hormLong: An *R* package for longitudinal data analysis in wildlife endocrinology studies

Benjamin Fanson, Kerry V Fanson

The growing number of wildlife endocrinology studies have greatly enhanced our understanding of comparative endocrinology, and have also generated extensive longitudinal data for a vast number of species. However, the extensive graphical analysis required for these longitudinal datasets can be time consuming because there is often a need to create tens, if not hundreds, of graphs. Furthermore, routine methods for summarising hormone profiles, such as the iterative baseline approach and area under the curve (AUC), can be tedious and non-reproducible, especially for large number of individuals. We developed an *R* package, *hormLong*, which provides the basic functions to perform graphical and numerical analyses routinely used by wildlife endocrinologists. To encourage its use, *hormLong* has been developed such that no familiarity with *R* is necessary. Here, we provide a brief overview of the functions currently available and demonstrate their utility with previously published Asian elephant data. We hope that this package will promote reproducibility and encourage standardization of wildlife hormone data analysis.

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1546v1 | CC-BY 4.0 Open Access | rec: 30 Nov 2015, publ: 30 Nov 2015

For PeerJ

hormLong: An *R* package for longitudinal data analysis in wildlife endocrinology studies

Benjamin G. Fanson and Kerry V. Fanson

Centre for Integrative Ecology, School of Life and Environmental Sciences, Deakin University, Waurn Ponds, VIC, Australia

Corresponding author:

Benjamin Fanson

Centre for Integrative Ecology, School of Life and Environmental Sciences, Deakin University, 75 Pigdons Road, Waurn Ponds, VIC 3216, Australia

E-mail: bfanson@gmail.com

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1546v1 | CC-BY 4.0 Open Access | rec: 30 Nov 2015, publ: 30 Nov 2015

ABSTRACT

The growing number of wildlife endocrinology studies have greatly enhanced our understanding of comparative endocrinology, and have also generated extensive longitudinal data for a vast number of species. However, the extensive graphical analysis required for these longitudinal datasets can be time consuming because there is often a need to create tens, if not hundreds, of graphs. Furthermore, routine methods for summarising hormone profiles, such as the iterative baseline approach and area under the curve (AUC), can be tedious and non-reproducible, especially for large number of individuals. We developed an *R* package, *hormLong*, which provides the basic functions to perform graphical and numerical analyses routinely used by wildlife endocrinologists. To encourage its use, *hormLong* has been developed such that no familiarity with *R* is necessary. Here, we provide a brief overview of the functions currently available and demonstrate their utility with previously published Asian elephant data. We hope that this package will promote reproducibility and encourage standardization of wildlife hormone data analysis.

Keywords

Area under the curve, baseline, peak detection, non-invasive hormone monitoring, steroid, faecal glucocorticoid metabolites, stress, ovulation

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1546v1 | CC-BY 4.0 Open Access | rec: 30 Nov 2015, publ: 30 Nov 2015

INTRODUCTION

Longitudinal hormone monitoring is routinely used in wildlife endocrinology studies and provides a unique insight into endocrine physiology that cannot be obtained from single samples. The amount of longitudinal endocrine data is rapidly increasing due to the development of new techniques and advances in technology (e.g., non-invasive hormone monitoring, catheterization techniques, cheaper assays). Consequently, researchers routinely handle large endocrine datasets with an extensive number of samples. One of the greatest challenges with these large datasets is efficient and reproducible data analysis. Analysing longitudinal hormone data generally includes (1) graphical visualization of the data, (2) identification of peaks, and (3) quantifying the magnitude of the response.

Similar to other time series data (Cowpertwait & Metcalfe 2009; Montgomery et al. 2015), graphical analysis plays an important role in identifying patterns in hormone profiles. Researchers often monitor dozens of individuals, but create profiles for each individual one-at-a-time. Furthermore, temporal events (e.g. pregnancy, mating, stressors) are often added to graphs by hand. This process of creating dozens of graphs, marking events, and updating each graph separately becomes quite time-consuming. In addition, when multiple hormones are being monitored, it is useful to overlay hormone profiles in order to explore temporal correlations. However, this involves restructuring each individual dataset, which takes yet more time and can introduce error when done by hand.

