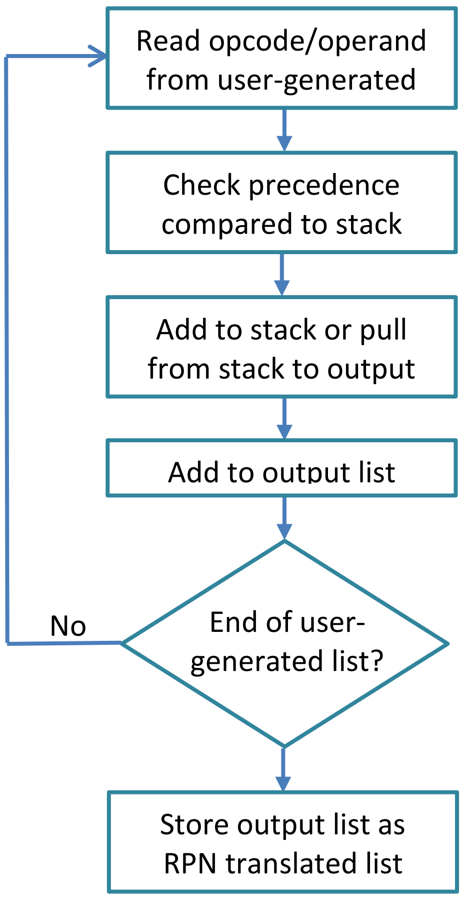
Paper:

Wilson, R., Holton, M., di Virgilio, A., Williams, H., Shepard, E., Lambertucci, S., Quintana, F., Sala, J., Balaji, B., et. al. (2018). Give the machine a hand: A Boolean time-based decision-tree template for rapidly finding animal behaviours in multisensor data. *Methods in Ecology and Evolution*

http://dx.doi.org/10.1111/2041-210X.13069

Supplementary Material 1: LoCoD method operation

Each command the user selects has an opcode and possibly an operand. As the user selects various commands, a list of these opcodes/operands is stored in order of entry. This is then parsed into Reverse-Polish-Notation (AKA Post-Fix):



For example:

If (Accel X Smoothed < 1.5) then “Mark-event”
1,3,7,8,11,1.5,15,22
1: If (
3: Channel (7) “Accel X Smoothed”
8: <
11: Value (1.5)
15:)
22: Mark-event

This is simply translated to Post-fix as:
Accel X Smoothed (value of), Value 1.5, <

3,7,11,1.5,8
i.e. Post-fix would process this by reading the value of Accel X Smoothed and pushing this onto a stack, pushing value of 1.5 onto the same stack. The “<” command requires the two previously stored values on the stack. A “<” comparison between these two values results in a True or False.

The Time-Series algorithm:
Definition: *ETNE* is ‘*Extend To Next Event’*, which is where an *Element’s* validity is checked beyond its stated valid range. It is checked from the starting *Event* number to the end of *Event* number + *Range* + *Flex* i.e. as far as the next *Element* might exist. The point where it fails (if at all) is stored
For every *Event* stored in memory, each *Element* is parsed and the result stored
Once all Elements have been parsed through all in-memory data, the program checks if the TS expression passes for each *Event*

- For all *n Elements*, beginning from the first *Event*, the program begins with *Element* 1
- The program checks if there are *Element*$_i$ (*Valid*) consecutive points beginning at *Event n* to *n+Valid*
- If *Element*$_i$ has passed, the program then checks if *Element*$_i$ has *ETNE* enabled; if so, the program also checks from *Event n* to *Event n+Range+Flex* and stores the point of fail, or simply *n+Range+Flex* if no fail occurred
- If parsing *Element* > 1, the program then checks if the previous *Element* had *ETNE* as part of its construct. If so, it checks at which datapoint the previous *Element* failed. If it failed before the point the current *Element* passed, then the current *Element* fails.
- If the current *Element* failed, the program then moves onto *Event n+1* and starts again with *Element* 1
- If the current *Element* passed, the program then moves onto the next *Element*
- If all *Elements* have passed, the program then *Bookmarks* from the first *Element’s Event* to the last *Element’s Event + Valid* width; it then moves point *n* onto the end of the *Bookmark* just created as this behaviour’s existence has been confirmed.

