openCR 2.2 - properties of movement kernels

Murray Efford

2022-01-17

Contents

Introduction

The R package **openCR** fits movement models in spatial capture–recapture using a discretized kernel. 'Movement' refers to a change in activity centre between primary sessions. The main motivation is to improve estimates of vital rates (survival and recruitment).

The shape, size and resolution of the discretized kernel can affect estimates of the movement parameters, particularly the scale of movement (α or parameter 'move.a') and turnover (survival ϕ and per capita recruitment f). The 'size' of a discretized kernel here means its radius in terms of the integer number of (mask) cells in north-south and east-west directions.

The spatial model was described by Efford and Schofield (2020; see openCR-vignette.pdf for further background). These notes investigate the various software settings.

Quick start

A movement submodel may be specified for any of the spatial model types[1] for `openCR.fit` (i.e. those including 'secr' in their name - 'PLBsecrf', 'JSSAsecrD' etc.). The default movement model is 'static' (no movement).

These settings are important –

Function	Argument	Notes
`make.mask`	buffer	compare multiple buffer widths to be sure this is large enough
	spacing	choose carefully to minimise size of mask (up to 60% of HN sigma); the digitized kernel has the same spacing
`openCR.fit`	movementmodel	'BVT' (bivariate t) and 'BVNzi' (zero-inflated BVN) are good 2-parameter options; compare single-parameter 'BVC' (bivariate Cauchy) and 'RDE' (radial exponential); see below for descriptions
	kernelradius	make sure this is large enough (measured in units of mask spacing)
	sparsekernel	set to TRUE unless you have unlimited time; see Sparse kernel
	edgemethod	only the default 'truncate' makes much sense; see Edge effects

Here is an example using the small ovenbird dataset from **secr** that loads automatically with **openCR** –

```
library(openCR)
setNumThreads(7)
msk <- make.mask(traps(ovenCHp[[1]]), buffer = 300, spacing = 20)
fit0 <- openCR.fit(ovenCHp, type = "PLBsecrf", mask = msk)  # static default
fitc <- openCR.fit(ovenCHp, type = "PLBsecrf", mask = msk,  # bivariate Cauchy
    movementmodel = "BVC", kernelradius = 30, sparsekernel = TRUE)
```

To work with a set of fitted models it helps to combine them as an 'openCRlist'. Then compare the fitted models with respect to the maximized log likelihood, model rank (roughly, the number of estimable parameters), Akaike's information criterion (AIC), etc. –

```
fits <- openCRlist(fit0, fitc)
AIC(fits)[,-1]    # drop one column to save space
```

```
##      npar rank    logLik      AIC     AICc   dAIC AICwt
## fitc    5    5 -1430.293 2870.586 2871.523  0.000     1
## fit0    4    4 -1448.709 2905.417 2906.032 34.831     0
```

[1] That includes 'CJSsecr' if you want only survival and movement estimates, but remember to set the 'details' argument `CJSp1 = TRUE`. The default `CJSp1 = FALSE` gives bad estimates because it does not model the bias of initial locations towards the detectors (Efford and Schofield 2020).

Estimates may be obtained from the fitted models as usual with `predict(fits)`. Also:

```
make.table(fits, parm = "phi")
```

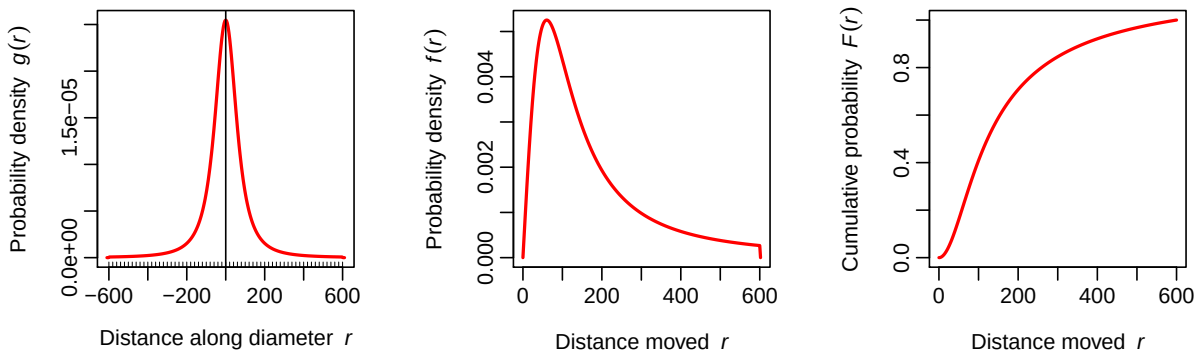
```
##       session
## model       2005      2006      2007      2008 2009
##   fit0 0.5282841 0.5282841 0.5282841 0.5282841
##   fitc 0.6053745 0.6053745 0.6053745 0.6053745
```