Another challenge with analysing longitudinal hormone data is being able to distinguish the signal from the noise. There is a certain amount of inherent variability in any hormone profile due to both biological (e.g., pulsatile release, variability in steroid metabolism) and methodological factors (e.g., sampling design, pipetting error, assay variability). One common approach for identifying meaningful increases (peaks) in longitudinal datasets is the iterative baseline approach (Brown et al. 1996; Clifton & Steiner 1983). In this approach, hormone values exceeding the mean + (*n* * SD) are excluded, where *n* is the criterion for the number of standard deviations (SD) used in the calculation. The mean and SD are recalculated, and this culling processes is repeated until no points exceed the cut-off. Remaining values are considered "baseline" values and excluded points are considered "peaks". The appropriate value of *n* needs to be adjusted depending on the characteristics of the dataset (number of samples and amount of variation). Although this approach is really useful for identifying peaks, it can be tedious to run these iterative calculations for each study subject, and this becomes even more cumbersome when calculating and comparing different values of *n*.

In addition to detecting presence/absence of peaks (above), it is often desirable to quantify the magnitude of the response. One approach is to calculate the magnitude of the peak using either absolute difference (peak minus baseline) or relative increase (ratio of peak to baseline). A more complicated method is to calculate the area under the curve (AUC; (Cockrem & Silverin 2002; Sheriff et al. 2010). An advantage of this technique is that it incorporates both the magnitude of the peak as well as the duration, which are both biologically meaningful. Without specialized software, the AUC can be a tedious calculation and hinders reproducibility.

To facilitate efficient and reproducible data analysis, we developed a user-friendly *R* package that provides wildlife endocrinologists with a toolkit for analysing longitudinal hormone data and requires no prior programming experience. The package includes functions allowing for exploratory graphical analysis (including mass production of longitudinal profiles, box plots, and overlaying multiple hormones), iterative baseline calculation, and AUC calculation. To demonstrate the utility of this package, we analysed a previously published hormone dataset (Fanson et al. 2014). This study looked at changes in circulating cortisol across the estrous cycle (i.e., relative to progesterone) in Asian elephants. We included these data as an example dataset called *hormElephant* in the package.

DESCRIPTION

(a) Philosophy

The goal of this package is to provide a toolkit that facilitates efficient and reproducible analysis of longitudinal hormone data commonly used by wildlife endocrinologists. With that in mind, we created functions that perform routine characterization methods (e.g. iterative baseline and AUC calculations), as well as a suite of data visualization functions to facilitate graphical analysis.

To encourage researchers who are less familiar with *R* to use these functions, we developed an *R*-minimal workflow which allows users with no prior *R* experience to be able to run the functions. To this end, we created a detailed manual that includes instructions on how to install *R*, load the *hormLong* package, and prepare data, in addition to detailed explanations and examples of each function. We also developed an *R* script template that can be easily modified for analysis of a researcher's own data, eliminating most *R* coding (manual is located at http://hormlong.weebly.com and the package is available on GitHub at https://github.com/bfanson/hormLong). Output files are in *csv* and *pdf* format. *csv* files can be used in any spreadsheet or statistical software (e.g. Excel, SPSS, JMP) and *pdf* files can be opened in vector-graphics programs (e.g. Illustrator, Inkscape) and modified easily for manuscripts.

(b) Typical workflow

Figure 1 illustrates a standard workflow for *hormLong*. In short, data are imported and date/time formatted. Then the baseline analysis (*hormBaseline*) is run, which creates a *hormLong* object. This object can then be used for other functions that create graphs or calculate summary data. The list of current functions in *hormLong* is in Table 1.

(c) Data preparation and import

The data needs to be organized in Excel (or similar program) prior to importing to *R*. The data should be in 'long form' (i.e. one hormone concentration per row) to take advantage of grouping capabilities of *hormLong*. For example, the elephant dataset has five columns: (1) elephant name (e.g. 'Ele1', 'Ele2'), (2) date sample collected (e.g. '29-Apr-07', '01-May-07'), (3) hormone type (e.g. 'Progesterone', 'Cortisol'), (4) hormone concentration (e.g. 0.34, 0.28), (5) name of an event (e.g. 'mated', 'ovulated'). At a minimum, the dataset must have animal identifier, date collected (or numeric days), and hormone concentration. Please see manual for detailed examples. The data must be saved as a *csv* file.