Supplementary Material 2: Behaviour description in terms of LoCoD

Any behaviour can be described by the sequence of defined motions, each motion defining a base element of the behaviour. Each base element differs in the time over which it is performed and hence so does the entire duration of the behaviour. The examples shown here have been selected as they differ in the type and number of base elements involved as well as the duration of the behaviour.

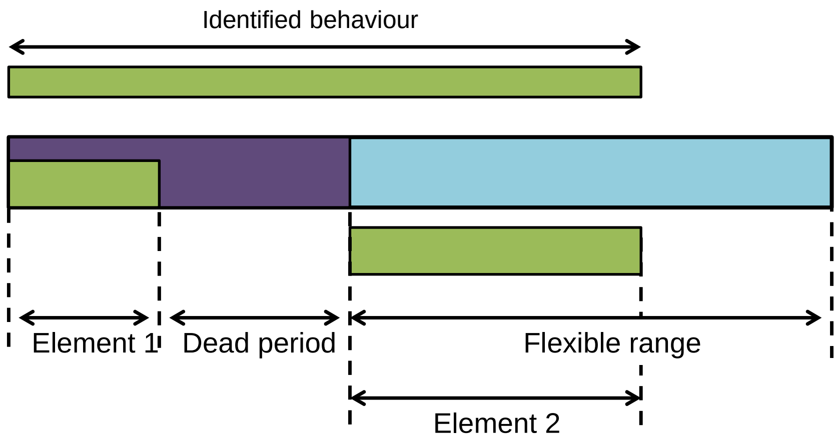


FIGURE 1 Schematic diagram of a behavior in terms of; behavioural elements, a ‘dead period’ (see text) and a flexible range of time within which behavioural elements must follow for all the behavioural elements to constitute one LoCoD

Short-period behaviours

Sheep biting

For sheep and most herbivores, grazing is a complex behaviour that can be decomposed into smaller behaviours such as biting and chewing. Biting consists in the extraction of the foodstuff from the environment and chewing is the first stage of food processing. Herbivore bites are typically short and high frequency behaviours that can occur in periods of less than a second. Biting is commonly characterized by an abrupt head movement that typically occurs in one of two main directions; forward and backward, which is also accompanied by an increase in standard movement metrics (e.g. VeDBA). These abrupt head movements indicative of biting are well represented in the surge axis of the acceleration, and the differential of this signal can be smoothed to reduce the influence of overall motion of the animal while feeding. Although sheep biting is a short-period behaviour, the duration and frequency of bites can be variable according the type of vegetation that individuals consume. For this reason, we included a flexibility window of 10 consecutive data points (corresponding to 0.25 s if the data are recorded at 40 Hz) to be able to detect this variability (see supplementary information 2 for detail).