NOTE: Further testing shows that the 300-m buffer is inadequate, and these estimates of annual survival should not be trusted.

We can examine the fitted movement model by constructing a 'kernel' object and plotting it in different ways. Use the `plot` method for kernel objects (see `?plot.kernel` for help, and below for more on the distribution) functions.

```
k <- make.kernel(fitc)  # kernel based on fitted Cauchy model
```

```
par(mfrow = c(1,3), pty = "s", mar = c(6,3,2,2), cex = 0.9, mgp = c(2.4, 0.6, 0))
plot(k, type = c("gr", "fr", "Fr"), col = "red", lwd = 2)
```



Functions to manipulate kernels are listed in Appendix 1.

Movement kernels in general

Movements are assumed to follow a random walk with step length and direction governed by a 2-dimensional probability density function (pdf). A pdf with circular contours can be treated as a function $g(r)$ of distance from the initial location, r. We distinguish the univariate distribution of distance moved as $f(r)$, where $f(r) = 2\pi r\, g(r)$.

The bivariate normal (BVN), Laplace (BVE), Cauchy (BVC), and t (BVT) distributions are shapes for the circular 2-D kernel pdf defined directly as $g(r)$ (Clark et al., 1999; Cousens et al., 2008). The kernel may also be specified in polar coordinates using the distribution of radial distance moved $f(r)$ and a uniform orientation θ; this leads to the exponential (RDE), gamma (RDG) and log-normal (RDL) forms of Ergon and Gardner (2014). Both sets of options are defined in Table 1 according to the resulting distribution of distance moved. User-defined kernel functions may also be used, but they are incompatible with multithreaded execution and therefore take a long time to fit.

Possible movement models are listed on a help page. See `?'Movement models'` or `help('Movement models', 'openCR')`.

Table 1. Kernel shapes in **openCR**. $F(r)$ is the cumulative distribution of distances moved. The 2-D pdf is $g(r) = f(r)/2\pi r$.

Kernel	Alias	$f(r)$	Mean r	$F(r)$	$F^{-1}(p)$
Bivariate normal	BVN	$\frac{r}{\alpha^2} e^{\frac{-r^2}{2\alpha^2}}$	$\frac{\alpha\sqrt{\pi}}{\sqrt{2}}$	$1 - e^{\frac{-r^2}{2\alpha^2}}$	$\sqrt{-2\ln(p)\alpha^2}$
Bivariate Laplace	BVE	$\frac{r}{\alpha^2} e^{\frac{-r}{\alpha}}$	2α	$1 - (\frac{r}{\alpha} + 1) e^{\frac{-r}{\alpha}}$	no simple form
Bivariate Cauchy	BVC	$\frac{r\alpha}{(\alpha^2 + r^2)^{\frac{3}{2}}}$	undefined	$1 - \frac{\alpha}{\sqrt{\alpha^2 + r^2}}$	$\frac{\alpha\sqrt{p(2-p)}}{1-p}$
Bivariate t	BVT	$\frac{2\beta r}{\alpha^2} \left(1 + \frac{r^2}{\alpha^2}\right)^{-(\beta+1)}$	$\alpha \frac{\sqrt{\pi}}{2} \frac{\Gamma(\beta - 0.5)}{\Gamma(\beta)}$, $\beta > 0.5$	$1 - \left(\frac{\alpha^2}{\alpha^2 + r^2}\right)^{\beta}$	$\alpha\sqrt{p^{\frac{-1}{\beta}} - 1}$
Exponential distance	RDE	$\frac{1}{\alpha} e^{\frac{-r}{\alpha}}$	α	$1 - e^{\frac{-r}{\alpha}}$	$-\alpha \ln(1-p)$
Gamma distance	RDG	$\frac{1}{\Gamma(\beta)\alpha^{\beta}} r^{\beta-1} e^{\frac{-r}{\alpha}}$	$\alpha\beta$	$\frac{1}{\Gamma(\beta)} \gamma(\beta, \frac{r}{\alpha})$	no simple form
Lognormal distance	RDL	$\frac{1}{r\sigma\sqrt{2\pi}} e^{\frac{-(\ln(r)-\mu)^2}{2\sigma^2}}$	$e^{\mu + \frac{\sigma^2}{2}}$	$\Phi\left(\frac{\ln(r)-\mu}{\sigma}\right)$	$e^{\mu + \sqrt{2\sigma^2}\mathrm{erf}^{-1}(2p-1)}$