Once data are suitably prepared, the *csv* file can be imported into *R* using the function *hormRead()*. If dates and/or times are part of the dataset, the function *hormDate()* handles formatting of these variables so they are compatible with all *hormLong* functions.

Example code for import and date formatting:

```
hormElephant = hormRead()
hormElephant = hormDate(data     = hormElephant,
                        date_var = 'Date_collected',
                        name     = 'Date')
```

(d) Baseline Analysis

The iterative baseline calculation is a common method used for detecting peaks in longitudinal datasets (Brown et al. 1996; Clifton & Steiner 1983). In this method, the mean and standard deviation (SD) are calculated for the dataset. Any values that are greater than

PeerJ PrePrints | https://dx.doi.org/10.7287/peerj.preprints.1546v1 | CC-BY 4.0 Open Access | rec: 30 Nov 2015, publ: 30 Nov 2015

the cutoff value (determined as the *mean + (n * SD)*) are removed, and this process is repeated until no values exceed the cutoff. Values remaining at the end of this process are considered “baseline”, whereas those that have been excluded are classified as “peaks”.

The *hormBaseline()* function allows users to easily run these iterative calculations using a single line of code. This function can run separate baseline calculations for multiple groups (e.g., individuals, species, and/or hormones) at the same time because it allows the user to define the grouping of the hormone data using the *by_var* argument. For instance, *by_var*=’species, id’ would perform separate calculations for each individual for each species. The function returns a *hormLong* object that is used as the basis of the other functions described below. The ease of performing these calculations makes it much faster to adjust criteria and identify an appropriate cutoff criteria for your dataset. If the criteria is too conservative (i.e., high value of *n*), then it is less likely to identify any peaks. Conversely, if the criteria is too low then it may result in the majority of the values being classified as “peaks”.

For the elephant dataset, we ran *hormBaseline()* in order to identify peaks in the cortisol and progesterone data. We wanted to calculate a separate baseline for each individual elephant and each hormone, so we included *by_var*=‘Ele, Hormone’, where ‘Ele’ is the column name containing the elephant’s identifier. We tested 3 different baseline cutoff criteria in order to identify an appropriate criteria for our dataset: (1) mean + 1.5 SD, (2) mean + 2 SD, and (3) mean + 3 SD (Figure 2). For this dataset, the first criteria is too liberal and consequently nearly all the values are identified as peaks, which is not useful (Figure 2A). On the other hand, the third criteria is too strict and no points were identified as peaks (Figure 2C). For this dataset, we decided to use a criteria of 2 SD (Figure 2B). The *hormBaseline()* function produces an object (called “*result*” in the example code below) that can then be graphed to visualize the calculated baseline cutoff for each elephant.

Example code for mean + 1.5 SD:

```
result =  hormBaseline(data         = hormElephant,
                       by_var       = ‘Ele, Hormone’,
                       conc_var     = ‘Cong_ng_ml’,
                       time_var     = ‘Date’,
                       event_var    = ‘Event’,
                       criteria     = 1.5)
```

(e) Data Visualization

Data visualization is an essential component of identifying patterns in longitudinal hormone profiles. To facilitate this process, we have developed several plotting functions. The *hormPlot()* function is the basic plotting function that creates longitudinal profiles, broken up according to the *by_var* statement and plotted with the baseline cutoff. Specific events (e.g. mating, parturition, stressor) can be plotted onto profile graphs by adding an event column into the user's dataset prior to import. If large temporal gaps exist in the data, *hormPlotBreaks()* can be used remove those gaps. When considering multiple hormones, *hormPlotOverlap()* overlays multiple hormone profiles, and *hormPlotRatio()* plots the ratio of two specified hormones. In order to visualize differences in the distribution of multiple groups, *hormBoxPlot()* creates vertical boxplots for all groups specified. All plots are exported as *pdf* files and have several formatting options (e.g. plot size, number of plots per page, date format, setting all x-axes/y-axes to the same range).