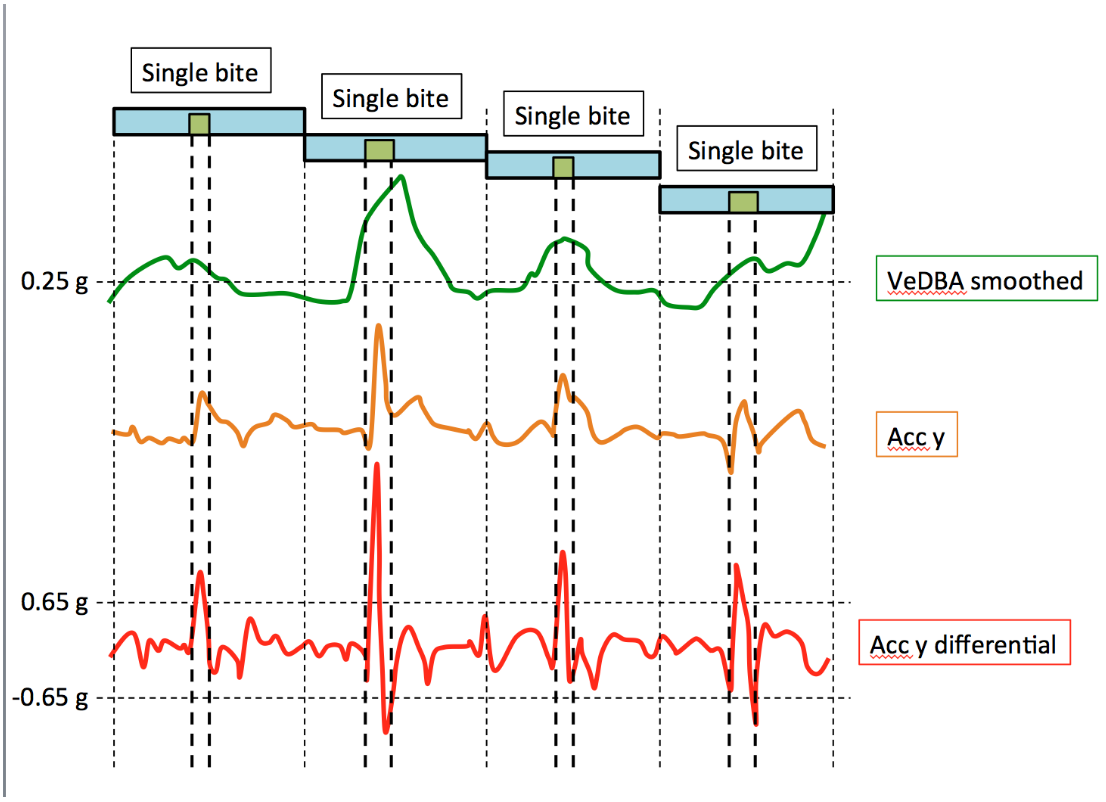


FIGURE 2 Schematic diagram to demonstrate how the single bites of a sheep can be defined within the BE and flexible search criteria (colour coding for these as in Fig. 1). For precise details, see supplementary information 2. Four single bites are shown here as performed in sequence.

Walking Penguin

In contrast to the dive, the signal created by a penguin as it walks is comparatively short-period and complex, yet highly stylised in its pattern of motion. As the penguin makes a double step in walking, it sways from side to side, creating peaks and troughs in the smoothed signal in the sway axis of the acceleration. The differential of this signal can be smoothed again to reduce the noise manifest in effects of style or gait on the overall motion of the behaviour and can thus be used to classify all examples of a double step in walking. Differences in speed will still be apparent however, and so the use of a time flexible algorithm to classify the behaviour is ideal (see supplementary information 2 for detail).

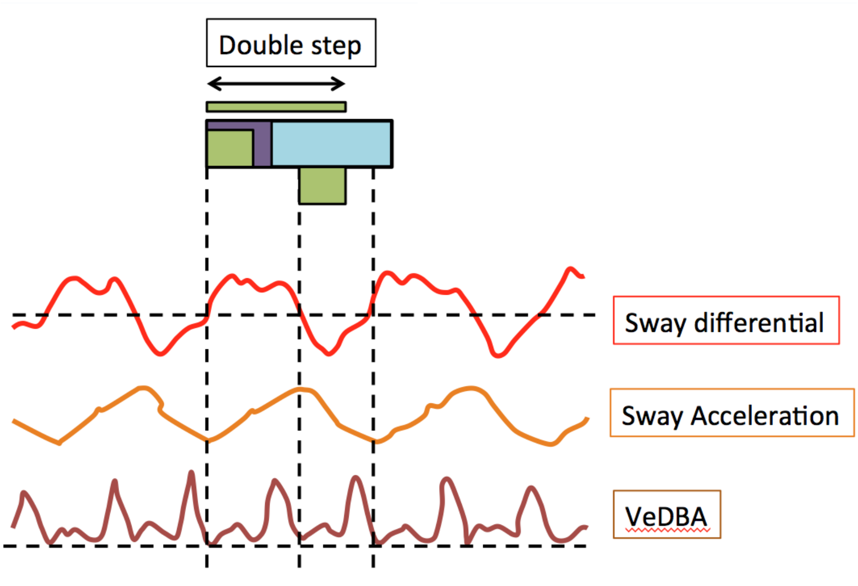


FIGURE 3 Schematic diagram to demonstrate how walking by a penguin can be defined within various BEs, dead elements and flexible search criteria (colour coding for these as in Fig. 1). For precise details, see supplementary information 2.

