMM model is that it is able to take the temporal structure of the state sequence into account: The classification of a time point does not only depend on its VeDBA measure but also on the classification of neighboring time points. This allows HMMs to distinguish states even if their marginal distributions overlap.

Figure 19 demonstrates the problem with methods that base their classification on "vertical" information only: from 4:30 to 5:00 we have a period of activity that is visually clearly separated from the low activity before and after. However, because the lowest values in this period of activity are lower than some of the values during resting the threshold methods necessarily cut this period of activity vertically in half which in turn leads to constant state changes during that activity period and a weaker temporal separation of the states.

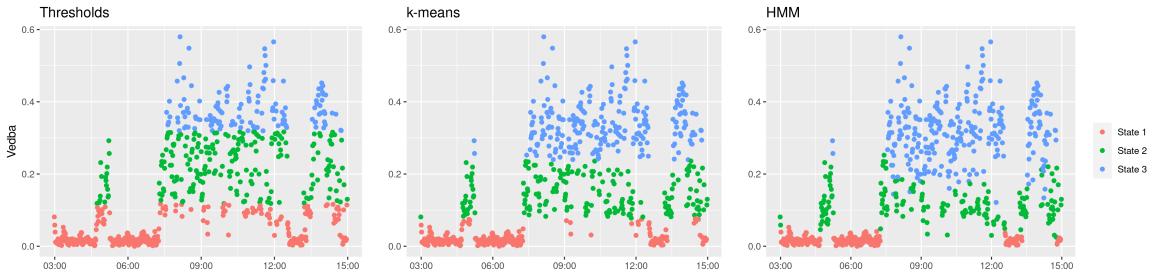

Figure 19: 12 hour state sequence as clustered via thresholds à la Jannetti et al. (2019), k-means, and HMM.

This observation can be confirmed using the quantitative measures introduced in the last chapter: The state sequence predicted by the HMM exhibits a much smaller number of state changes and bigger dissimilarity indices (see table 3). The fact the manual thresholds seem to perform so good on the JUL16 data can be explained by the small number of time points that get classified as state 3 (4% vs 9% for k-means and 13% for the HMM).

Method	Thresholds			
Animal	MAR01	MAR02	JUL16	OCT09
Nr. state changes	1226	1278	575	1370
Avg. dissimilarity	0.64	0.60	0.75	0.59
Dissimilarity 1 vs. 2	0.58	0.56	0.74	0.62
Dissimilarity 1 vs. 3	0.78	0.72	0.86	0.72
Dissimilarity 2 vs. 3	0.46	0.44	0.53	0.29
Method	k-means			
Animal	MAR01	MAR02	JUL16	OCT09
Nr. state changes	971	1188	709	1252
Avg. dissimilarity	0.67	0.61	0.72	0.60
Dissimilarity 1 vs. 2	0.63	0.60	0.75	0.59
Dissimilarity 1 vs. 3	0.81	0.70	0.81	0.72
Dissimilarity 2 vs. 3	0.54	0.47	0.45	0.36
Method	HMM			
Animal	MAR01	MAR02	JUL16	OCT09
Nr. state changes	438	611	418	561
Avg. dissimilarity	0.70	0.64	0.73	0.65
Dissimilarity 1 vs. 2	0.65	0.60	0.76	0.59
Dissimilarity 1 vs. 3	0.84	0.73	0.82	0.82
Dissimilarity 2 vs. 3	0.59	0.55	0.45	0.52

Table 3: Quantitative measures of different clustering methods: The number of state changes in the predicted state sequence, the dissimilarity indices, and their average weighted by the share of the corresponding states of the state sequence.

Due to the fact that aggregation using the median, as we have seen in the last chapter, produces data that has a clear "vertical structure", one could actually justify using the simpler methods for this specific data set with this specific preprocessing for separating the data into up to two or maybe even three clusters. However, if we want more clusters, there will inevitably be more overlap between the states and separating them will become impossible without taking temporal dependencies into account. In the next section we will investigate, whether a higher number of behavioral states can be identified in the data.