For the elephant dataset, we ran *hormPlot()* to visualize the longitudinal plots with three different baseline cutoff criteria (see above; Figure 2). This produced longitudinal plots for each elephant with a reference line showing the baseline cutoff and arrows indicating all events. Next, we wanted to overlay cortisol and progesterone plots (Figure 3A). This allowed us to identify when cortisol peaks occurred relative to progesterone peaks. Using this function, it was clear that peaks in cortisol predominantly occurred during the follicular phase, just before progesterone began to increase.

Example code for longitudinal plots with baseline cutoff:

```
hormPlot(result)
```

Example code for overlaying cortisol and progesterone plots:

```
hormPlotOverlap(result,
                    hormone_var='Hormone',
                    colors='green, purple' )
```

(f) Summary Statistics

After identifying peaks using baseline criteria, it is often necessary to extract summary statistics from longitudinal profiles for subsequent analyses (e.g. ANOVA in the user's preferred statistical software). The function *hormSumTable()* exports summary statistics into a *csv* file for this purpose. For the elephant data, the exported summary statistics are shown in Table 2.

Alternatively, the user may want run a statistical analysis (e.g. linear mixed model) on the original dataset, but need each sample identified as ‘baseline’ or ‘peak’, as determined from the iterative baseline method. This can be achieved by including *save_date*=TRUE in *hormBaseline*() and a *csv* file will be created.

Example code for obtaining summary statistics:

```
hormSumTable(result)
```

(g) Area Under the Curve Analysis

Area under the curve (AUC) is often used to calculate the magnitude of a response. The *hormArea()* function performs this calculation using the following algorithm: 1) for subsequent time points, determine whether the line crosses the lower bound cutoff threshold (see below for options); 2) if it does cross, calculate the time at which the line crosses the cutoff threshold; 3) using these new end time points, calculate the AUC (see below for calculation methods). As with baseline calculations, AUC can be calculated for multiple groups in a single step using the *by_var* statement.

Three different lower bounds can be used for AUC calculations: 1) area from the x-axis (‘origin’); 2) area from the baseline mean (‘baseline’); or 3) area from peak cutoff value determined from *hormBaseline()* (‘peak’). For each scenario, *hormArea()* calculates the area above the reference line and counts the number of discrete peaks. Therefore, in the origin scenario, the entire profile constitutes a single peak. Users can also choose between two commonly used calculation methods: 1) trapezoid method $[\sum \frac{1}{2} * (t_i - t_{i+1}) * [(c_i + c_{i+1}) - cutoff]]$; or 2) spline [integrating over *spline(method=’natural)* from *stats* package in *R*] (Adams et al. 2011; Cockrem & Silverin 2002; Littin & Cockrem 2001). After calculating AUC for each peak, the function produces a summary table that includes each peak identity with its corresponding AUC value. Longitudinal plots of the peak AUCs are also produced (Figure 3B), allowing the user to match up peak identity in table with specific points on the plot and, especially for the spline method, to assess the appropriateness of the fit.

For the elephant dataset, we ran *hormArea()* to quantify the area of each cortisol peak in each longitudinal profile (Figure 3B). This allows for comparisons of the magnitude of cortisol peaks across cycles or among individuals.

Example code for obtaining summary statistics:

```
hormArea(result, lower_bound = 'peak')
```

CONCLUSIONS

hormLong is an *R* package tailored to the analysis of longitudinal hormone data in wildlife endocrinology studies. This package provides an efficient and easy method for implementing the iterative baseline approach and calculating AUC for a large number of individuals. Furthermore, the graphical capabilities of this package greatly reduce the time-consuming process of graph creation, producing searchable *pdf* files with separate profiles for each individual in seconds. We have simplified the *R* code so that minimal *R* experience is required by the user, with all results exported from the *R* environment to allow the user to use other software when preferred. We hope that wide-spread adoption of *hormLong* will result in more reproducible hormone analysis and comparable results. The manual can be downloaded from http://hormlong.weebly.com and the package is available on GitHub at https://github.com/bfanson/hormLong.

ACKNOWLEDGEMENTS

We would like to thank the members of the International Society of Wildlife Endocrinology who provided valuable suggestions for the development of this package. We would also like to thank the people at the Smithsonian Conservation Biology Institute, particularly Dr. Katie Edwards, who helped test this package and provided valuable feedback.

