secrIWS - evaluation of spatially explicit capture-recapture with irregular data

Murray Efford

2018-07-13

Contents

The R package **secrIWS** approximately replicates the simulations of Ivan et al. (2013b) and allows the scenarios to be varied.

Introduction

Ivan et al. (2013a) proposed a density estimator that combines closed-population non-spatial capture–recapture by Huggins' (1989) conditional likelihood method with telemetry data on the activity distribution of some or all of the individuals captured (TELEM). In a simultaneous publication they evaluated the performance of the method with simulated data and compared it with existing model-based spatially explicit capture–recapture (SECR) and ad hoc boundary-strip methods (MMDM) (Ivan et al. 2013b, henceforth 'IWS'). Broadly speaking, TELEM performed very well in simulations, SECR less well, and MMDM quite badly. Weakness of the SECR method in the simulations was attributed to breaches of SECR assumptions, particularly between-animal heterogeneity and non-circularity of home ranges. Such breaches are also to be expected in real populations, so the simulations carried significance for field biologists and have been cited to explain other results, e.g., by Gerber and Parmenter (2015).

The 'secrIWS' package was written to reproduce in R the simulation results of IWS, and to further explore their interpretation. The original simulations used SAS; I am grateful to Jake Ivan for patiently answering a series of questions and providing the original code as needed for clarification.

Caveat: The package allows you to modify the scenarios of IWS in various ways. Testing has focussed on the IWS scenarios, and other combinations should be used with care.

IWS design

This section spells out the essentials of the IWS simulations that also form the basis of 'secrIWS'.

Trap layout

The IWS simulation scenarios all used a 10×10 grid of traps nominally at 10-m spacing and centred within an 'arena' 160 m square. Each trap was represented by a grid cell (width equal to trap spacing s). The arena was bounded by coordinates $(-3.5s, 12.5s)$ in each dimension and the lower-left 'trap' cell was centred at (0,0).

Home range shape

Two types of home range were modelled, either (i) a bivariate normal distribution of activity (BVN), or (ii) a set of 16 contiguous grid cells (here designated 'CELLS').

CELLS home ranges could be square (4×4), rectangular (2×8), or irregular (16 cells grown by random accretion from an initial cell); cell-specific activity was assigned from a uniform random distribution.

BVN home ranges could be circular $(\sigma_X = \sigma_Y = 0.921s)$ or elliptical $(\sigma_X = 0.4s, \sigma_Y = 1.8s$; Jake Ivan pers. comm.). In both cases the σ_X and σ_Y parameters were adjusted so that the area within the 95% activity contour matched the area of the CELLS home ranges $(16s^2)$. For a particular BVN home range, cell-specific activity was computed as the integral of the bivariate normal probability density over the area of the grid cell.

Home range placement

Placement was treated only lightly by IWS. Jake has clarified some details, allowing this description to be more specific.

The centres of BVN ranges (both circular and elliptical) were placed at random within an area extending from $(-3s, -3s)$ to $(+12s, +12s)$. This placement allows some of each utilization distribution to 'spill' outside the arena. Note, however, that the square within which centres were placed has area $225s^2$ rather than $256s^2$[1].

CELLS ranges were placed at integer positions on the grid. Rectangular CELLS ranges were placed at random so that the bottom-left cell lay within the window extending from the bottom left of the arena to the top-right cell of the trapping grid. There were $13^2 = 169$ possible locations, each with equal probability. The initial cell of irregular CELLS ranges was also selected at random from these locations. This scheme for placing CELLS home ranges ensures that all utilization for 4×4 ranges lay entirely within the arena, but 2×8 and irregular ranges could spill outside.

Trap-specific capture probability

Capture probability was modelled as a function of home range overlap with the 'trap' grid cell; overlap represents the fraction of time (activity or 'utilization') each animal is expected to spend in a cell. Capture probability was related to cell-specific activity by a variable representing overall trappability (p1 in 'secrIWS'). An animal can be caught at only one trap per sampling occasion, so it is necessary to resolve competition among traps for animals. IWS did this by a random draw from a multinomial distribution that included the possibility of non-capture along with the trap-specific probabilities.

Estimators

Ivan et al. (2013b) should be consulted for details of the estimators. These have been reproduced as faithfully as possible in 'secrIWS' using internal R code or by calling 'secr.fit' in the R package 'secr'.

TELEM was simulated with four levels for the percentage of captured animals that were telemetered (25%, 50%, 75% and 100%) and three levels of telemetry sampling (5, 10, and 20 locations per individual).

SECR used a Horvitz-Thompson estimate of density based on an observation model fitted by maximizing the integrated likelihood conditional on the number of individuals captured. The area of integration (defined by a habitat mask) was set to the area of the arena.