The symbols α and β correspond in **secr** and **openCR** to parameters move.a and move.b, respectively. In each case α is a scale parameter.

Bivariate normal distribution BVN

An uncorrelated bivariate normal distribution BVN with equal variance in x- and y- directions is a convenient and widely used option, but its tails are short (light). One consequence of light tails is that occasional outlying movements will have a disproportionate effect on the fitted scale parameter. The pdf of distance moved is a Weibull distribution with shape parameter 2 and scale $\sqrt{2}\sigma$, otherwise known as a Rayleigh distribution (e.g., Hooten et al. 2017 Ch. 5).

Bivariate Laplace distribution BVE

A univariate Laplace distribution declines exponentially either side of the mean. The bivariate Laplace kernel (BVE) looks like a simple analogue of the Gaussian with heavier tail. However, there are complications: the negative exponential form used in ecology (e.g., Clark et al. (1999 Eq 5a), Cousens et al. (2008 Table 5.2), Nathan et al. (2012 Table 15.1)) is not the bivariate symmetric Laplace distribution considered by Kotz et al. (2001 p. 231). That distribution has infinite pdf at $r = 0$.

The pdf of distance moved for the bivariate Laplace distribution used in ecology is $f(r) = \frac{r}{\alpha^2} \exp - \frac{r}{\alpha}$ which corresponds to a gamma distribution with shape parameter $k = 2$ and scale parameter α. This provides a handy way to simulate movements: draw r from a random gamma distribution with $k = 2$ and scale α using, e.g., the R function stats::rgamma), and travel in a random direction θ uniform on $(0, 2\pi)$.

The properties of the gamma distribution are inconvenient: there is no simple closed form for the inverse of the distribution function. This is computed numerically in `qkernel()`. Other properties of $f(r)$ for BVE are simpler: the mode is at α and the mean at 2α.

Bivariate Cauchy distribution BVC

The bivariate Cauchy is a heavy-tailed single-parameter distribution identical to the following BVT with parameter $\beta = 0.5$ (i.e. df = 1).

Bivariate t-distribution BVT

The bivariate t-distribution BVT is less rigid than the single-parameter distributions, but there are complications. The distribution may have a heavier or lighter tail than the bivariate Laplace distribution depending on the shape parameter β (half the degrees of freedom). Ecological usages vary in their parameterisation, but nevertheless follow the bivariate generalisation of a univariate t distribution in Kotz and Nadarajah (2004 Eq. 1.1). The parameterisation used in **openCR** is related to others in Table 2 (see also Appendix 2).

Table 2. Parameterisations of standard uncorrelated bivariate t-distribution.

Source	Scale	Shape
Kotz and Nadarajah (2004)	$\Sigma = \begin{bmatrix} \sigma^2 & 0 \\ 0 & \sigma^2 \end{bmatrix}$	ν (degrees of freedom)
Clark et al. (1999)	$u = \nu\sigma^2$	$p = \frac{\nu}{2}$
Nathan et al. (2012)	$a = \sqrt{\nu}\sigma$	$b = \frac{\nu}{2} + 1$
openCR	$\alpha = a = \sqrt{u} = \sqrt{\nu}\sigma$	$\beta = p = b - 1 = \frac{\nu}{2}$

Cousens et al. (2008) use the form of Clark et al. (1999), but re-label u as b and p as a.