Working example with penguin walking

We present a step-by step example of the application of the LoCoD method to label walking behaviour within a section of data recorded from the device attached to the Magellanic Penguin (penguin walking_data section.raw). This includes a video attached (penguin walking.mp4) of the precise actions undertaken during the process. [Note that the program provided can load acceleration data even if they are not derived from the 'Daily Diary' provided that the data are arranged in columns with TAB as a separator. In this case, the filename must be Xxx.col and under 'file of type', the 'col' section needs using. In this case, you will be asked to specify sampling rate.]
Firstly, walking is identified as the signal in Figure 3 and a pattern of change identified in the smoothed y-acceleration. The specific process (mirrored in the video) is;

1. Load raw data file with 40 Hz sampling frequency
2. Smooth the acceleration
3. Derive the differential of the smoothed y-acceleration
4. Use the 'display overlay window' to establish thresholds from templates
5. Define the base element equations based on the values in the display
6. Open 'build a time series based behaviour' and define time series windows
7. Bookmark matches to the template
8. Further rules can be applied to improve classification accuracy. In this case, walking can be regarded as a continuous behaviour so we merge bookmarks that occur within 80 data points (2 seconds at 40 Hz sampling) and then remove bookmarks that are fewer than 80 data points in duration.
9. Note that all walking is correctly identified by this process.
10. These bookmarks can be exported as a master txt file for analyses in other software.

Medium-period behaviour

Themalling Condor

During thermal soaring, a condor must make a series of complete rotations to maintain a position within the thermal and rise in the updraft. Each complete rotation can easily be seen in the magnetometer data as the bird turns through all headings in relation to magnetic north. Hence each complete turn is defined by a sine wave pattern in the x-axis of the magnetometer sensor, the length of which depends on the time it takes for the bird to complete the turn. The behaviour is also expected to increase in duration from seconds to minutes through a single climb and with increasing thermal strength and so this behaviour lends itself to classification with temporal flexibility rather than any restricted classification by pair-wise correlation, for example. In terms of classification the sine wave can be reduced to two base elements, the first and second halves of the complete turn (see supplementary information 2).