[1]This follows from the SAS code provided by Jake. Specifically, lines 37-38 in 'Jake Bivariate Normal Circle4.SAS' place animals in a $15s \times 15s$ arena instead of $16s \times 16s$ (need &xmax+1 for that): `xmean=ranuni(seed)*(&xmax+2*&xBorder)-&xBorder; ymean=ranuni(seed)*(&ymax+2*&yBorder)-&yBorder;`

The MMDM estimators defined the effective trapping area A_W by a convex buffer of width W around the centroids of the outermost trap cells, where W = MMDM or W = MMDM/2. The density estimate was $\hat{D} = \hat{N}/A_W$ where $\hat{N}$ was a conventional Huggins closed population estimate.

Assessment of performance

Ivan et al. (2013b) evaluated the accuracy of the TELEM, SECR and MMDM density estimators by comparing each with the 'true density' computed separately for each simulated realisation. They used a different benchmark of realisation-specific true density for each estimator, hoping to be 'fair' by using the benchmark they considered most appropriate to each (Table 1).

Table 1. Density benchmarks used by Ivan et al. (2013b)

Estimator type	'True density'	Label
TELEM	sum of home-range fractions overlying trapping grid, divided by its area	telem
SECR	number of home-range centroids on trapping grid divided by its area	inner
MMDM	number of individuals in arena divided by area of arena	total

Using 'secrIWS' to reproduce IWS

We focus on the 'intermediate' scenario of IWS: 20 individuals sampled on 7 occasions with overall capture probability 0.4.

First it is necessary to install the package. This requires the packages 'secrlinear', 'rgeos', 'sp' and 'mvtnorm'; all are available from CRAN. The package can then be loaded with

```r
library(secrIWS)
vignettefolder <- "c:/density secr 3.1/secrIWS/vignettes/"
```

Each of the five IWS scenarios for home range placement may be simulated with function `runsim`. We start by setting up the trapping grid (grid10) and defining a 'template' for each home range configuration. This works for BVN ranges and the fixed CELLS configurations (template44, template28). Irregular CELLS configurations are constructed later on the fly' with the `build.irregular` function.

```r
grid10 <- secr::make.grid(10,10, spacing = 10)
templateCircle <- templatefn(nx = 1, ny = 1,
                  sigmaX = 0.921*10, sigmaY = 0.921*10, spacing = 10)
templateEllipse <- templatefn(nx = 1, ny = 1,
                  sigmaX = 0.45*10, sigmaY = 1.8*10, spacing = 10)
template44 <- templatefn(nx = 4, ny = 4, spacing = 10)
template28 <- templatefn(nx = 2, ny = 8, spacing = 10)
```

Plotting realisations is a good way to verify the algorithm. The BVN ranges are plotted at the 95% activity contour. We assign the result to a variable to hide the numerical output.

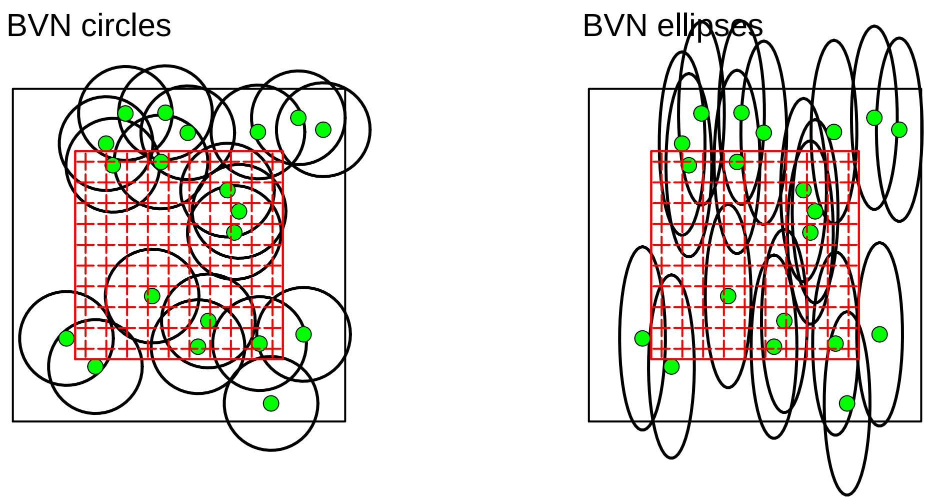
```
par(mfrow = c(1,2), pty = 's', mar = c(3,3,3,3), cex = 1, xpd = TRUE)
BVN.a <- runsim(nrepl = 1, traps = grid10, template = templateCircle, N = 20,
    IWSlowerleft = TRUE, plt = TRUE, allinside = FALSE, gridaligned = FALSE, SECR = FALSE)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
mtext (side = 3, 'BVN circles', line = 1, adj = 0)
BVN.b <- runsim(nrepl = 1, traps = grid10, template = templateEllipse, N = 20,
    IWSlowerleft = TRUE, plt = TRUE, allinside = FALSE, gridaligned = FALSE, SECR = FALSE)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
mtext (side = 3, 'BVN ellipses', line = 1, adj = 0)
```



For rectangular home ranges we shade each cell according to its random utilization. The switch ‘IWSlowerleft’ forces the particular placement algorithm used by IWS. Lower left cells are marked ‘x’ and a green dot is added for the home range utilization centroids.