5.2 Number of States

When dealing with HMMs, we also have to ask the question how many states we should consider, when fitting the model. In the previous section we talked about using 3 states. This decision can be based upon looking at the data and fitting a baseline model, but there should also be some sort of method-based justification on why we chose a certain number of states.

In figure 20 we can see a comparison of the fit of our model measured by three different metrics, namely the normalized log-likelihood, the AIC, and the BIC (see section 4.1.1). Unsurprisingly, we can observe that adding more states, therefore more free parameters of the model, leads to a better fit to the data, therefor an increase in likelihood and a decrease in the AIC and BIC score. When looking at the plots we can see that according the AIC we should choose a model with 7-9 states, while the BIC prefers 6-8.

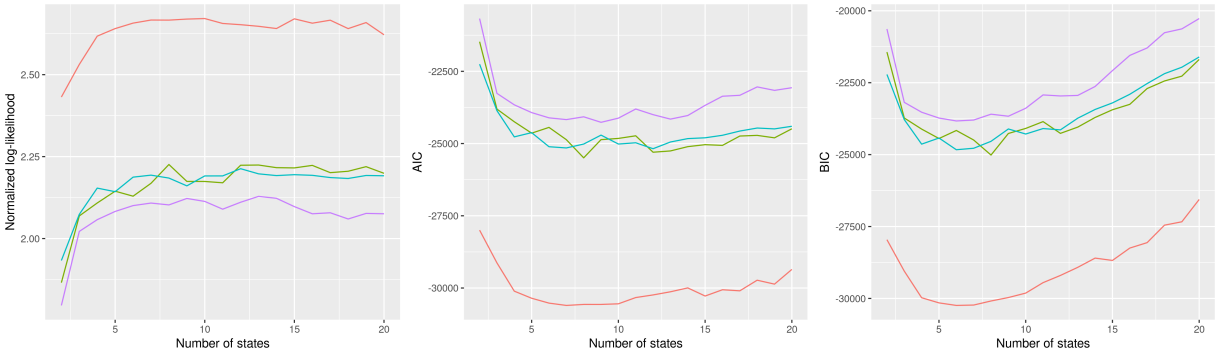

Figure 20: Fit of the model as measured by the normalized log-likelihood, the AIC, and the BIC.

In the previous chapter we also introduced residuals or rather pseudo residuals as a type of model fit measure especially for HMMs. There we used it to determine the best downsampling factor and method, but we can also use it to compare the fit for different numbers of states. Figure 21 depicts these properties of the residuals. We can see that the KSS is very noisy and improves only marginally. For reference: the expected KSS of a normal distribution for that data size is approximately 0.001. In our case we can observe much higher values. Therefore we cannot really depend our decision of the number of states on this statistic. The autocorrelation of the residuals, on the other hand, shows a value of 0 at a number of states of 6, thus is optimal for a six state model.

As we have found out in our tests above, more states result in a good fitting model, meaning a better explanation of the data at hand. This comes at a cost though: With a bigger set of states to choose from for every time point, comes a higher uncertainty about the state predictions of the model as we can see from the (normalized) entropy of the state sequence in figure 22.

More importantly, regarding the underlying question of this project, a good fit is not necessarily our primary goal. If we wanted to predict future VeDBA values of the specific animals we would opt for the model with the best fit possible, but the VeDBA values are actually not directly of interest to us – We are primarily interested in the underlying behavioral states of the animals. So rather than asking what is the optimal number of states to fit the best model, we should ask the question how many states can we fit and still retain a biologically reasonable interpretation of the model states as behavioral states?

As you can see in figure 23, even for the model preferred by the BIC, the model does not find six distinct states. State 1 and 2 basically always occur together with state 2 mopping up noisy measurements from the resting