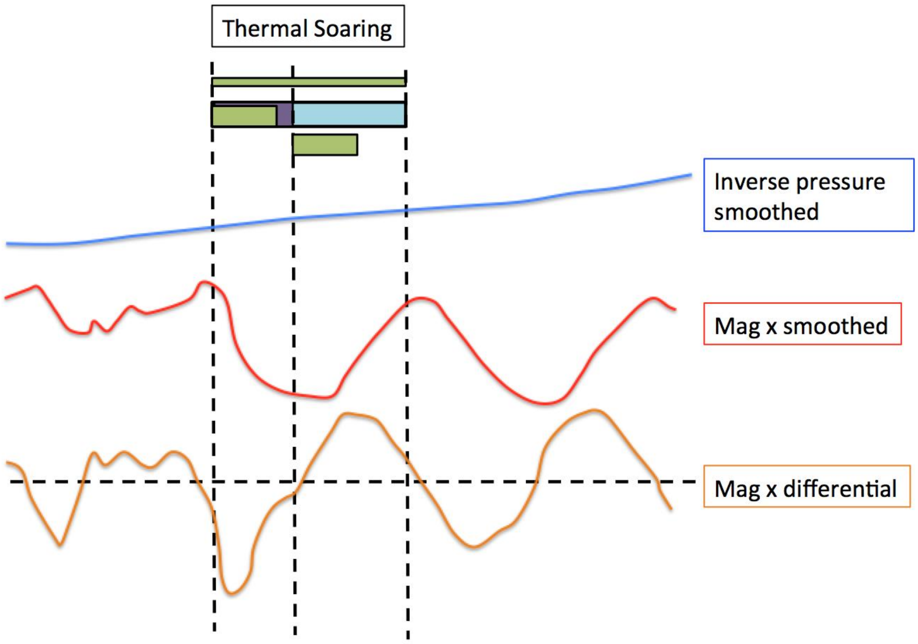


FIGURE 4 Schematic diagram to demonstrate how thermal soaring by a condor can be defined within various BEs, dead elements and flexible search criteria (colour coding for these as in Fig. 1) using patterns in the output from the magnetometer and barometric pressure sensor. For precise details, see supplementary information 2.

Long-period behaviour

Cheetah Stalking

When a cheetah stalks its prey, it reduces the acceleration signal in its movement, crouching low to the ground, moving slowly closer to its prey. Thus, in terms of signal outputs, the rate of change of smoothed acceleration defines the stalk poorly as there is very little change in the animal's postural orientation. Instead, the defining feature is a lack of variation in any of the three acceleration signals and hence a consistently low VeDBA. In this stalking phase, as the animal moves in on its prey, changes in the smoothed magnetometry signals may also be evident although the rate of change in directional orientation is not specific to the behavior. The chase follows the low VeDBA stalk immediately. This is characterized by sprinting and a dramatic increase in the dynamism of movement, resulting in an extremely high VeDBA relative to other behaviours. Stalking behavior in the cheetah can therefore be identified using the two BEs that make up the LoCoD; i) a low VeDBA stalk, followed by ii) a high VeDBA chase, each BE lasting at least several seconds.

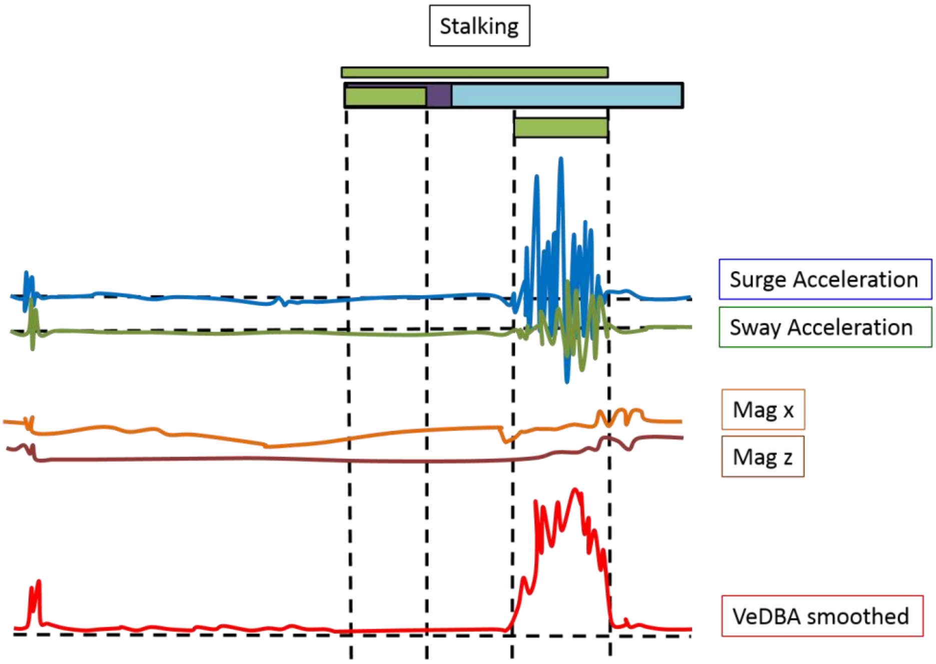


FIGURE 5 Schematic diagram to demonstrate how stalking by a cheetah can be defined within various BEs, dead elements and flexible search criteria (colour coding for these as in Fig. 1). For precise details, see supplementary information 2.