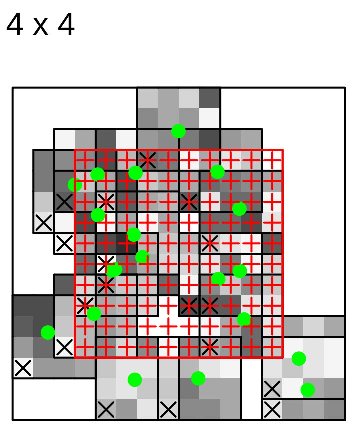
```
par(mfrow = c(1,2), pty = 's', mar = c(3,3,3,3), cex = 1, xpd = TRUE)
HR44.c <- runsim(nrepl = 1, traps = grid10, template = template44, N = 20,
    IWSlowerleft = TRUE, plt = TRUE, allinside = FALSE, gridaligned = TRUE, SECR = FALSE)
```

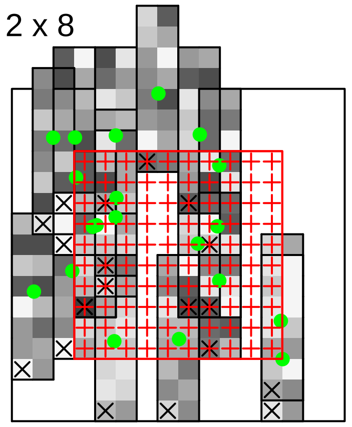
```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
mtext (side = 3, '4 x 4', line = 1, adj = 0)
HR28.d <- runsim(nrepl = 1, traps = grid10, template = template28, N = 20,
    IWSlowerleft = TRUE, plt = TRUE, allinside = FALSE, gridaligned = TRUE, SECR = FALSE)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
mtext (side = 3, '2 x 8', line = 1, adj = 0)
```



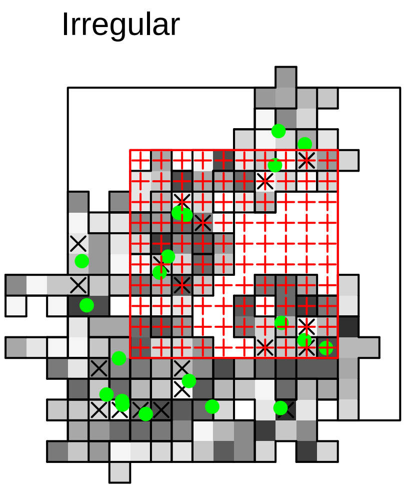


Irregular ranges are more fun: we pass a function (build.irregular) rather than a template. Here 'x' marks the initial cell. Many are truncated at the boundary of the arena.

```
par(mfrow = c(1,2), pty = 's', mar = c(3,3,3,3), cex = 1, xpd = TRUE)
Irregular.e <- runsim(nrepl = 1, traps = grid10, template = build.irregular, N = 20,
    IWSlowerleft = TRUE, plt = TRUE, allinside = FALSE, gridaligned = TRUE, SECR = FALSE)
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type =
## ifelse(type == : prediction from a rank-deficient fit may be misleading
```

```
mtext (side = 3, 'Irregular', line = 1, adj = 0)
```



Running simulations

To perform simulations, we set `SECR = TRUE` and increase the number of replicates. Here is the code. Note we can specify ncores > 1 to use multiple cores (here 7 out of 8 on a quad-core Windows PC), and add arguments

CL and start that are passed to secr.fit. Default values of runsim arguments specify the 'intermediate' scenario of IWS.