REFERENCES

Adams NJ, Farnworth MJ, Rickett J, Parker KA, and Cockrem JF. 2011. Behavioural and corticosterone responses to capture and confinement of wild blackbirds (Turdus merula). *Applied Animal Behaviour Science* 134:246-255. http://dx.doi.org/10.1016/j.applanim.2011.07.001

Brown JL, Wildt DE, Wielebnowski N, Goodrowe KL, Graham LH, Wells S, and Howard JG. 1996. Reproductive activity in captive female cheetahs (*Acinoyx jubatus*) assessed by faecal steroids. *Journal of Reproduction and Fertility* 106:337-346.

Clifton DK, and Steiner RA. 1983. Cycle Detection: A Technique for Estimating the Frequency and Amplitude of Episodic Fluctuations inBlood Hormone and Substrate Concentrations. *Endocrinology* 112:1057-1064. doi:10.1210/endo-112-3-1057

Cockrem JF, and Silverin B. 2002. Variation within and between Birds in Corticosterone Responses of Great Tits (Parus major). *General and Comparative Endocrinology* 125:197-206. http://dx.doi.org/10.1006/gcen.2001.7750

Cowpertwait PS, and Metcalfe AV. 2009. *Introductory time series with R*: Springer Science & Business Media.

Fanson KV, Keeley T, and Fanson BG. 2014. Cyclic changes in cortisol across the estrous cycle in parous and nulliparous Asian elephants. *Endocrine connections* 3:57-66.

Littin K, and Cockrem J. 2001. Individual variation in corticosterone secretion in laying hens. *British Poultry Science* 42:536-546.

Montgomery DC, Jennings CL, and Kulahci M. 2015. *Introduction to time series analysis and forecasting*: John Wiley & Sons.

Sheriff MJ, Krebs CJ, and Boonstra R. 2010. Assessing stress in animal populations: Do fecal and plasma glucocorticoids tell the same story? *General and Comparative Endocrinology* 166:614-619. 10.1016/j.ygcen.2009.12.017

Table 1: List of functions in *hormLong*.

Type	Name	Description
Import and data handling	*hormRead()*	Provides a pop-up window to import file
	hormDate()	Converts character date (e.g. “2014-01-01”,’01-January-2014’) to numeric date field. If a time column (‘18:10:01’) is also supplied, then a date-time field is created.
Analysis	*hormBaseline()*	Main function that calculates peak cutoff value using iterative algorithm. Produces a *hormLong* object that is used for most other functions
	hormSumTable()	Calculates basic statistics for hormone data, such as mean, min, max, baseline mean, %CV
	hormArea()	Calculates area under the curve (AUC) for all peaks
Visualization	*hormPlot()*	Produces longitudinal plots of hormone profiles for each group specified in *by_var*. Includes baseline cutoff and individual specific events
	hormPlotBreaks()	Similar to *hormPlot()*, except that temporal gaps in endocrine profiles are removed.
	hormPlotOverlap()	Produces longitudinal plots in which multiple hormone are overlaid.
	hormArea()	Produces longitudinal plots in which AUC for peaks are delineated and numbered. This plot complements hormAUC analysis table so that numbered peaks can be assessed visually.
	hormBoxplot()	Produces simple boxplots comparing hormone concentrations using grouping function *by_var*.

Table 2: Example output for *hormSumTable()*. Base_mean is the mean of baseline values from iterative process. Peak_mean is mean of all peak values. Cutoff is the cutoff threshold (mean + (*n* * SD) determined from *hormBaseline()*. Other statistics are based on all hormone values.