Supplementary Material 3: LoCoD method algorithm design

Table S3.1: LoCoD method algorithm design for sheep biting. The different design components showed in this table are; the variables used for processing, the base elements identified, and the time series of those base elements. Numeric values shown refer to numbers of consecutive data points recording at 40 Hz so that, for example, the smoothing window is over 1 s.

Sheep Biting – short-period behaviour				
Processing	**Signal**	**Smoothing window**	**Differential range**	
	VeDBA	40	-	
	Acc y	2	5	
Base elements	Bite	If (VeDBA smoothed > 0.25) AND ABS(Diff_Accel y)>0.65) then mark events *(Include forward and backward head movement by using* ABS()		
Time series	**Element**	**Present**	**range**	**Flexibility**
	1 Head movement (Forward or Backward)	1	-	10

Table S3.2: LoCoD method algorithm design for penguin walking. The different design components showed in this table are; the variables used for processing, the base elements identified, and the time series of those base elements. Numeric values shown refer to numbers of consecutive data points recording at 40 Hz.

Penguin Walking – short-period behaviour				
Processing	**Signal**	**Smoothing window**	**Differential range**	
	Acc y	10	5	
Base elements	Step left	If (SM (Diff_Accel Y smooth, 5) < -0.1) then mark events		
	Step right	If (SM (Diff_Accel Y smooth, 5) > 0.1) then mark events		
Time series	**element**	**present**	**range**	**Flexibility**
	1 Step left	6	16	16
	2 Step right	6	-	-

Table S3.3: LoCoD method algorithm design for condor thermalling. The different design components showed in this table are; the variables used for processing, the base elements identified, and the time series of those base elements. Numeric values shown refer to numbers of consecutive data points recording at 40 Hz.

Condor Thermalling – medium-period behaviour				
Processing	**signal**	**Smoothing window**	**Differential range**	
	pressure	830	200	
	Mag x	40	80	
Base elements	½ turn section 1	if((smooth(diff_mag_x_smooth,20)>0) AND (diff_pressure_smooth<0))then mark events		
	½ turn section 2	if((smooth(diff_mag_x_smooth,20)<0) AND (diff_pressure_smooth<0))then mark events		
Time series	**element**	**present**	**range**	**Flexibility**
	1 ½ turn section 1	200	400	200
	2 ½ turn section 2	200	-	-

Table S3.4: LoCoD method algorithm design for cheetah stalking. The different design components showed in this table are; the variables used for processing, the base elements identified, and the time series of those base elements. Numeric values shown refer to numbers of consecutive data points recording at 40 Hz.

Cheetah Stalking – long-period behaviour				
Processing	**signal**	**Smoothing window**	**Differential range**	
	VeDBA	10	NA	
Base elements	Stalk	If (SM (VeDBA Smoothed, 5) < 0.5) then mark events		
	Chase	If (SM (VeDBA Smoothed, 5) > 0.55) then mark events		
Time series	**element**	**present**	**range**	**Flexibility**
	1 Stalk	400	600	1200
	2 Chase	340	-	-

Supplementary Material 4: LoCoD and Machine learning performance

Here, we provide a brief description of each machine learning algorithm available in AccelerRater:

K-Nearest neighbors: This is a non-parametric method that labels a new sample/observation using a vote between the K points in the training data set nearest to it. The method is a primitive form of machine learning that is often referred to as 'lazy learning' because induction occurs during run time. By default, we set K=3. For more detail, see James et al. (2013) and Bidder et al. (2014).

Linear SVM: Linear support vector machines compute the maximum margin hyperplane between two classes. The multi-class extension used computes such a hyperplane between every two classes and uses a vote to determine the class for a new point quantifying the similarity of a pair of observations using Pearson correlation. More detail is provided by James et al. (2013).

RBF kernel SVM: This model is similar to a Linear SVM, but instead of using Gaussian kernels employs Radial Basis Functions (RBF) kernels. The algorithm automatically determines centres, weights and thresholds that minimize an upper bound on the expected test error. See Scholkopf et al. (1996) for more detail.