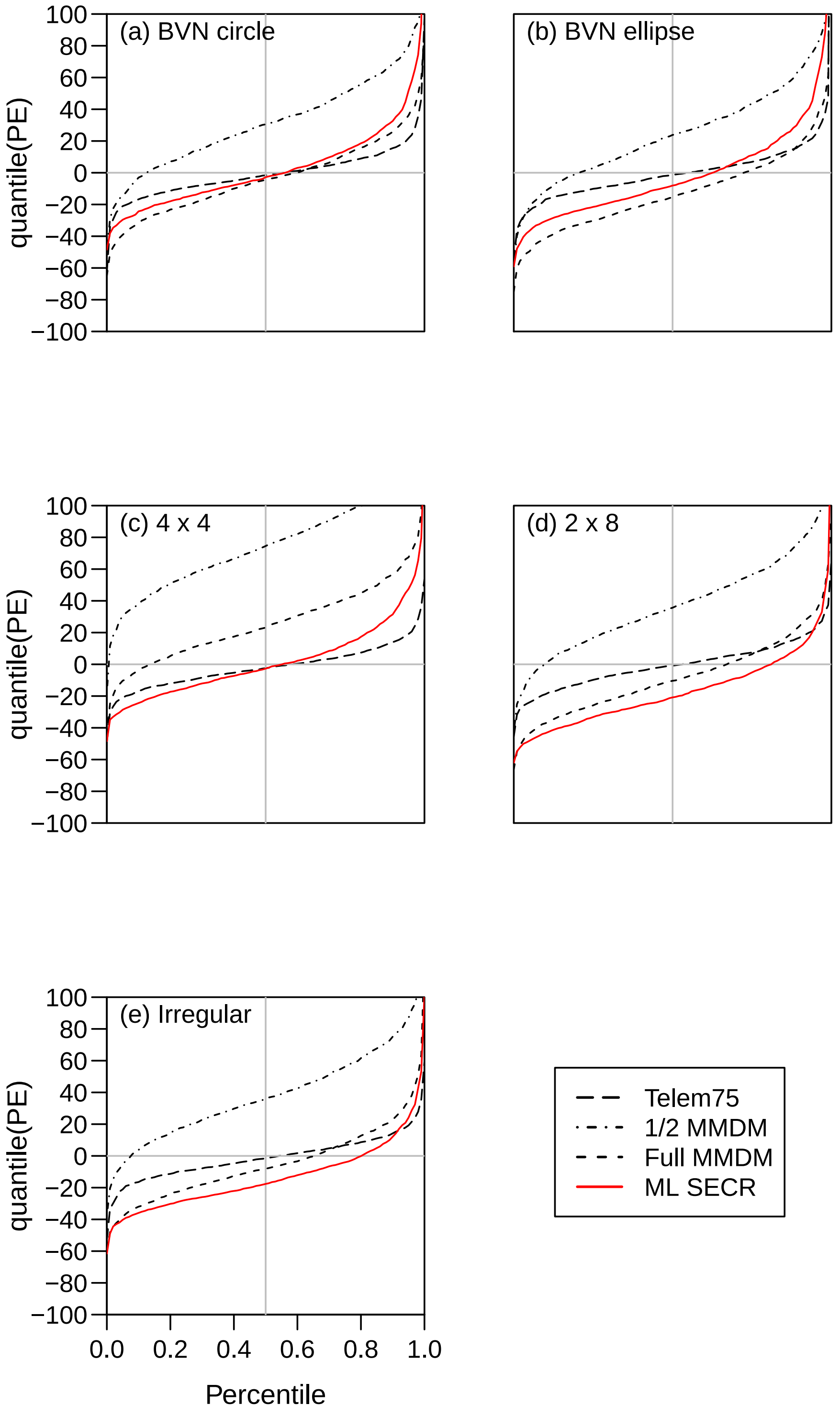
```
out <- vector('list', 5)
names(out) <- letters[1:5]
nrepl <- 1000
out$aIWS <- runsim(nrepl, traps = grid10, template = templateCircle, N = 20, detectfn = 'HHN',
                CL = TRUE, start = list(lambda0 = 0.1, sigma = 10), IWSlowerleft = TRUE,
                allinside = FALSE, gridaligned = FALSE, ncores = 7)
out$bIWS <- runsim(nrepl, traps = grid10, template = templateEllipse, N = 20, detectfn = 'HHN',
                CL = TRUE, start = list(lambda0 = 0.1, sigma = 10), IWSlowerleft = TRUE,
                allinside = FALSE, gridaligned = FALSE, ncores = 7)
out$cIWS <- runsim(nrepl, traps = grid10, template = template44, N = 20, detectfn = 'HHN',
                CL = TRUE, start=list(lambda0 = 0.1, sigma = 10), IWSlowerleft = TRUE,
                allinside = FALSE, gridaligned = TRUE, ncores = 7)
out$dIWS <- runsim(nrepl, traps = grid10, template = template28, N = 20, detectfn = 'HHN',
                CL = TRUE, start = list(lambda0 = 0.1, sigma = 10),  IWSlowerleft = TRUE,
                allinside = FALSE, gridaligned = TRUE, ncores = 7)
out$eIWS <- runsim(nrepl, traps = grid10, template = build.irregular, N = 20, detectfn = 'HHN',
                CL = TRUE, start=list(lambda0 = 0.1, sigma = 10),  IWSlowerleft = TRUE,
                allinside = FALSE, gridaligned = TRUE, ncores = 7)
```