Ele	Hormone	mean	median	sd	percent_cv	min	max	cutoff	base_mean	peak_mean	peak_base
Ele1	Cortisol	0.83	0.7	0.51	61.62	0	2.49	1.09	0.61	1.62	2.67
Ele1	Progesterone	0.41	0.36	0.37	89.72	0	1.31	0.94	0.34	1.13	3.36
Ele2	Cortisol	0.62	0.46	0.52	84.72	0.19	2.84	0.66	0.42	1.42	3.41
Ele2	Progesterone	0.85	0.82	0.52	61.85	0.05	2.77	1.66	0.78	2.15	2.75

mean	average (of all points for that set of grouping variables)
median	median (of all points for that set of grouping variables)
sd	standard deviation (of all points for that set of grouping variables)
percent_cv	percent coefficient of variation (SD/mean*100)
min, max	minimum and maximum values (of all points for that set of grouping variables)
cutoff	threshold value for peaks, calculated as mean+(*n**SD) for final iteration of baseline calculation (i.e., when no more points are removed). Points below this are baseline and above are peaks.
base_mean	average of all points classified as baseline
peak_mean	average of all points classified as peaks
peak_base	ratio of peak-to-baseline (calculated as peak_mean/base_mean)

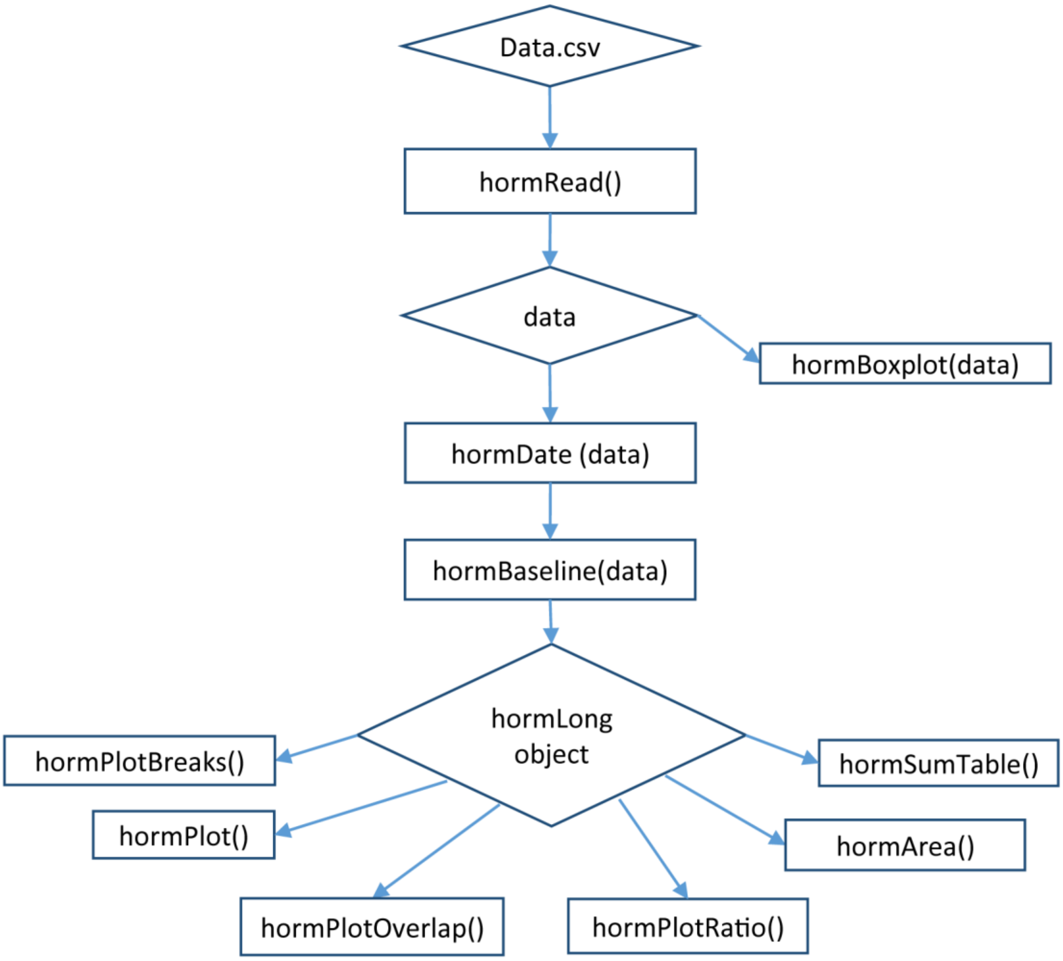

Figure 1: Flowchart of a typical *hormLong* analysis. Diamonds show *R* objects and boxes are functions.