Decision tree: This is a probabilistic method that works on binary decisions that are constructed hierarchically. Basically, this method consists of a set of hierarchical decision rules developed to predict the class of unclassified samples. Each rule can branch into another rule or a terminal category.

Random forest: This method consists of a combination of decision trees where each classifier is generated using a random vector sampled independently from the input vector. This means that the procedure is similar to a decision tree but includes introduced stochasticity. Instead of potentially using all the variables to determine the best split at each node, only a randomly selected subset of variables is used. For more detail, see Breiman (1999) and Breiman (2001).

Naïve Bayes: The Naïve Bayes algorithm is a simple probabilistic classifier that calculates a set of probabilities by counting the frequency and combinations of values in a given data set. The algorithm uses Bayes theorem and has a strong assumption that all attributes are independent given the value of the class variable (i.e., features are conditionally independent). More detail is given in Patil & Sherekar (2013).

LDA: The Linear Discriminant Analysis method is basically a linear model assuming Gaussian distributions with equal covariance. See James et al. (2013) for more detail.

QDA: The Quadratic Discriminant Analysis method is the same as LDA, but without assuming equal covariance (i.e., assumes that each class has its own covariance matrix). For more information, see James et al. (2013).

ANN: Artificial Neural Networks (ANNs) are computer-based algorithms that imitate the structure and behavior of neurons in the human brain. These algorithms can be trained to recognize and categorize complex patterns. Pattern recognition is achieved by adjusting parameters of the ANN by a process of error minimization through learning from experience. They can be calibrated using any type of input data and the output can be grouped into any given number of categories. More detail is given in Bishop (1995).

References:

Bidder, O. R., Campbell, H. A., Gómez-Laich, A., Urgé, P., Walker, J., Cai, Y., ... & Wilson, R. P. (2014). Love thy neighbour: automatic animal behavioural classification of acceleration data using the k-nearest neighbour algorithm. PloS one, 9(2), e88609

Bishop, C.M. 1995. Neural Networks for Pattern Recognition. Clarendon Press, Oxford.

Breiman, L. 1999. Random forests - Random Features. Technical Report 567, Statistics Department, University of California, Berkeley, ftp://ftp.stat.berkeley.edu/pub/users/breiman.

Breiman, L. 2001. Random forests. Machine learning, 45(1), 5-32.

Scholkopf, B., Sung, K. K., Burges, C. J., Girosi, F., Niyogi, P., Poggio, T., & Vapnik, V. 1997. Comparing support vector machines with Gaussian kernels to radial basis function classifiers. IEEE transactions on Signal Processing, 45(11), 2758-2765.

James, G., Witten, D., Hastie, T., & Tibshirani, R. 2013. An introduction to statistical learning. New York: springer.

Patil, T. R., & Sherekar, S. S. 2013. Performance analysis of Naive Bayes and J48 classification algorithm for data classification. International Journal of Computer Science and Applications, 6(2), 256-261.

Table S3.1: LoCoD and Machine learning performance for sheep biting. Where performance is measured in terms of the number of True Positive (TP), True Negative (TN), False positive (FP) and False Negative (FN) results and the performance metrics of Recall (R), Precision (P) and Accuracy (A) have been calculated. Note that for the Machine learning methods, each data point is labelled, so it is possible to assign a category of TN. However, for the LoCoD method, data is labelled within the LoCoD, so it is not possible to assign a category of TN as a non-existent LoCoD cannot be falsely identified. For the latter, an accuracy value cannot be calculated.

behaviour	**Sheep biting**						
	Time (s)	Performance			Cases		
		R	P	A	TP	FN	FP
Manual	2039	1	1	1	171	0	0
LoCoD	**1.5**	**0.887**	**0.871**	**NA**	156	35	29
Nearest Neighbour	243	0.000	0.000	0.998	0	0	171
Linear SVM	3189	0.000	0.000	0.998	0	0	171
RBF SVM	253	0.000	0.000	0.998	0	0	171
Decision Tree	242	0.000	0.000	0.997	0	171	0
Random Forest	281	0.000	0.000	0.998	0	171	0
Naïve Bayes	317	0.000	0.000	0.998	0	0.0171	171
LDA	264	0.000	0.000	0.998	0	0	171
QDA	353	0.988	0.002	0.977	169	3	75
ANN	3451	0.000	0.000	0.988	0	0	171