Summarising simulations

The function plotsim computes the percentage error for a specified estimator using one of the benchmark methods in Table 1 and makes a quantile plot as used by IWS. Here we wrap several calls to plotsim in a new function to reproduce IWS Fig. 3.

```
Fig3plots <- function (out, plotnum = 1:5, cols = c('black','black','black','red', 'blue'),
                       true = c('telem', 'total', 'inner'), secrt = FALSE) {
    par(mfrow=c(3,2), xpd=F, cex = 0.9, pty = 's', mar=c(3,3,0,0), mgp=c(2.2,0.65,0), oma = c(3,3,3,3))
    for (i in plotnum) {
        plotsim(out[[i]], estimator = 'TELEM', type = true[1], lty = 5, col=cols[1],
                label = c('y','','y','','xy','x')[i])
        plotsim(out[[i]], estimator = 'MMDM2',  type = true[2], add = TRUE, lty=10, col=cols[2])
        plotsim(out[[i]], estimator = 'MMDM',  type = true[2], add = TRUE, lty=2, col=cols[3])
        plotsim(out[[i]], estimator = 'SECR',  type = true[3], add = TRUE, lty=1, col=cols[4])
        if (secrt)
            plotsim(out[[i]], estimator = 'SECRT', type = true[1], add = TRUE, lty=1, col=cols[5])
        text (0.04, 88, adj = 0, paste('(', letters[i],')',
            c(' BVN circle', ' BVN ellipse', ' 4 x 4', ' 2 x 8', ' Irregular')[i], sep = ''))
        }
    plot(c(0,1), c(0,1), type = 'n', xlab = '', ylab = '', axes = FALSE)
    OK <- 1:(4+secrt)
    legend (0.1, 0.8, legend = c('Telem75', '1/2 MMDM','Full MMDM', 'ML SECR', 'ML SECR + T')[OK],
            lty = c(5, 10, 2, 1, 1)[OK], lwd = 1.5, col= cols[OK])
}
```

```
 ## load previously saved simulations
load(file = paste(vignettefolder, 'out.RData', sep=''))
Fig3plots(out)
```


(a) BVN circle
(b) BVN ellipse
(c) 4 x 4
(d) 2 x 8
(e) Irregular
quantile(PE)
100
80
60
40
20
0
−20
−40
−60
−80
−100
0.0
0.2
0.4
0.6
0.8
1.0
Percentile
Telem75
1/2 MMDM
Full MMDM
ML SECR

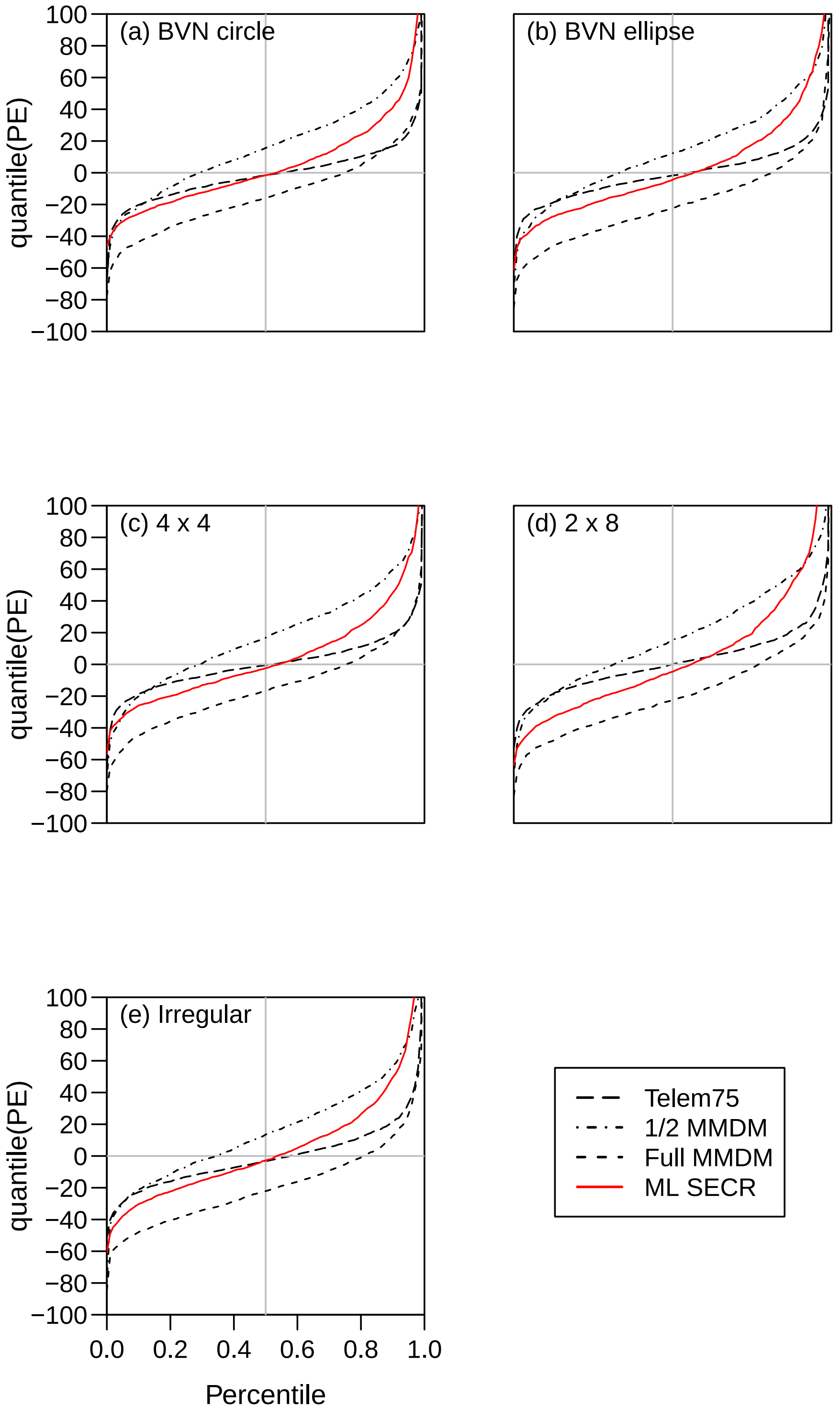
This is close to Fig. 3 of IWS. There is noticeable bias of SECR estimates in (b), (d) and (e).