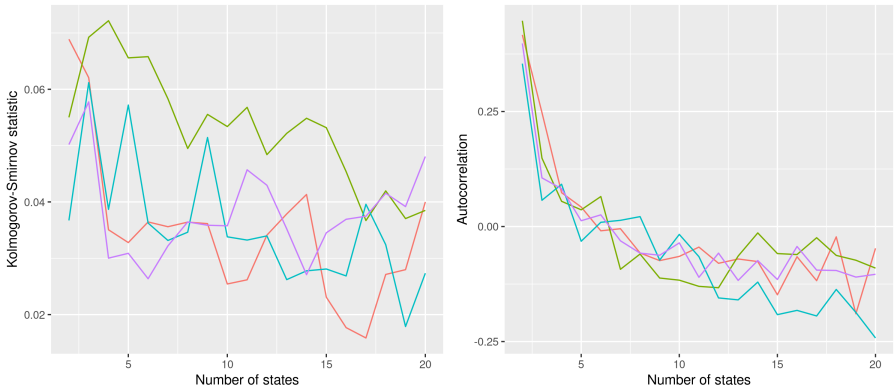

Figure 21: Normality and autocorrelation of the residuals.

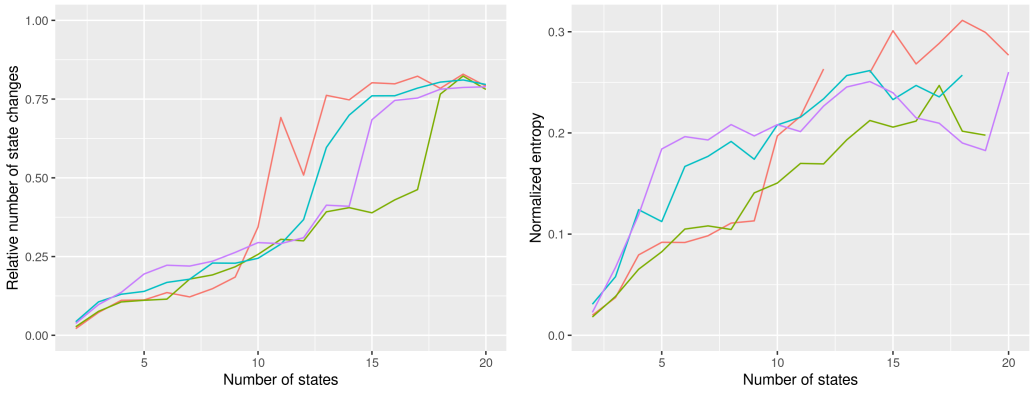

Figure 22: Left: Number of state changes in the predicted state sequence divided by the number of time points (5759). Right: Normalized entropy of the state sequence. Three values are missing due to numerical underflow during the calculation of the entropy measure.

intervals, resulting in a very low dissimilarity index of 0.36. Similarly, state 4 and 5 divide the same medium activity episodes vertically among each other with more or less the same dissimilarity index of 0.37. With that regard, state 3 arguably also just covers the lower values of these intervals (dissimilarity index 0.46 for 3 vs. 4)

So while the additional states may help fitting the model vertically to the data (figure 24), they do not enrich the temporal structure of the data significantly and hence do not contribute anything of relevance to the question of the circadian rhythm of the tuco-tucos.

With that in mind one could even argue that 3 states are too much: If we take a look at figure 23 again and compare the two state and three state model, we could claim that two states are enough to explain the data properly. As you can see there is a clear separation between state 1 and 2, whereas in the three state model state 2 and 3 are much less distinct. So simply speaking, we could justify that it is enough to assume two states: one for activity and the other for resting.

One the other hand, adding a third state improves the fit of the model substantially in terms of likelihood and especially autocorrelation. The improvement is much better comparing it to adding any further state, while the increase in entropy still is within an acceptable range and regarding the dissimilarity index, which is 0.52, state 2 and 3 are arguably sufficiently distinct. More so, if we take a look at Jannetti et al. (2019) and their data, we see that they had a daylight sensor attached to the animals to see if they are below or above ground. Maybe, with this additional information, we could be able to separate different types of behavior: rest, activity below ground and activity above. With this, we would have a strong biological motivation to opt for three states. Therefore, regarding all the previous points we made, we prefer the three state model.