Bivariate distributions defined in terms of polar coordinates RDE, RDG, RDL

The 'exponential', 'gamma' and 'log-normal' kernels of Ergon and Gardner (2014) were defined in terms of the radial distance moved $f(r)$ and direction θ (for simplicity, θ is distributed uniformly on $(0, 2\pi)$). The polar coordinates (r, θ) then define a bivariate distribution $g(r)$.

The movement models RDE, RDG and RDL in **openCR** correspond to the models of Ergon and Gardner (2014). The parameterization of RDL is unconventional: $\alpha = \exp(\mu)$ is the median, and $\beta = 1/\mathrm{CV}^2(r) = 1/(e^{\sigma^2} - 1)$ is a convenient shape parameter that is independent of scale (cf Table 1).

Kernel-free movement IND

The 'independent' model of Gardner et al. (2018) is a further movement model, signified 'IND' in **openCR**. The independent model does not specify a kernel: movers are allowed to relocate randomly and uniformly within the habitat mask (state space). The extent of movement depends critically on the buffer width where this is arbitrary. There seems to be little justification for using the model; it is included for completeness, and because its zero-inflated form (below) is quite interesting.

Zero-inflated kernels 'zi'

Movement kernels may in principle be mixtures of different 2-D probability density functions, with the mixing fraction(s) controlled by one or more additional parameters. Zero-inflated distributions are a particular type

of 2-part mixture that is intuitively useful when some animals are stationary and others move (Ergon and Gardner 2014).

openCR 2.2 implements zero-inflated forms of each of the 1-parameter discretized movement kernels (BVN, BVE, BVC, RDE), with the zero-inflation as a second parameter. We define zero inflation in terms of the cell-specific probabilities of the discretized kernel, denoted $p(r)$ for a cell centred at distance r from the origin:

$$\begin{aligned} p_z(r) &= \pi_z + (1-\pi_z)p(r), \quad r = 0, \\ &(1-\pi_z)p(r) \qquad\quad r > 0. \end{aligned}$$

Zero inflation is also implemented for two further movement options:

1. A uniform kernel 'UNI' truncated at a predetermined radius, and
2. The 'independent' movement model of Gardner et al. (2018).

Each of these options is ostensibly 'parameter-free' in that no movement parameter is fitted, but the scale of movement is determined by a user setting – the kernel radius for (1) and buffer width for (2). In the zero-inflated versions a single parameter α is fitted.

Extreme values of the zero inflation parameter drive the independent 'IND' model to either the 'static' (no movement) scenario ($\alpha = 1$) or completely uncorrelated locations ($\alpha = 0$). This allows a useful check on the coding. For $\alpha > 0$ locations are no longer uncorrelated between sessions.

The implementation in **openCR** 2.2 uses the suffix 'zi' for zero inflation (hence BVNzi, BVEzi, INDzi etc.).

Distribution of distance moved

Kernel shape is represented by the argument `movementmodel` of several **openCR** functions. Possible values include 'BVN', 'BVE', 'BVC', and 'BVT'. For each 2-D kernel centred on the initial location there is a corresponding 1-D distribution of step length $f(r) = 2\pi r\, g(r)$. The distribution of step length ($f(r)$) is sometimes referred to as the movement kernel (e.g., Schaub and Royle 2014); that term is reserved here for the 2-D pdf. However, $f(r)$ is a handy summary of movement, with intuitive summary statistics such as expected (mean) and median distance moved. It is also useful in simulation: in **secr** $\geq$ 4.4.7 all BVN, BVE and BVT simulations with `sim.popn()` use polar-coordinate form, sampling direction θ from a uniform distribution on $(0, 2\pi)$) and distance r from the appropriate $f(r)$.

Distribution functions in openCR

The functions in Table 3 are useful for summarising and plotting theoretical values of the distance distribution for kernels given numerical values for the parameters.

Table 3. Distribution functions in **openCR**. The arguments `move.a` and `move.b` correspond to α and β in Table 1.