Revised simulations

We now re-run the simulations using uniform random placement of home ranges, instead of the idiosyncratic placement used by IWS, by resetting the argument `IWSlowerleft`. The code can fail unpredictably with the message “finite coordinates are needed”. The source of this error has not been traced; the solution for now is to re-run the simulations.

```
outr <- vector('list', 5)
names(outr) <- letters[1:5]
nrepl <- 1000
outr$a <- runsim(nrepl, traps = grid10, template = templateCircle, N = 20, detectfn = 'HHN',
                CL = TRUE, start = list(lambda0 = 0.1, sigma = 10), IWSlowerleft = FALSE,
                allinside = FALSE, gridaligned = FALSE, ncores = 7)
outr$b <- runsim(nrepl, traps = grid10, template = templateEllipse, N = 20, detectfn = 'HHN',
                CL = TRUE, start = list(lambda0 = 0.1, sigma = 10), IWSlowerleft = FALSE,
                allinside = FALSE, gridaligned = FALSE, ncores = 7)
outr$c <- runsim(nrepl, traps = grid10, template = template44, N = 20, detectfn = 'HHN',
                CL = TRUE, start=list(lambda0 = 0.1, sigma = 10), IWSlowerleft = FALSE,
                allinside = FALSE, gridaligned = TRUE, ncores = 7)
outr$d <- runsim(nrepl, traps = grid10, template = template28, N = 20, detectfn = 'HHN',
                CL = TRUE, start = list(lambda0 = 0.1, sigma = 10), IWSlowerleft = FALSE,
                allinside = FALSE, gridaligned = TRUE, ncores = 7)
outr$e <- runsim(nrepl, traps = grid10, template = build.irregular, N = 20, detectfn = 'HHN',
                CL = TRUE, start=list(lambda0 = 0.1, sigma = 10), IWSlowerleft = FALSE,
                allinside = FALSE, gridaligned = TRUE, ncores = 7)
```

```
load(file = paste(vignettefolder, 'outr.RData', sep='')) ## load previously saved simulations
Fig3plots(outr)
```


(a) BVN circle
(b) BVN ellipse
(c) 4 x 4
(d) 2 x 8
(e) Irregular
quantile(PE)
100
80
60
40
20
0
−20
−40
−60
−80
−100
0.0
0.2
0.4
0.6
0.8
1.0
Percentile
Telem75
1/2 MMDM
Full MMDM
ML SECR