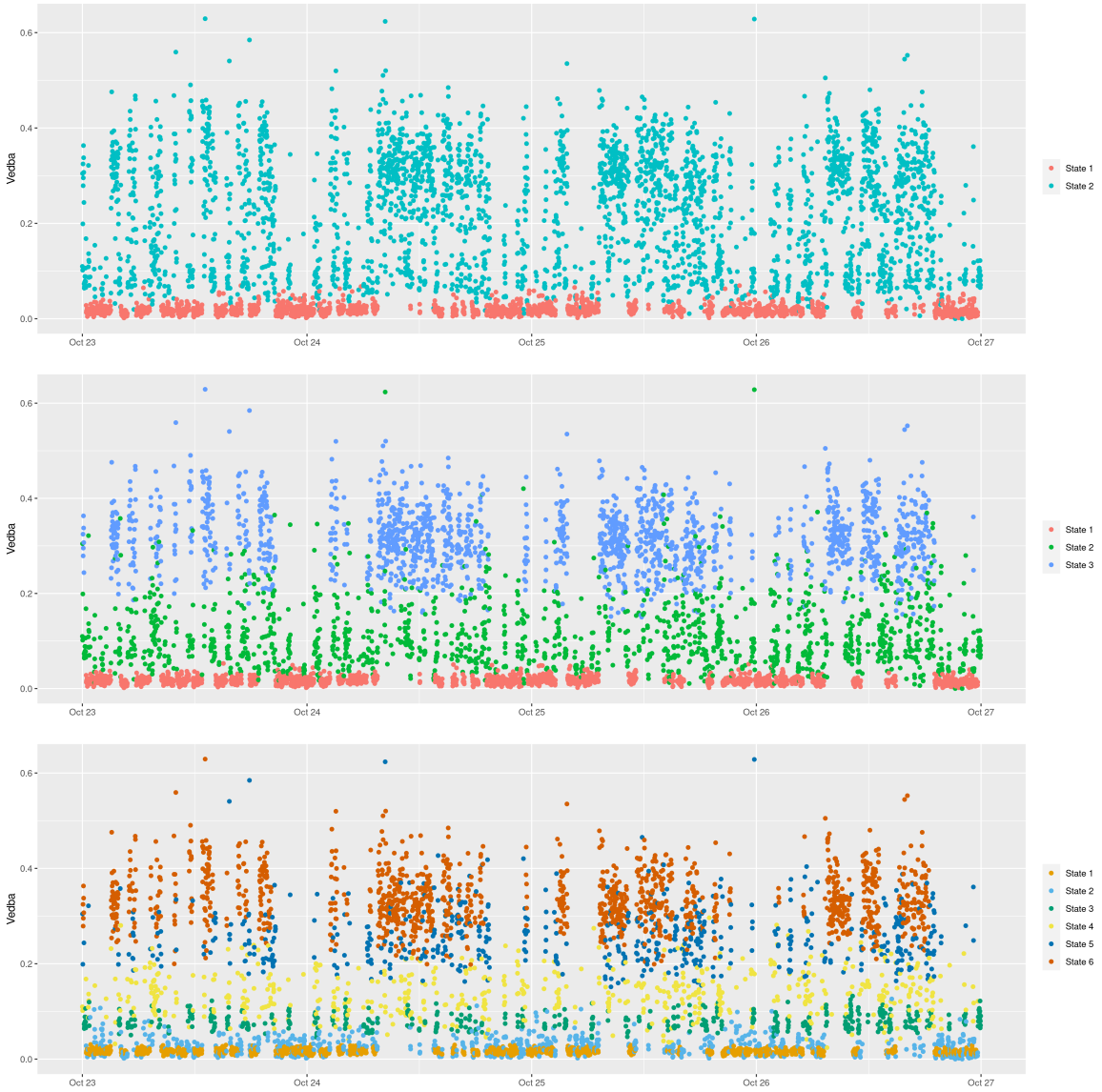

Figure 23: Sequence of VeDBA values for animal OCT09 colored by the state predictions of a two, three, and six state model.