Function	Key arguments	Value	Note
dkernel	r, movementmodel, move.a, move.b	$f(r) = 2\pi r\, g(r)$	pdf of distance moved
pkernel	q, movementmodel, move.a, move.b	$F(q) = \int_0^q f(r)\, dr$	cdf of distance moved
qkernel	p, movementmodel, move.a, move.b	$F^{-1}(p)$	quantile function
gkernel	r, movementmodel, move.a, move.b	$g(r)$	pdf of 2-D kernel

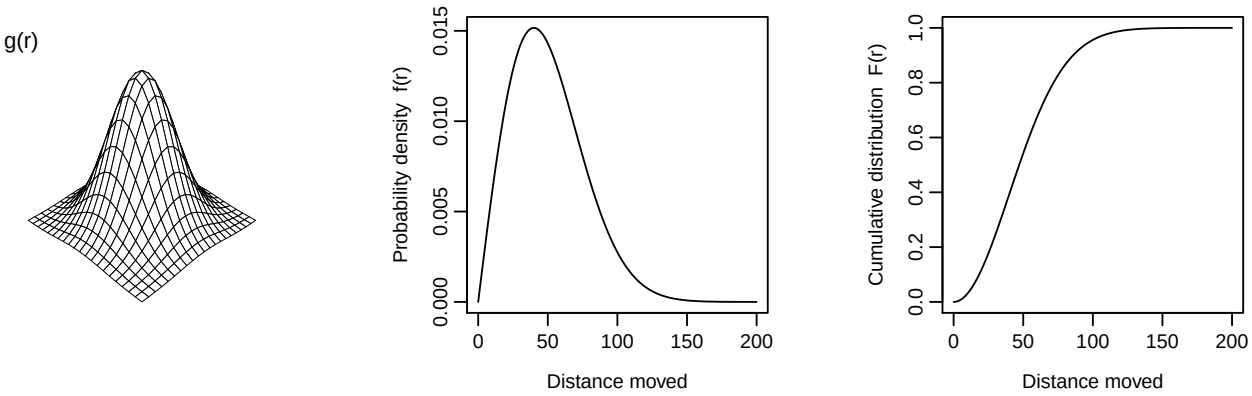


Figure 1. Representations of movement kernel. 2-D pdf $g(r)$ using `gkernel` (left) and corresponding plots of distance moved using `dkernel` (centre) and `pkernel` (right). Bivariate normal (BVN) with $\alpha = 40$.

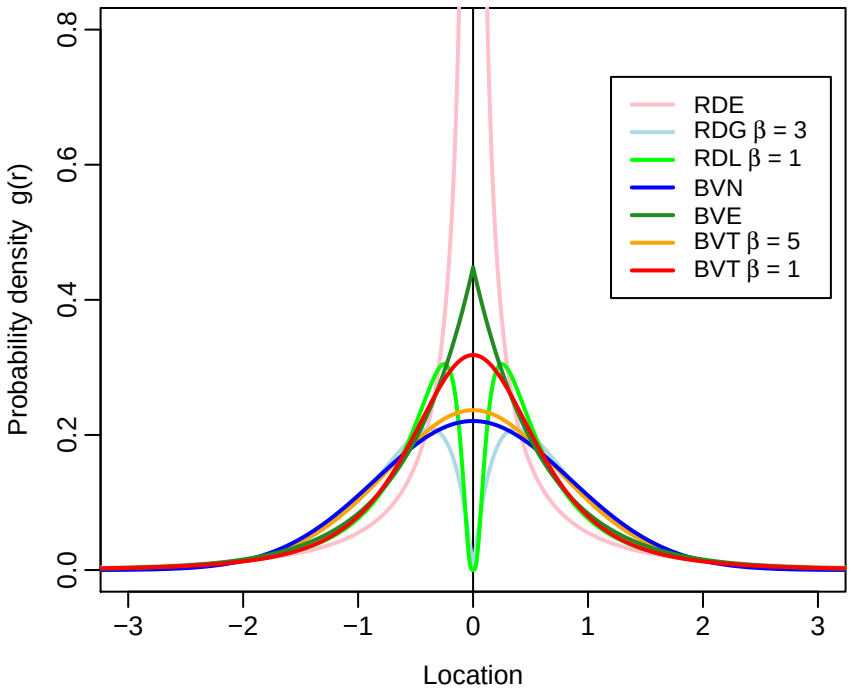


Figure 2. Cross-sections of bivariate movement kernels, all with median expected distance moved 1.0.

