Marine Biology

Environmental cycles and individual variation in the vertical movements of a benthic elasmobranch

Edward Lavender[1,2*], Dmitry Aleynik[3], Jane Dodd[4], Janine Illian[5], Mark James[2], Peter J. Wright[6], Sophie Smout[1,2], James Thorburn[2]

[1] Centre for Research into Ecological and Environmental Modelling, University of St Andrews, St Andrews, Fife, Scotland

[2] Scottish Oceans Institute, University of St Andrews, St Andrews, Fife, Scotland

[3] The Scottish Association for Marine Science, Oban, Argyll, Scotland

[4] NatureScot, Oban, Argyll, Scotland

[5] School of Mathematics and Statistics, University of Glasgow, Glasgow, Lanarkshire, Scotland

[6] Marine Scotland Science, Aberdeen, Aberdeenshire, Scotland

Corresponding author: el72@st-andrews.ac.uk

Supplementary Information: Methods

1. Data processing

Data around capture events were removed to focus on undisturbed vertical movements in a two-stage process:

A. For tag deployment and recovery events, all observations from 0600 hours until midnight on the day of (re)capture were removed.
B. Captures recorded by recreational anglers were identified from a capture-recapture database maintained by NatureScot, Marine Scotland Science and the Scottish Association for Marine Science. The precise time of these events was defined from peaks in the vertical movement and temperature time series using a semi-automatic approach implemented by define_recapture function in Tools4ETS package (Lavender 2020). All data from two hours before each event until midnight on the day of capture were removed.

A potential issue with this approach is the removal of differing amounts of data for different individuals. However, in this study, as all analysed skate were captured at similar times (in the afternoon), the amount of data excluded following a capture event (median = 9.33 hours) was broadly similar (median absolute deviation = 1.21 hours). Furthermore, while recent work has indicated rare examples of irregular post-release behaviour in flapper skate, such changes are short-lived and should be accounted for within this time frame (Lavender et al. in review). In contrast, as indicated by previous research (Thorburn et al. 2021) and the data exploration in this manuscript, other variables such as diel cycles seem to be more important drivers of vertical movement for flapper skate, with substantial movements into shallower water often apparent at night. In this situation, the 'standard' data processing approach that excludes the first *n* hours of post-release data is potentially problematic because the loss of data is non-random with respect to individual behaviour. In contrast, the removal of all post-release data until midnight on the day of capture should exclude any irregular post-release behaviour(s) from the time series while respecting the observation that individuals often move into shallower water at night. Nevertheless, it is important to note that the impact of data processing is likely to be overwhelmed by the rest of the data in each individual's time series (45–772, median = 216, days in duration among modelled individuals), so the results are robust to this choice.

Another potential issue with this workflow is that it does not account for unrecorded capture events (from angling vessels or the shore) that may have occurred during individuals' time at liberty. For the skate analysed in this study, recent work has indicated that unrecorded, vessel-based catch-and-release angling events, if they occurred, were extremely rare (Lavender et al. in review). However, the possibility of shore-based angling—which may leave distinct signatures in the depth time series that are not clearly discernible as capture events without additional information—cannot be excluded (Lavender et al. in review). Nevertheless, as above, the influence of these events (if they occurred) is likely to be minimal in the context of the rest of each individual's time series.

2. Functional principal components analysis (FPCA) for observed time series

FPCA was used to examine similarities and differences in the shape and structure of observed depth time series, for each month from March 2016 until February 2017, using the protocol shown below for an example month:

```
#### Define observations
# For each time window, all of the observations from that month
# ... were identified. In order to make the time series comparable,
# ... given breaks in some time series associated
# ... with (re)capture events and variable times at liberty,
# ... only individuals with the most common number of observations
# ... in that month were included in the analysis.
# ... These data were coerced into a matrix, with
# ... one row for each depth observation (two-minute resolution)
# ... and one column for each individual. This matrix is named 'mat'.
# ... FPDA was then performed on this matrix, as shown below.

#### Make B splines
# 1,000 basis functions sufficient to represent
# ... complex time series effectively
n_basis    <- 1000
rng        <- 1:nrow(mat)
bs_splines <- fda::create.bspline.basis(range(rng), n_basis)

#### Create a functional data object using the matrix and B-spline basis
fda_df <- fda::Data2fd(mat, argvals = rng, basisobj = bs_splines)

#### Compare smooths and observations
# Extract necessary objects
fdobj  <- fda_df
y      <- rng
Lfdobj <- fda::int2Lfd(0)
fdmat  <- fda::eval.fd(y, fdobj, Lfdobj)
# Plot first time series and smooth as an example
plot(y, fdmat[, 1], col = "red")
lines(y, mat[, 1])

#### Implement FPCA
pcs_1 <- fda::pca.fd(fda_df)

#### Visualise FPCA scores
plot(pcs_1$scores)
```