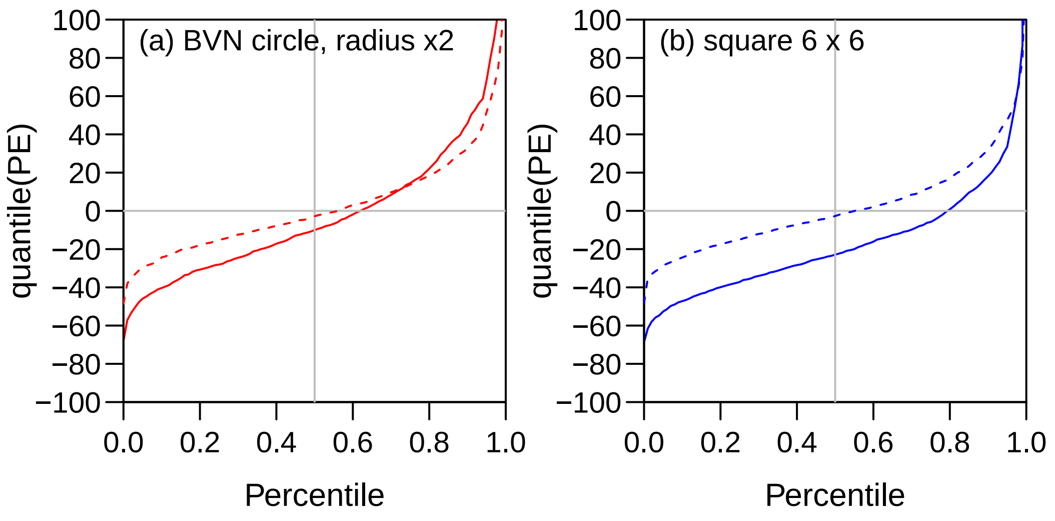
Large symmetrical ranges

A critical difference between the elongated ranges (ellipse, rectangle, irregular) and the symmetrical ranges (square CELLS, circular BVN) is that the latter would not overlap the 'traps' if centred in the empty 'edge' zone of the arena. We repeat the simulations with enlarged square and circular ranges.

```
outenl <- vector('list', 2)
names(outenl) <- c('square', 'circle')
nrepl <- 1000
templateCircle2 <- templatefn(nx = 1, ny = 1,
                 sigmaX = 0.921*10*2, sigmaY = 0.921*10*2, spacing = 10)
template66 <- templatefn(nx = 6, ny = 6, spacing = 10)
outenl$circle2 <- runsim(nrepl, traps = grid10, template = templateCircle2, N = 20, detectfn = 'HHN',
                   CL = TRUE, start = list(lambda0 = 0.1, sigma = 10), IWSlowerleft = TRUE,
                   allinside = FALSE, gridaligned = FALSE, ncores = 7)
outenl$square <- runsim(nrepl, traps = grid10, template = template66, N = 20, detectfn = 'HHN',
                  CL = TRUE, start = list(lambda0 = 0.1, sigma = 10), IWSlowerleft = TRUE,
                  allinside = FALSE, gridaligned = FALSE, ncores = 7)
```

```
## Warning in runsim(nrepl, traps = grid10, template = template66, N = 20, :
## TELEM estimator only implemented for grid aligned or BVN HR
```