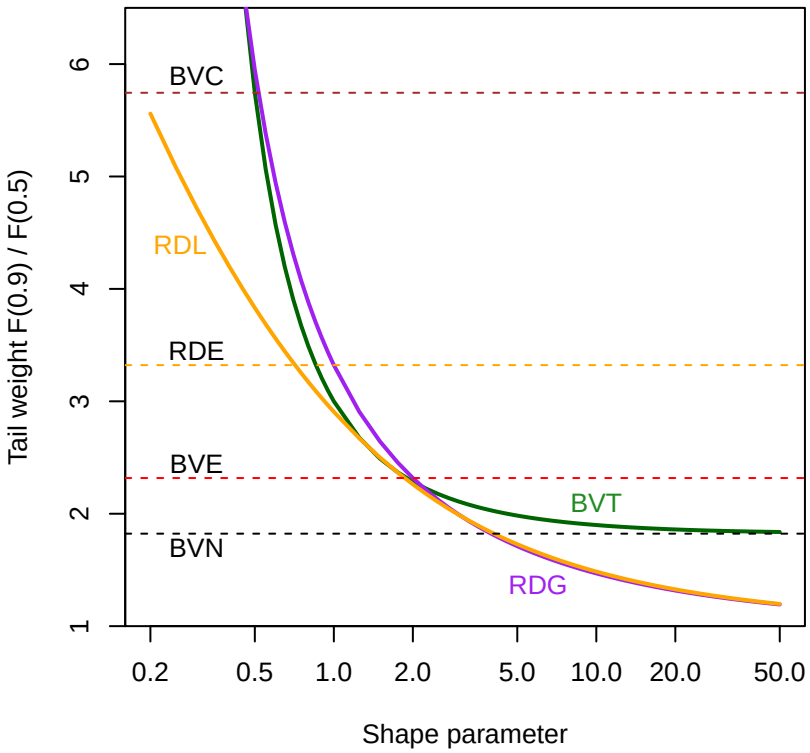


Figure 3. Relative tail weight of movement kernels. An index of tail weight (T_{90} = ratio of 90th and 50th percentiles of distance moved) is plotted against the logarithm of the shape parameters β of the 2-parameter kernels BVT, RDL and RDG, and when constant (1-parameter kernels BVN, BVE, BVC, RDE) as horizontal dashed lines

Tail weight

Tail weight is of particular interest for its relation to emigration. We measure tail weight by the ratio of the 0.9 quantile to the 0.5 quantile (the median distance moved) for which we use the symbol T_{90}. This becomes large in heavy-tailed distributions.

Fig. 3 makes the point that the single-parameter kernels BVN, BVE, BVC, and RDE have a constant shape, whereas BVT is more heavy-tailed than BVE for small β (few degrees of freedom). BVT has a heavier tail than BVN for all df.

For RDE $T_{90} \approx 3.32$,for BVN $T_{90} \approx 1.82$ and for BVE $T_{90} \approx 2.32$. The value of β for which BVT T_{90} = BVE T_{90} is approximately 1.89. The BVT distribution approaches BVN for large β (i.e. when df tends to infinity) and is identical to Cauchy (BVC) when $\beta = 0.5$.

Kernel discretization

The implementation of spatial capture–recapture in **secr** and **openCR** restricts animal activity centres to points on a fine grid known as the habitat mask. Reasons for discretization are listed in secr-habitatmasks.pdf. The probability of movement between primary sessions is represented in **openCR** by a discretized kernel with the same spacing as the habitat mask. The discretized kernel is truncated at a distance chosen to have minimal effect on the estimates.

