Analysing data 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. First, you need to get your data into the format `secr` wants.

Analyse data

This sheet shows you how to build SECR models and extract results on animal abundance, detectability, and important covariates.

(1) Read in SECR inputs

To build models in secr you need to have already loaded:

1 A “capthist” object, which contains the capture histories and the trap locations

2 A “mask” object, a set of grid cells that defines the area that is potentially occupied and not so far from detector locations that observations are extremely unlikely.

```
ch <- read.capthist(captfile = “ch.txt”,
  trapfile = “tf.txt”, detector = "count",
  fmt = "trapID")
my_traps <- traps(ch)
my_mask <- make.mask(my_traps, buffer =
 24000, spacing = 1000, type =
 "trapbuffer")
```

See the guide on “Setting up data” for more details

Detection models

A core SECR assumption is that detection probability (or frequency) **decreases with distance** to activity centre.

Shape is given by the **detection function** (`detectfn` in `secr.fit`), with a small number of parameters to be estimated.

Detection function models

These model the **probability** of detection. The most common option is “half-normal” (HN), with parameters `g0` and `sigma`, see `?detectfn` for others.

```
m0a <- secr.fit(ch, detectfn = "EX",
 mask = my_mask, model = list(D ~ 1,
 g0 ~ 1, sigma ~ 1))
```

“EX” function, another option

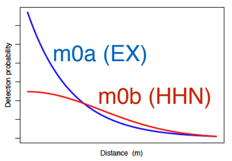
Encounter function models

These model the **expected number** of (equivalently the **hazard)** of detection. They are particularly useful for“count” detectors. The most common option is “hazard half-normal” (HHN), with parameters `lambda0` and `sigma`, see `?detectfn` for others.

```
m0b <- secr.fit(ch,
 detectfn = "HHN",
 mask = my_mask, model
 = list(D ~ 1, lambda0
 ~ 1, sigma ~ 1))
```

Plot detection functions

`plot(m0a)`



(2) Fit a model

SECR models jointly estimate two spatial models, one for animal density and one for the detection process.

Run SECR models with `secr.fit`, starting with the simplest possible model.

```
m0 <- secr.fit(ch, detectfn = "HHN",
 mask = my_mask, model = list(D ~ 1,
 lambda0 ~ 1, sigma ~ 1))
```

density, ~ 1 for constant density

Detection function parameters, `lambda0` control encounter rate, `sigma` controls range of animal movement

! ‘~ 1’ means no covariate effects, and a single parameter is estimated for each of `D, lambda0,` and `sigma`

Including covariates

Any of `D, lambda0,` and `sigma` can depend on covariates in the call to `secr.fit.`

```
m1 <- secr.fit(ch, detectfn = "HHN",
 mask = my_mask, model = list(D ~ elev,
 lambda0 ~ water, sigma ~ 1))
```

Density depends on elevation

Encounter hazard `lambda0` depends on whether detector is close to water

Very flexible e.g. can do regression splines with `D ~ s(elev)`

! Covariates on density (`D`) must be attached to the mask object, covariates on detection parameters (`g0, lambda0, sigma`) must be attached to the trap object.

```
coef(m1)
```

```
Beta parameters (coefficients)
                     beta    SE.beta         lcl        ucl
D              -9.5184241 0.27550956 -10.0584129 -8.9784353
D.elev          0.2443394 0.39160813  -0.5231985  1.0118772
lambda0        -4.4403272 0.17332682  -4.7800415 -4.1006128
lambda0.WaterYes 0.2277942 0.27803197 -0.3171385  0.7727268
sigma           8.8583684 0.08326936   8.6951634  9.0215733
```

`secr` has a number of automatically generated “canned predictors” that can be referred to directly in formulae without needing to be constructed. These include `b` (learned animal responses to detectors), `k` (site learned response), and `session`, `t` and `T` (time effects), among others.

(3) Inspect model output

To view model output use `print(m0)`

```
N animals       :  14
N detections    :  99
N occasions     :  1
Count model     :  Poisson
Mask area       :  211725 ha

Model           :  D~1 lambda0~1 sigma~1
Fixed (real)    :  none
Detection fn    :  hazard halfnormal
Distribution    :  poisson
N parameters    :  3
Log likelihood  :  -210.31
AIC             :  426.6199
AICc            :  429.0199

Beta parameters (coefficients)
             beta    SE.beta        lcl       ucl
D       -9.510329 0.26789360 -10.035391 -8.985267
lambda0 -4.387800 0.16494892  -4.711094 -4.064506
sigma    8.852195 0.08157649   8.692308  9.012082

Variance-covariance matrix of beta parameters
                    D       lambda0         sigma
D        0.0717669818 -0.0006783997 -0.0008909627
lambda0 -0.0006783997  0.0272081446 -0.0088657027
sigma   -0.0008909627 -0.0088657027  0.0066547240

Fitted (real) parameters evaluated at base levels of covariates
        link     estimate  SE.estimate          lcl          ucl
D        log 7.408266e-05 2.020773e-05 4.382129e-05 1.252414e-04
lambda0  log 1.242805e-02 2.064016e-03 8.994936e-03 1.717147e-02
sigma    log 6.989717e+03 5.711466e+02 5.956917e+03 8.201582e+03
```

`coef(m0)`

`vcov(m0)`

`predict(m0)`

Main results are in this last table. Density is in animals per hectare.

Model selection

Model selection is by AIC or AICc (small sample size)

```
AIC(m0,m0a,m0b,m1)
```

Goodness-of-fit tests are underdeveloped but see `secr.test`.

Multi-session models

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

Can run `secr.fit` as in (2). Parameters are shared between sessions by default but any of `D, lambda0,` and `sigma` can be session-specific.

```
m2 <- secr.fit(ch, detectfn = “HHN”, mask =
  my_mask, model = list(D ~ 1, lambda0 ~ 1,
  sigma ~ session))
```

Covariate effects can vary by session.

```
m3 <- secr.fit(ch, detectfn = “HHN”,
  mask = my_mask, model=list(D ~
  elev*session, lambda0 ~ 1, sigma ~
  session))
```

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

Cheatsheet by Ian Durbach