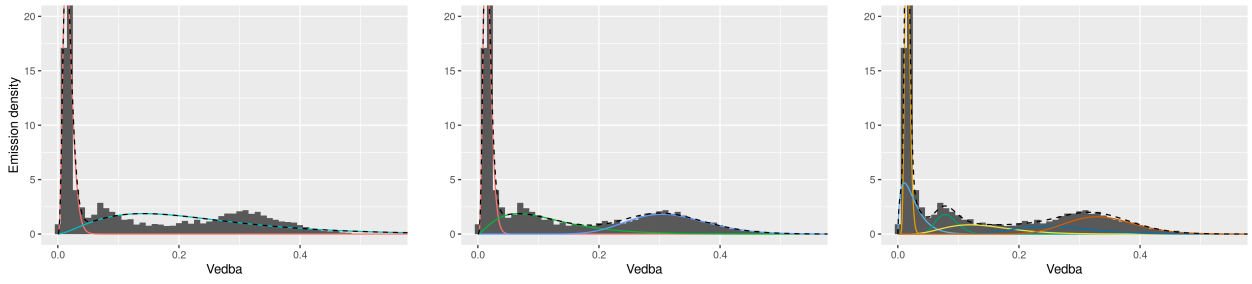

Figure 24: Marginal and emission distributions for animal OCT09 of a two, three, and six state model.

6 Interpretation of results

Now that we decided for a preprocessing procedure and the specification of the HMMs, lets take a look at the results. In this chapter we will collect some first impressions and observations, the final verdict on how to interpret our results can only be delivered by biologists, not mere statisticians.

First of all, it is remarkable how similar the emission distributions of the three states are. As you can see in table 4, the mean of the data varies much more across individuals than the means of the emission distributions of state 1 and 2. Only the means of state 3 show a similar variation. However, state 3 is also the state with the highest variance, therefore the overlap between the emission distributions of state 3 is still very high, despite the different means (see figure 25). To oversimplify: Animals that are more active don't run faster, they run more often. For example, the least active animal (JUL16) does not display significantly different emission distributions but rests more often and is less often in the high activity state (see table 5).

Animal	JUL16	MAR01	MAR02	OCT09	SD
Data	0.075	0.117	0.099	0.133	0.025
State 1	0.020	0.015	0.018	0.016	0.002
State 2	0.105	0.112	0.099	0.113	0.006
State 3	0.284	0.317	0.268	0.318	0.024

Table 4: Average VeDBA values of the data and means of the emission distributions of the models fitted on the data. The last column contains the standard deviation of the means across the different animals.

Animal	JUL16	MAR01	MAR02	OCT09	Avg.
δ_1	63	44	49	42	50
δ_2	23	32	26	27	27
δ_3	13	23	24	29	22

Table 5: Marginal distribution of the states (percentages).

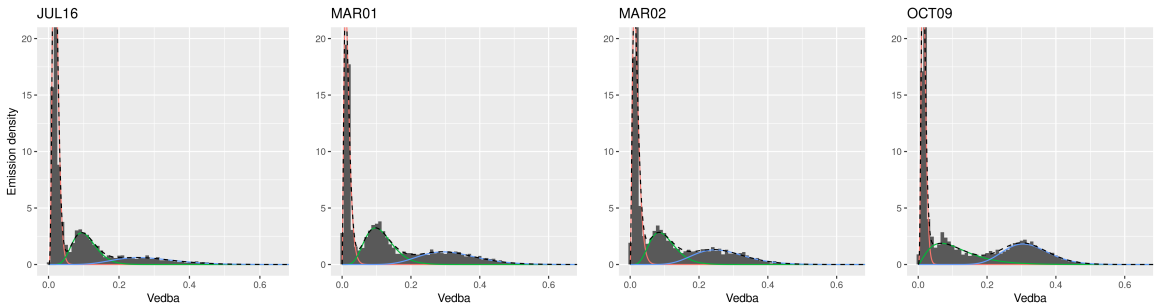

Figure 25: Marginal distributions of final models with histogram of the data and emission distribution of the three states.

Or to be more precise: The animal rests longer when it rests. We can read this temporal structure from the scatterplots of the VeDBA values colored by the predicted states in figure 26. All animals alternate periods of activity and times of rest very regularly throughout the whole day. Even in the night when the animals are least active there are never more than two hours without activity with the exception of JUL16. The JUL16 animal often rests for 2 to 3 hours while the other animals usually rest only 45 minutes to two hours without activity.[3]

[3]Note that due to the aggregation via the median, only minutes where the animal is active the majority of the time are visible as activity after the preprocessing. Therefore "two hours without activity" really means "two hours without more than 30 seconds of activity in a row". If we wanted to take shorter periods of activity into account we would need to keep higher resolution data or aggregate differently. For our purpose those are irrelevant, though.