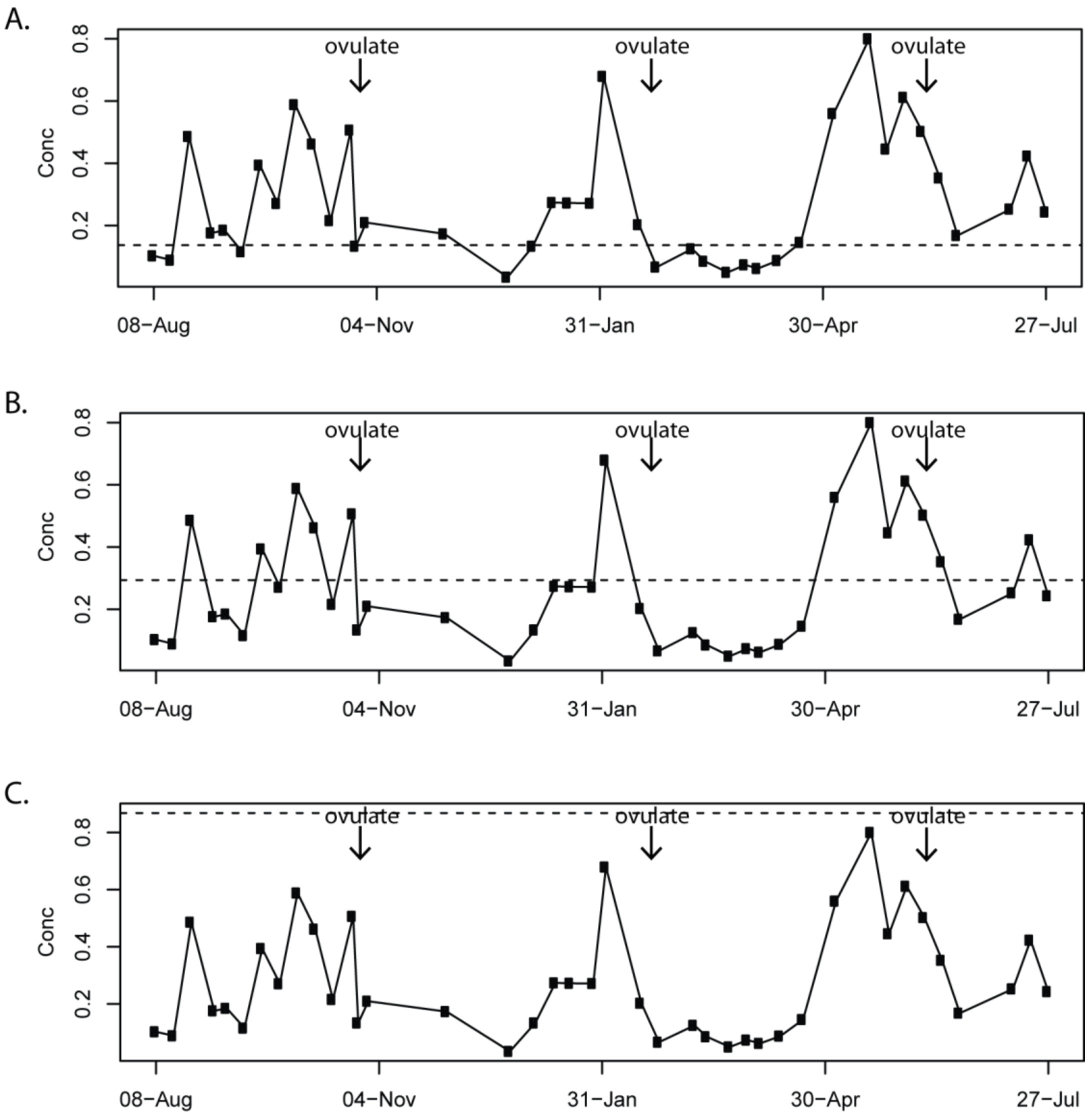

Figure 2: Example of *hormPlot()* with varying criteria for a single individual. The dashed line represents the cutoff criteria: A) mean + 1.5, B) mean + 2, and C) mean + 3.0. Arrows and text show the occurrence of an event.