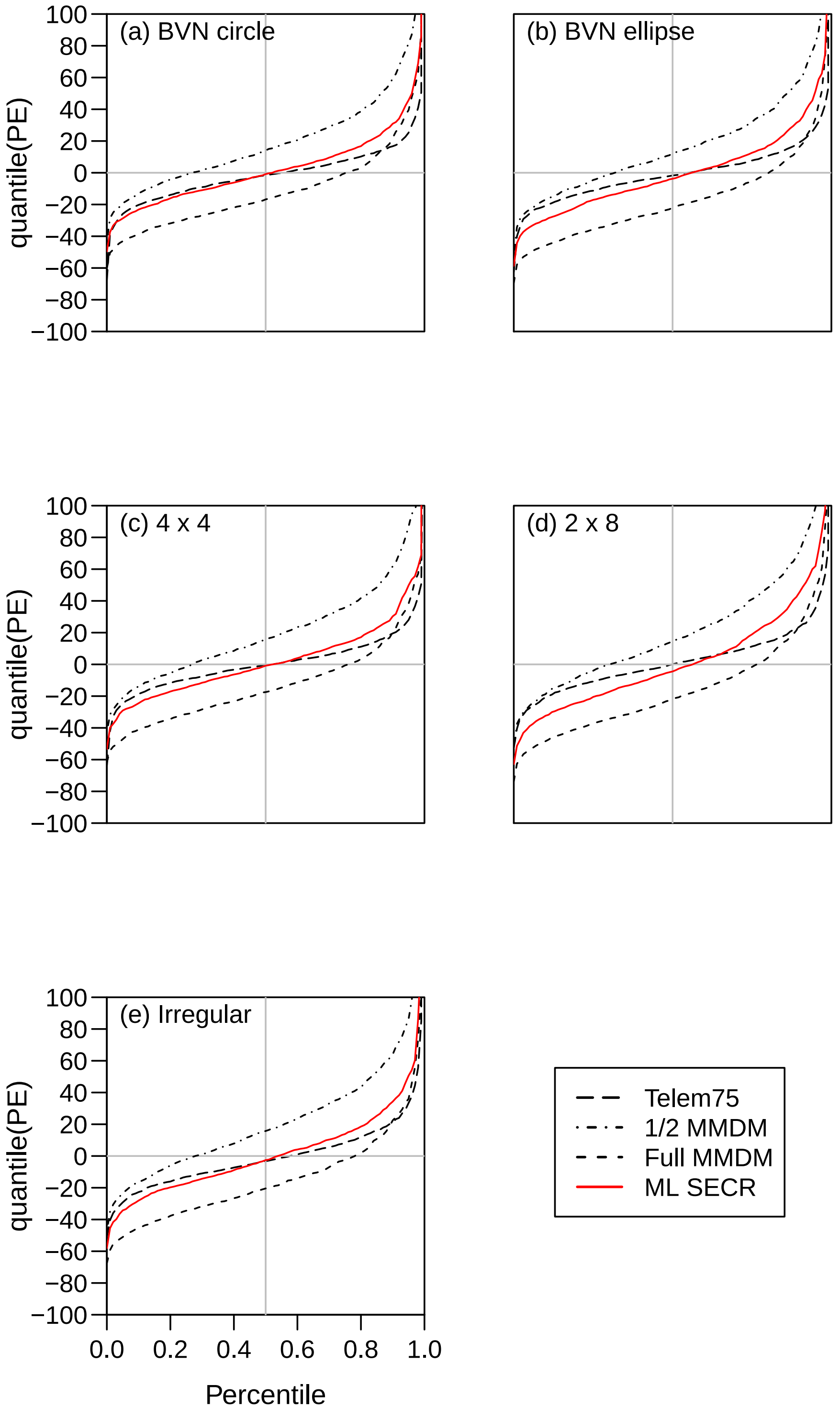
```
par(mfrow=c(1,2), xpd=F, cex = 0.9, pty = 's', mar=c(3,4,0,0), mgp=c(2.2,0.65,0), oma = c(3,3,3,3))
plotsim(outenl[['circle2']], estimator = 'SECR', type = 'inner', lty = 1, col = 'red')
plotsim(out[['aIWS']], estimator = 'SECR', type = 'inner', lty = 2, col = 'red', add = TRUE)
text (0.04, 88, adj = 0, '(a) BVN circle, radius x2')

plotsim(outenl[['square']], estimator = 'SECR', type = 'inner', lty = 1, col = 'blue')
plotsim(out[['cIWS']], estimator = 'SECR', type = 'inner', lty = 2, col = 'blue', add = TRUE)
text (0.04, 88, adj = 0, '(b) square 6 x 6')
```



Placing all comparisons on an equal footing

Each of the three classes of estimator was evaluated by IWS against different 'true densities'. The quest for 'fairness' ultimately penalized both MMDM and SECR. This can be seen by using a common basis of comparison ('telem' from Table 1).

```r
Fig3plots(outr, true = rep('telem',3))
```


(a) BVN circle
(b) BVN ellipse
(c) 4 x 4
(d) 2 x 8
(e) Irregular
quantile(PE)
100
80
60
40
20
0
−20
−40
−60
−80
−100
0.0
0.2
0.4
0.6
0.8
1.0
Percentile
Telem75
1/2 MMDM
Full MMDM
ML SECR

Interpretation

The erratic behaviour of MMDM is to be expected from its ad hoc nature, and needs no further comment.

SECR shows a small median bias when ranges are elongated (BVN ellipse, 2×8)

Details

The realism of the home ranges simulated in IWS is open to question, and does not exhaust the challenges faced by estimators. The effect of range elongation (noncircularity) will interact with the trap layout if the layout itself has a dominant direction (the IWS layout was a square grid with no dominant direction).

The definition of traps in IWS as grid cells rather than points leaves some ambiguity for concepts such as 'buffered area' for MMDM and 'distance to edge' (DTE) for TELEM. As far as I can tell IWS used the same solutions as 'secrIWS': area for the MMDM estimators buffered around the centroids of outer grid cells (rather than the gridcells themselves, and for calculation of MMDM captures were considered to occur at the centroids, as in the MMDM function of 'secr'.

The IWS calculation of DTE assumes the trapping grid ranges from 0 to xmax = 9, i.e. grid 9 units x 9 units. e.g. line 114 in 'Jake Bivariate Normal Circle4.SAS') so the edge is taken to be the convex hull of the centroids, not the grid-cell edge.

References

Huggins, R. M. (1989) On the statistical analysis of capture experiments. *Biometrika* **76**, 133–140.

Ivan, J. S., G. C. White, and T. M. Shenk. (2013a) Using auxiliary telemetry information to estimate animal density from capture–recapture data. *Ecology* **94**, 809–816.

Ivan, J. S., White, G. C. and Shenk, T. M. (2013b) Using simulation to compare methods for estimating density from capture–recapture data. *Ecology* **94**, 817–826.