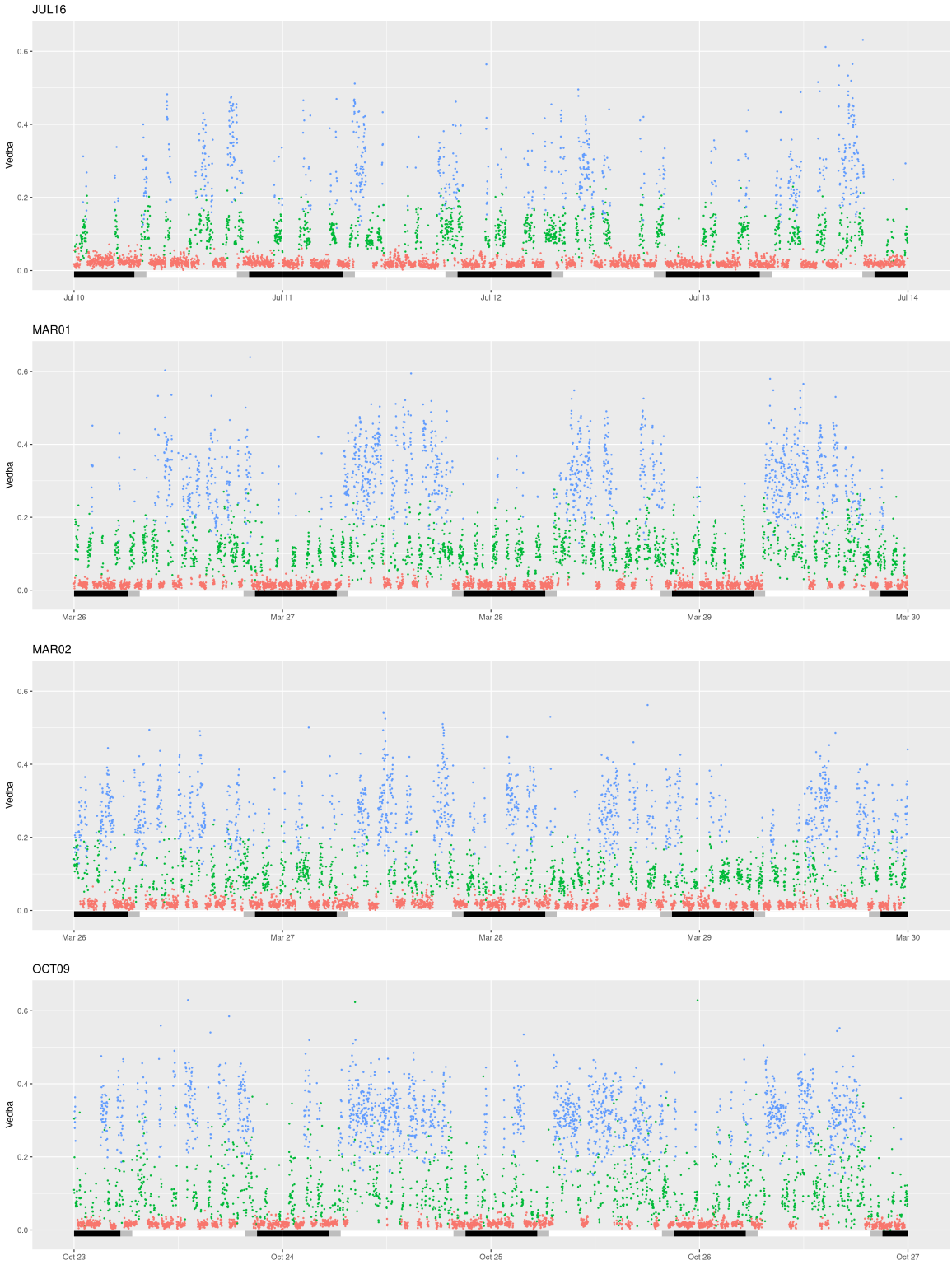

Figure 26: VeDBA values colored by states predicted by the final models. Black bars indicate nighttime, gray twilight, and white daylight.