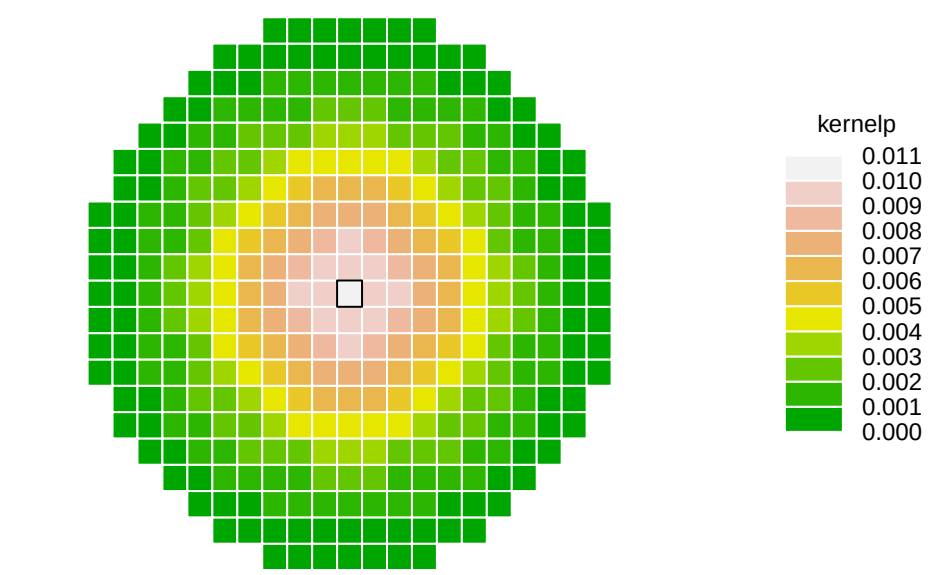
Kernels exist in the software at two levels. A 'kernel' object may be created in R with the **openCR** function `make.kernel`; kernel objects are used for plotting and examining the properties of the discretized kernel, but they are not used internally in model fitting. Kernel skeletons are also created and used internally by `openCR.fit`. The structure of the internal kernel (e.g., radius) is fixed early in a call to `openCR.fit`, but

the probability associated with each kernel cell is recalculated at each likelihood computation as parameter values change. The internal kernel has the same cellsize as the habitat mask, and the internel kernel is always clipped to the nominal radius. The coordinates of the internal kernel are returned as part of the openCR object from `openCR.fit` (**openCR** >= 1.6.0).

Full kernel

Here is a discretized kernel corresponding to Fig. 1, clipped to a radius of 10 cells.

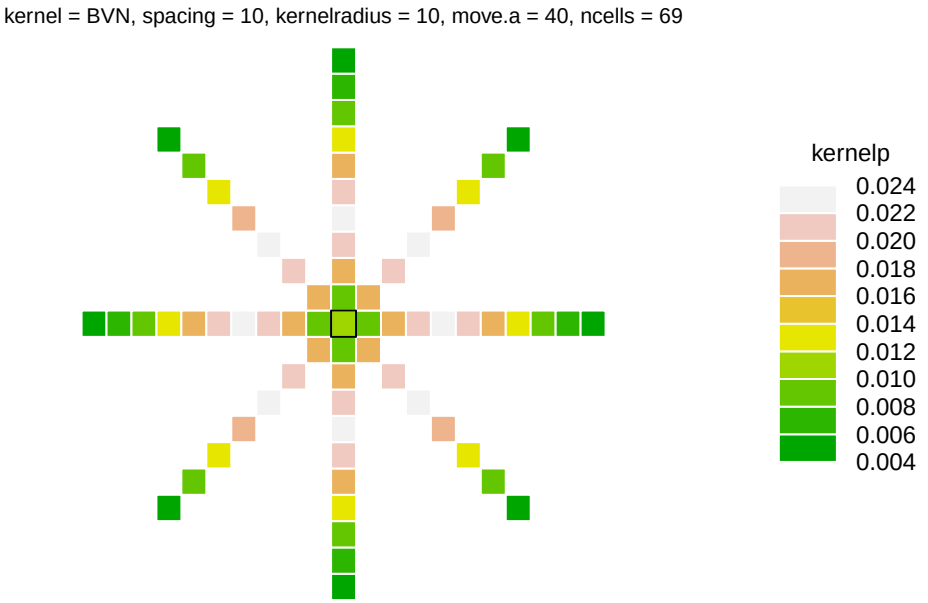
```
par(mar = c(3,1,4,5), cex = 0.8, cex.main = 0.7)
kfull <- make.kernel("BVN", kernelradius = 10, spacing = 10, move.a = 40, clip = TRUE)
plot(kfull)
```