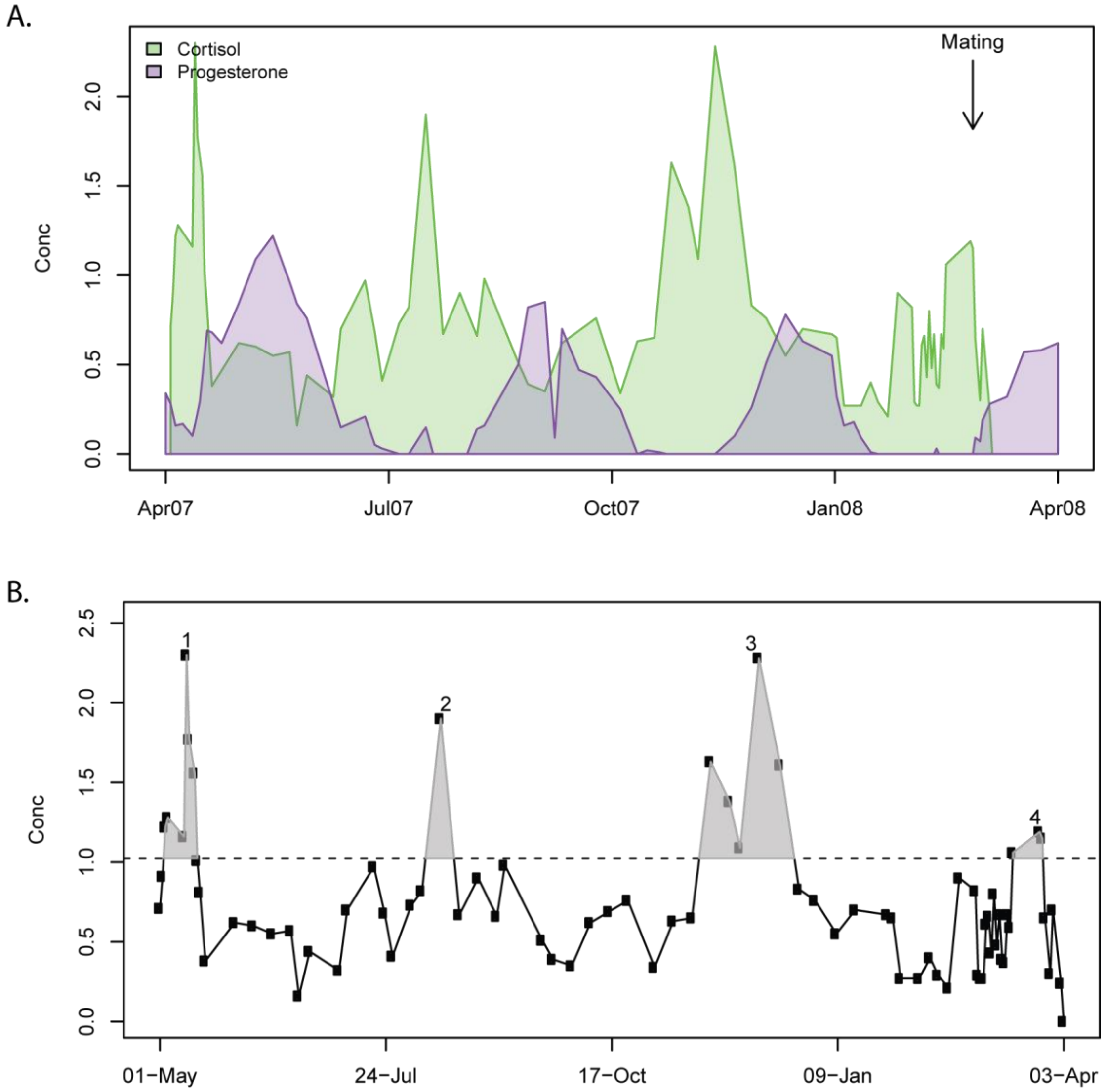

Figure 3: Example of (A) *hormPlotOverlap()*and (B) *hormArea()* plot. For (A), the different colours represent cortisol (green) and progesterone (purple). For (B), numbers indicate discrete peak number (matches up with outputted table) and shaded area shows the AUC calculated in the output data table. Dashed lines is the baseline cutoff value (note – other cutoff criteria can be used for *hormArea()*, see manual).