All animals are most active during the day, 77% of state 3 classifications are during daylight hours. However, all animals also have a significant amount of activity during the night, especially the medium activity state is almost equally distributed throughout the day, so much so that we would not call the animals clearly diurnal

(see table 6).

Animal	JUL16	MAR01	MAR02	OCT09	Avg.
State 1	38	25	45	32	36
State 2	42	54	45	55	50
State 3	67	88	63	83	77

Table 6: Percentage of time points classified as a certain state that occur during daylight hours.

Figure 27 shows the kernel density estimate of the temporal density of state 3. We can see a structure during the day here that was hard to see in figure 26: On many days, the animals exhibit a high concentration of high activity time points after sunrise and a second one before sunset with a lower concentration of high activity time points at ca 15:00. For all animals, the first day of recording is the least structured in that sense which might indicate a disturbance of the circadian rhythm due to the capture.

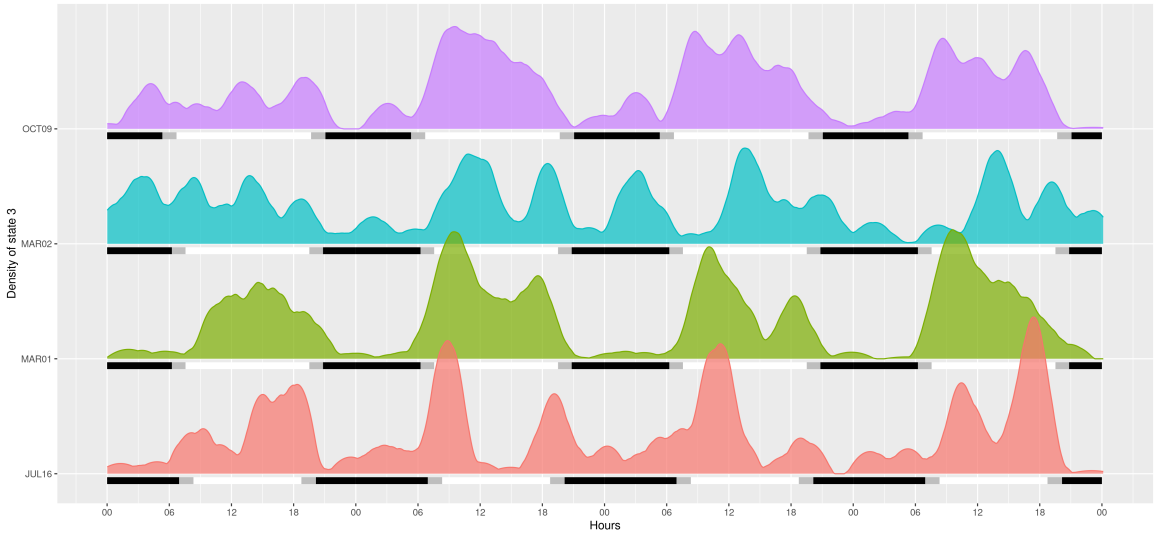

Figure 27: Kernel density estimation of the temporal density of state 3 with Epanechnikov kernel with a bandwidth of one hour.

To summarize, we could identify similar behavioral states for all four animals: Rest, medium and high activity. The animals differ in how often they exhibit these states. Further, there seems to be periodicity in the data: A 24 hour rhythm, especially in the high activity state and a faster and less regular rhythm of two to four hours, especially in the resting state. This temporal structure is not equally pronounced in all animals. For example MAR01 has a very clear 24 hour rhythm while MAR02 hasn't.

7 Conclusion

This report presented an analysis of the circadian rhythm of tuco-tucos. It was based on the data of four different animals collected and provided to us by Jefferson Silvério and his research partners. The raw data was summarized into the VeDBA as a measure for the activity level, which was used to construct and implement the presented models and methods.

We compared different downsampling factors and methods and concluded that aggregating each minute of data to a single sample using the median preserves the temporal structure in the data that is relevant for studying the circadian rhythm of the animals while removing irrelevant correlation on the micro level and allowing much more convenient handling of the data.

