Setting up data for analysis with secr

Gather SECR data

SECR surveys use detectors at fixed locations to record the presence of individually identifiable animals at those locations. Detectors can be camera-traps, hair snares and dung surveys, live-captures, or acoustic detectors.

Set up data

The R package `secr` provides methods for estimating animal abundance from SECR data under many different conditions. This sheet summarizes getting your data into the format `secr` wants.

Analyse data

Once the data has been set up, use it to build SECR models and extract results on animal abundance, detectability, and important covariates.

(1) Make the detector file

Each line of `trapfile` contains the location of each detector (e.g. camera), plus any extra information about that detector.

TrapID	X	Y	Effort	/	tri	temp
A1	0	0	10 20	/	0.6	25
A2	5	0	10 19	/	0.9	23
A3	0	5	0 20	/	0.8	31

1. TrapID, X, and Y must be specified in the order given. X and Y contain the detector locations.
2. Effort records length of time each detector recorded for (optional). One value per occasion, separated by white space
3. Any other variables record covariates at the detectors (optional). These are stored to the right of the "/" column (also optional).

- Detector covariates only used if detection function parameters vary across traps (`g0, lambda0, sigma`).
- If using multiple sessions with detector changes between sessions, need one trapfile per session (see below).
- Save as a .txt file to read into R later (.csv and .xlsx options also available).
- Header row should begin with a # if saving as .txt

(2) Make the capture history file

Each line of `captfile` contains one detection, with ID variables recordings information about that detection.

Session	Animal	Occasion	TrapID
1	z001	1	A2
1	z174	2	A1
1	z024	1	A1

1. Each detection is recorded as a session identifier, animal identifier, occasion identifier.
2. Each detection includes a detector identifier, either as trapID (as above) or as X- and Y-coordinates (replace trapID with two columns X, and Y)

- Session and occasion columns required even if you only use one session or occasion.
- Occasion must be an integer starting from 1.
- Save as a .txt file with header row starting with # (.csv and .xlsx options also available)

(3) Read it all in

Load both your `trapfile` and `captfile` files with `read.capthist`.

```
ch <- read.capthist(captfile = "ch.txt",
  trapfile = "tf.txt", detector = "count",
  fmt = "trapID", trapcovnames = c("tri",
  "temp"), binary.usage = FALSE)
```

Important options

- captfile, trapfile - the files made in the previous steps.
- detector - specifies the type of detector you have. Most camera trap surveys will use "multi", "proximity" or "count".
- fmt - if trapID used as detector identifier in `captfile` then `fmt = "trapID"`. If X and Y used then `fmt = "XY"`.
- trapcovnames - names of covariates in `trapfile`
- binary.usage - indicates if continuous effort variable present.

(4) Make the habitat mask

A mask is a **set of square grid cells** representing habitat **in the vicinity of detectors** that is **potentially occupied**.

A **mask object** is a 2-column dataframe, each row gives the **x- and y-coordinates** of the centre of one cell.

Constructing masks from detectors with `make.mask`

```
my_traps <- traps(ch)
my_mask <- make.mask(my_traps, buffer =
 24000, spacing = 1000, type = "trapbuffer")
```

Makes a grid extending 24km N, S, E and W of any detectors

Puts mask points down at 1km intervals within the grid

Make your own mask and `read.mask`

Just 4 mask points for illustration

```
my_mask_df <- data.frame(X = c(0,1,0,1), Y
 = c(0,0,1,1), elevation = c(0,110,80,30))
my_mask <- read.mask(data = my_mask_df,
 spacing = 1)
```

Remember that `my_mask_df` must include the buffer region.

X	Y	elevation
0	0	0
1	0	110
0	1	80
1	1	30

Optional covariates

(5) Add mask covariates

Mask covariates are used to model density (`D`), not detection parameters (`g0, lambda0, sigma`).

Adding covariates from a dataframe

```
covariates(my_mask) <-
 data.frame(elevation =
 c(0,110,80,30), temp =
 c(25,26,36,37)
```

can also add covariates before `read.mask` as in the bottom box in (4)

Adding covariates from a spatial data source

Assumes you have covariates stored in a spatial data source, which can be e.g. an ESRI polygon shapefile, SpatialPolygonsDataFrame, SpatialGridDataFrame (called `spdata` below)

```
addCovariates(object = ch,
 spatialdata = spdata,
 columns = c("elevation",
 "temp"))
```

Buffers

Choose buffer width large enough that animals beyond the buffer have negligible chance of being detected.

Rough rule of thumb is buffer > 4*sigma. Can get a rough estimate of sigma with `RPSV(ch,CC=TRUE)`.

Spacing

Too few grid points means a poor approximation of likelihoods, too many points slows down model fitting.

Rough rule of thumb is spacing < 1*sigma, and try for 1000-3000 grid points.

Sessions

A session is a sampling block that is treated as independent.

Can be spatial (arrays far apart enough that no animals are detected on both) or temporal.

```
ch <- read.capthist(
  captfile="ch.csv",
  trapfile =
  c("sess1.csv",
  "sess2.csv")
my_mask <-
 make.mask(traps(ch))
```

One trap file per session. 1st file used for 1st session, 2nd file for 2nd session, etc.

Detector types

"multi" - animals can be detected at most once across all detectors in each occasion.

"proximity" - animals can be detected at most once at each detector in each occasion.

"count" - animals can be detected any number of times at each detector in each occasion.

See `?detector` for others.

secr version 4.1.0. Package created by Murray Efford. Learn more about the material shown here with the secr vignettes: secr-overview, secr-datainput, secr-tutorial, secr-habitatmasks, available at https://www.otago.ac.nz/density/SECRinR.html.