Table S3.2: LoCoD and Machine learning performance for penguin walking represented by single steps. Where performance is measured in terms of the number of True Positive (TP), True Negative

(TN), False positive (FP) and False Negative (FN) results and the performance metrics of Recall (R), Precision (P) and Accuracy (A) have been calculated. Note that for the Machine learning methods, each data point is labelled, so it is possible to assign a category of TN. However, for the LoCoD method, data is labelled within the LoCoD, so it is not possible to assign a category of TN as a non-existent LoCoD cannot be falsely identified. For the latter, an accuracy value cannot be calculated.

behaviour	**Penguin walking**						
	Time (s)	Performance			Cases		
		R	P	A	TP	FN	FP
Manual	2040	1.000	1.000	1.000	343	0	0
LoCoD	**14**	**0.982**	**0.984**	**NA**	335	8	8
Nearest Neighbour	77	0.971	0.965	0.973	337	6	6
Linear SVM	359	1.000	0.752	0.862	343	0	81
RBF SVM	79	0.939	0.973	0.964	322	21	6
Decision Tree	80	0.965	0.964	0.971	331	12	9
Random Forest	82	0.979	0.964	0.976	336	7	9
Naïve Bayes	75	0.992	0.761	0.866	340	3	0
LDA	74	0.988	0.762	0.867	343	0	78
QDA	77	0.759	0.709	0.771	261	82	76
ANN	405	0.925	0.966	0.947	317	26	13

Table S3.3: LoCoD and Machine learning performance for condor thermalling. Where performance is measured in terms of the number of True Positive (TP), True Negative (TN), False positive (FP) and False Negative (FN) results and the performance metrics of Recall (R), Precision (P) and Accuracy (A) have been calculated. Note that for the Machine learning methods, each data point is labelled, so it is possible to assign a category of TN. However, for the LoCoD method, data is labelled within the LoCoD, so it is not possible to assign a category of TN as a non-existent LoCoD cannot be falsely identified. For the latter, an accuracy value cannot be calculated. The machine learning methods presented in this table are those that could be completed within 5 hours.

behaviour	***Condor thermalling***						
	Time (s)	Performance			Cases		
		R	P	A	TP	FN	FP
Manual	2220	1	1	1	146	0	0
LoCoD	**9**	**0.87**	**0.73**	**NA**	127	19	47
Nearest Neighbour	2182	0.144	0.257	0.797	21	125	11
Decision Tree	2358	0.006	0.355	0.838	1	145	0
Random Forest	2998	0.000	0.000	0.840	0	146	0

Table S3.4: LoCoD and Machine learning performance for cheetah stalking. Where performance is measured in terms of the number of True Positive (TP), True Negative (TN), False positive (FP) and False Negative (FN) results and the performance metrics of Recall (R), Precision (P) and Accuracy (A) have been calculated. Note that for the Machine learning methods, each data point is labelled, so it is possible to assign a category of TN. However, for the LoCoD method, data is labelled within the LoCoD

LoCoD, so it is not possible to assign a category of TN as a non-existent LoCoD cannot be falsely identified. For the latter, an accuracy value cannot be calculated. The machine learning methods presented in this table are those that could be completed within 5 hours.

behaviour	***Cheetah stalking***						
	Time (s)	Performance			Cases		
		R	P	A	TP	FN	FP
Manual	180	1	1	1	10	0	0
LoCoD	**7.2**	**0.89**	**0.89**	**NA**	8	1	1
Nearest Neighbour	4045	0.996	0.986	0.983	10	0	9
Decision Tree	3470	0.999	0.986	0.985	10	0	10
Random Forest	4217	1	0.985	0.985	10	0	10
Naïve Bayes	3179	0.189	0.030	0.897	2	8	1
LDA	3016	0.056	0.259	0.984	9	0	9