Then, we compared HMMs with different amounts of states. While HMMs with high amounts of states were able to fit the data closely, those states could not be plausibly interpreted as behavioral states of the model. Therefore, we decided for a model with only three states: Rest, medium, and high activity.

With our three state model we are able to show that all four animals are clearly most active during daylight hours, but they are consistently showing medium activity during night time, as well. Hence, based on our data we cannot say that the animals have a strictly diurnal rhythm.

It is important to note that, as we have mentioned in the previous section, this conclusion is not a final verdict but merely our first impression as non-experts: This is what the data indicated us and what we deemed to be most plausible, a chronobioligist might draw different conclusions from our results.

There are different directions in which one could continue studying the tuco-tuco behavior based on this dataset. For example:

- One could analyse the effect of environmental factors on the activity of the tuco-tucos. For example, the fact that the least active animal (JUL16) was recorded in the winter might indicate that temperature has an influence on the activity of the animals. This would reinforce results from Jannetti et al. (2019).
- One could try to explicitly model the 24 hour rhythm of the animals using the time of day as a covariate to the model.
- Assuming the 24 hour rhythm was modeled successfully, one could analyse how much the first hours of the data deviate from that model to find out how much and for how long the circadian rhythm of the animals is disturbed by the capture.
- One could try to learn more about the faster 1-4 hour rhythm of alternating rest and activity periods. In an HMM, the time we stay in a state i without switching is geometrically distributed with success probability $1 - \gamma_{ii}$. We can compute the expected time the animals rest until switching as $\gamma_{11}/(1 - \gamma_{11})$. This yields 42, 32, 17, and 19 minutes for the animals JUL16, MAR01, MAR02, and OCT09. The fact that these values do not really match our observation from figure 26 might indicate that the geometric distribution is not the best way to model state switching in this case. One could devise a hidden semi-Markov model instead.
- More information about the behavior of the animals (for example Jannetti et al. (2019) use daylight sensors to tell if an animal is above or bellow ground during the day) would possibly allow to train HMMs with more substantially distinct states.

References

C. V. Bouten, K. R. Westerterp, and J. D. J. Maarten Verduin. Assessment of energy expenditure for physical activity using a triaxial accelerometer. *Medicine & Science in Sports & Exercise*, 26(12):1516–1523, 1994.

A. C. Gleiss, R. P. Wilson, and E. L. C. Shepard. Making overall dynamic body acceleration work: on the theory of acceleration as a proxy for energy expenditure. *Methods in Ecology and Evolution*, 2(1):23–33, 2011.

D. Hernando, V. Crespi, and G. Cybenko. Efficient computation of the hidden markov model entropy for a given observation sequence. *IEEE Transactions on Information Theory*, 51:2681 – 2685, 08 2005.

M. Jannetti, C. L. Buck, V. Valentinuzzi, and G. Oda. Day and night in the subterranean: measuring daily activity patterns of subterranean rodents (ctenomys aff. knighti) using bio-logging. *Conservation physiology*, 7, 2019.

G. A. Meijer, K. R. Westerterp, and F. T. H. Hans Koper. Assessment of energy expenditure by recording heart rate and body acceleration. *Medicine & Science in Sports & Exercise*, 21:343–347, 1989.

L. Qasem, A. Cardew, A. Wilson, I. Griffiths, L. G. Halsey, E. L. C. Shepard, A. C. Gleiss, and R. Wilson. Tri-axial dynamic acceleration as a proxy for animal energy expenditure; should we be summing values or calculating the vector? *PloS one*, 7(2):e31187, 2012.

R. Wilson, C. White, F. Quintana, L. Halsey, N. Liebsch, G. Martin, and P. Butler. Moving towards acceleration for estimates of activity-specific metabolic rate in free-living animals: The case of the cormorant. *The Journal of animal ecology*, 75:1081 – 90, 2006.

W. Zucchini, I. L. MacDonald, and R. Langrock. *Hidden markov models for time series*. CRC Press, Boca Raton ; London ; New York, second edition edition, 2016.