3. Depth models

3.1 Model fitting

Depth (depth) was modelled in relation to tidal height (th_interp), sun angle (sun_angle), lunar phase (lunar_phase), photoperiod (photoperiod) and photoperiod direction (photoperiod_direction) for each individual (dst_id), according to the following protocol shown for individual 1547:

```r
#### Flag independent time series
# Here, 'arc_1547' is the observed time series for DST ID 1547
# ... and 'date_time' distinguishes time stamps.
# ... Serially independent time series are defined as those
# ... with breaks longer than the duration between regular archival
# ... observations (i.e., due to the removal of data around
# ... recapture events).
flags <- Tools4ETS::flag_ts(x = arc_1547$date_time,
                            duration_threshold = 2)
# Add flags to dataframe:
arc_1547$start_event    <- flags$flag1
arc_1547$start_event_id <- flags$flag3

#### Thin time series, accounting for breaks
# Here, we thin the time series by n = 30.
# ... This degree of thinning is sufficiently large to reduce autocorrelation
# ... while sufficiently small not to influence model smooths.
# ... Further, models fitted to datasets thinned by lesser/greater amounts
# ... confirm that the results are robust to this choice.
arc_1547 <-
  Tools4ETS::thin_ts(dat = arc_1547,
                     ind = "start_event_id",
                     flag1 = "start_event",
                     first = 1,
                     nth = 30)

#### Define model formula
# For tidal height, sun angle and photoperiod, cubic regression splines
# ... are used as these are more computationally efficient than the default
# ... thin plate regression splines. For lunar phase, a cyclic spline
# ... is required.
fdriver <-
  formula(depth ~
            photoperiod_direction +
            s(th_interp, bs = "cr", k = 10) +
            te(sun_angle, lunar_phase,
               bs = c("cr", "cc"), k = c(5, 5)) +
            te(sun_angle, photoperiod,
               by = photoperiod_direction,
               bs = c("cr", "cr"), k = c(5, 5))
          )
```

```r
fdriver_knots <- list(lunar_phase = c(0, 2*pi))

#### Model fitting
# The models are fitted using mgcv::bam(),
# ... which is optimised for large datasets.
# This function can incorporate an AR1 correlation structure
# ... for models with a Gaussian likelihood, but the AR1 parameter
# ... must be specified: it is not estimated internally, unlike
# ... other commonly used routines, such as mgcv::gamm(). Therefore,
# ... we will implement the model without an AR1 structure, use this model
# ... to estimate the AR1 parameter, and then re-implement the model
# ... using the estimated AR1 parameter.

## Fit model without AR1 parameter
dat_mod <- arc_1547
mod_ar0 <- mgcv::bam(fdriver,
                     gamma = 1.4,
                     knots = fdriver_knots,
                     data = dat_mod)

## Estimate AR1 parameter
ar1     <- Tools4ETS::estimate_AR1(resid(mod_ar0),
                                   AR.start = dat_mod$start_event,
                                   verbose = FALSE)
## Fit model with AR1 parameter
mod_ar1 <- mgcv::bam(fdriver,
                     gamma = 1.4,
                     knots = fdriver_knots,
                     rho = ar1,
                     AR.start = dat_mod$start_event,
                     data = dat_mod)
```

Model smooths, predictions and diagnostics were examined using standard mgcv functions, posterior simulation and FPCA (see below). This process was repeated for each individual. For most individuals, there were some issues with model diagnostics (e.g., see Fig. S6) but these were not improved by alternative model formulations.

3.2 Posterior simulation for each individual

For each individual, posterior simulation was used to estimate the magnitude of covariate effects. In this process, values of the response (depth) were simulated from the ‘$AR1_{DP}$ model’ for low and high values of variables of interest (e.g., tidal height), using the simulate_posterior_mu function in the Tools4ETS package. For each variable of interest, this process generated two ‘posterior distributions’ for depth (e.g., one for low tidal height and one for high tidal height), from which a ‘posterior distribution’ of differences in depth was calculated. The median value of this distribution was taken as an estimate of effect size and the median absolute deviation was taken as a measure of uncertainty. This approach was applied to obtain estimates for the following effects:

- **The effect of tidal height.** Depths were simulated and compared for tidal height = 0.13 versus 4.53 m (the minimum and maximum tidal height experienced in the study area), with sun angle = 0.00°, lunar phase = 0π radians, photoperiod = 13.22 hours and photoperiod direction = 'lengthening'.
- **The effect of sun angle when the photoperiod was short.** Depths were simulated and compared for sun angle = -40.41° versus sun angle = 26.93° (the minimum and maximum sun angle experienced near the start of the year, on 2016-03-17, by which time all but two of the individuals with data were tagged and at a photoperiod experienced by all individuals), with tidal height = 2.50 m, lunar phase = 0π radians, photoperiod = 13.22 hours (the photoperiod on 2016-03-17) and photoperiod direction = 'lengthening'.
- **The effect of sun angle when the photoperiod was long.** Depths were simulated and compared as above but using with sun angle = -10.14° versus sun angle = 57.01° and photoperiod = 19.89 hours, the values for the summer solstice (2016-06-20). Note that this required limited extrapolation beyond the maximum photoperiod experienced for individuals 1507, 1523, 1539, 1558 and 1574.
- **The effect of lunar phase during the day.** Depths were simulated and compared for lunar phase = 0π (new moon) versus lunar phase = π radians (full moon), with tidal height = 2.50 m, sun angle = 25.00° (a daytime value broadly applicable throughout the year), photoperiod = 13.22 hours and photoperiod direction = 'lengthening'.
- **The effect of lunar phase at night.** Depths were simulated and compared as above but with sun angle = -25.00°.
- **The effect of photoperiod when days were lengthening.** Depths were simulated and compared for photoperiod = 13.22 hours versus photoperiod = 19.89 hours, with tidal height = 2.50 m, sun angle = 0.00°, lunar phase = 0π radians and photoperiod direction = 'lengthening'.
- **The effect of photoperiod when days were shortening.** Depths were simulated and compared as above, for individuals with sufficient data, but with photoperiod direction = 'shortening'.

As an example, the protocol used to estimate the statistical effect of tidal height on depth for a given individual is shown below.

```
# Define 'newdata' for which to simulate values of the response (depth)
nd <- data.frame(th_interp = c(0.13, 4.53),
                 sun_angle = 0,
                 lunar_phase = 0,
                 photoperiod = 13.21528,
                 photoperiod_direction = "lengthening")

# Simulate values of the response (depth) from the model
# ... Here, 'mod' the 'AR1DP model' for a specific individual
# ... 'nd' is the newdata dataframe defined above.
post <- Tools4ETS::simulate_posterior_mu(model = mod,
                                         newdata = nd,
                                         return = "full")
```

```r
# Define the posterior distribution of differences
post_diff <- post[2, ] - post[1, ]

# Summarise the posterior distribution of differences in depth
# ... as a measure of 'effect size'.
# The median difference in simulated depths between low/high tidal heights
stats::median(post_diff)
# The median absolute deviation of the differences
stats::mad(post_diff)
```

However, rather than focusing on estimated effect sizes for each individual, the purpose of this exercise was to generate posterior distributions that could be pooled across individuals, to provide an indicator of 'average' effect size to support graphical interpretation of Fig 4 (see §3.3, below).

3.3 Posterior simulation across individuals

To estimate the average effect size across all individuals, simulated outcomes from posterior simulation were aggregated across individuals, with the following function, and then summarised.

```r
#' @title Simulate the posterior distribution of differences between two
values
#' @description This function calculates the posterior distribution of
differences in simulated outcomes between two values of a variable, holding
other variables constant.
#' @param mods A list of models (one for each individual).
#' @param newdata A 2 x n dataframe that defines the values of the model
covariates at which predictions are required.
#' @return The function returns a numeric vector of differences.

get_posterior_difference <- function(mods, newdata){
  # Create a list of posterior distributions (for each model)
  post <- lapply(mods_depth_ar1, function(mod_ar1){
    # Simulate from the posterior of the model, given newdata
    Tools4ETS::simulate_posterior_mu(model = mod_ar1,
                                     newdata = newdata,
                                     seed = 1L,
                                     return = "full")
  })
  # Aggregate posterior distributions across individuals
  post <- do.call(cbind, post)
  # Return the distribution of differences between the first two rows
  post_diff <- post[2, ] - post[1, ]
  return(post_diff)
}
```

For example, to summarise the distribution of differences in the expected depth between low and high tide, the following approach was implemented:

```r
#### E.g. Simulate depths at low versus high tide
# ... setting sun angle to 0
# ... lunar phase to 0 (new moon)
# ... photoperiod to ~ 13.21 (hours of daylight on 2016-03-17)
# ... photoperiod_direction to 'lengthening' (increasing photoperiod)
nd <- data.frame(th_interp = c(0.13, 4.53),
                 sun_angle = 0,
                 lunar_phase = 0,
                 photoperiod = 13.21528,
                 photoperiod_direction = "lengthening")
post <- get_posterior_difference(mods = mods_depth_ar1, newdata = nd)

#### Examine frequency distribution of differences
graphics::hist(post)
```

The results of this process were taken to provide an ‘indicator’ of the average effect size across individuals to support graphical interpretation of model predictions.

3.4 FPCA of model smooths

FPCA was used to examine how estimated smooths varied in shape, by representing smooths as B-splines and then applying FPCA to those functions, as demonstrated below for tidal height:

```r
#### Define data for prediction
# Define a regular sequence of tidal heights across the range for that variable
# Hold other variables constant at appropriate values
nd <- data.frame(th_interp = seq(0.13, 4.53, length.out = 100),
                 sun_angle = 0,
                 lunar_phase = 0,
                 photoperiod = 13.21528,
                 photoperiod_direction = "lengthening")

#### Define centred predictions for each individual in a matrix
# Loop over a named list of model objects ('mods_depth_ar1')...
pred_by_id <- lapply(mods_depth_ar1, function(mod_ar1) {
  # Make predictions from nd
  pred <- predict(mod_ar1, newdata = nd)
  # Centre predictions to focus on the shapes of the smooths
  pred <- as.numeric(pred)
  pred <- pred - mean(pred)
  return(pred)
  })
# Convert to matrix
mat <- do.call(cbind, pred_by_id)
colnames(mat) <- names(mods_depth_ar1)


```

```r
#### Make B-splines
# 10 basis functions sufficient to represent smooths effectively
rng <- 1:nrow(mat)
bs_splines <- fda::create.bspline.basis(range(rng), 10)

#### Create a functional data object
fda_df <- fda::Data2fd(mat, argvals = rng, basisobj = bs_splines)

#### Compare functions with smooths in FDA
# As shown previously.

#### Implement FPCA
pcs_1 <- fda::pca.fd(fda_df)
```

4. Vertical activity models

Following the protocol developed above, the total vertical activity per hour (va) of each individual was modelled similarly in relation to day/night (diel), lunar phase, photoperiod and photoperiod direction, as shown below for individual 1547.

```r
#### Define model formula
# ... The response, VA, is va/diel_duration, where
# ... 'va' is the total (absolute) vertical activity in each diel period
# ... (i.e., day 1, night 1, day 2,..., day n, night n)
# ... and 'diel_duration' is the duration of that period (hours).
# ... Note that since VA is expressed in this way,
# ... there is no need to thin the time series prior to modelling,
# ... unlike in the depth time series models.
fdriver <- formula(va/diel_duration ~
                     diel +
                     s(lunar_phase,
                       bs = "cc", by = diel, k = 10) +
                     s(photoperiod,
                       by = interaction(diel, photoperiod),  k = 10)
                     )
fdriver_knots <- list(lunar_phase = c(0, 2*pi))

#### Fit model without AR1 parameter
# ... Use 'dat_mod', a dataframe with the variables defined above
# ... for individual 1547
mod_ar0 <- mgcv::bam(fdriver,
                       gamma = 1.4,
                       knots = fdriver_knots,
                       data = dat_mod)

#### Estimate AR1 parameter
ar1     <- Tools4ETS::estimate_AR1(resid(mod_ar0),
                                   AR.start = dat_mod$start_event,
                                   verbose = FALSE)
```

```r
#### Fit model with AR1 parameter
mod_ar1 <- mgcv::bam(fdriver,
                     gamma = 1.4,
                     knots = fdriver_knots,
                     rho = ar1,
                     AR.start = dat_mod$start_event,
                     data = dat_mod)
```

As for the depth time series, models were fitted to each individual and analysed using standard mgcv functions, FPCA and posterior simulation. Model diagnostics for these models were acceptable (e.g., see Fig. S10).

References

Lavender E (2020) Tools4ETS: Tools for ecological time series. R package version 0.1.0. http://github.com/edwardlavender/Tools4ETS

Thorburn J, Wright PJ, Lavender E, Dodd J, Neat F, Martin JCA, Lynam C, James M (2021) Seasonal and ontogenetic variation in depth use by a Critically Endangered benthic elasmobranch and its implications for spatial management. Front Mar Sci 8: 656368. https://doi.org/10.3389/fmars.2021.656368