Sparse kernel

A big problem with kernels as defined in **openCR** <2.0 is that the number of cells increases with the square of the radius. Processing time increases with the number of cells, and kernel size can easily become a problem. **openCR** 2.0 introduced novel 'sparse' kernels that include only those grid cells that lie on 4 axes (N-S, E-W, NW-SE, NE-SW). The number of cells increases only linearly with radius. Cell-wise movement probabilities are adjusted so that the distribution of dispersal distances is almost unchanged: cell probabilities `kernelp` are multiplied by $2\pi r/8$ at radius r, and cells on the oblique axes are up-weighted by $\sqrt{2}$.

```
par(mar = c(3,1,4,5), cex = 0.8, cex.main = 0.7)
ksparse <- make.kernel("BVN", kernelradius = 10, spacing = 10, move.a = 40, clip = TRUE,
    sparsekernel = TRUE)
plot(ksparse)
```



Note that the maximum on each axis is no longer at the centre. Each oblique arm has only 7 cells; these cells have higher weighting so that the total per arm remains the same, avoiding orientation bias.

Somewhat surprisingly, sparse kernels appear to work just as well as full kernels, only faster (Efford, submitted).

The number of cells in a sparse kernel increases with the kernel radius rather than radius squared (Table 4).

Table 4. Number of cells in full and sparse kernels of various sizes (clipped to radius).

Radius	Full	Sparse
5	97	33
10	349	69
20	1313	137
30	2933	205
40	5169	273
50	8021	341

Kernel radius

openCR truncates the fitted movement kernel at a user-controlled radius (argument `kernelradius` of `openCR.fit()`). Truncation effectively sets the probability of longer movements to zero, and must be set carefully.

Table 5. Proportion of potential movements discarded by truncating the kernel at varying radii (multiples of median distance moved) computed with `pkernel` for BVN, BVE, BVC and BVT kernels ($\beta = 5$ and $\beta = 1$).

Radius	BVN	BVE	BVC	BVT5	BVT1
1	0.500	0.500	0.500	0.500	0.500
2	0.063	0.152	0.277	0.097	0.200
3	0.002	0.039	0.189	0.014	0.100
4	0.000	0.009	0.143	0.002	0.059
5	0.000	0.002	0.115	0.000	0.038
6	0.000	0.000	0.096	0.000	0.027
7	0.000	0.000	0.082	0.000	0.020

Radius	BVN	BVE	BVC	BVT5	BVT1
8	0.000	0.000	0.072	0.000	0.015

The argument 'r0'

`openCR.fit` has a 'details' argument[2] `r0`, that appears also in `make.kernel`. This has an obscure purpose in controlling the probability assigned to the zero (origin) cell when a continuous kernel model is discretized. Some kernel functions are undefined at zero and most are not flat, so special action is required for the zero cell. The preferred method in `openCR` is to estimate the probability in the zero cell as the integral over a disc centred at the origin, using the well-behaved cumulative probability function F. The disc is used to approximate the integral over the square cell. The default in 2.2.0 is to assign the disc a radius of $1/\sqrt{\pi} \times$ cell width. An alternative is r0 = 0.5 (disc within cell). Setting r0 = 0 has a special meaning: use the value of the kernel function at zero distance; before 2.2.0 this was the default for functions that are defined at $d = 0$ (e.g. BVN), and it is useful to be able to return to that default (note r0 = 0 fails with kernels that are undefined or infinite at d = 0).

The consequences of changing r0 are often minor. Table 6 shows some results for kernels with median distance moved equal to twice the cell width.

Table 6. Probability assigned to origin cell in movement kernels discretized with different values of r0. All kernels were scaled to have median distance moved equal to twice cell width (BVT move.b = 1).

r0	BVN	BVE	BVC	BVT1
0	0.055	0.112	0.119	0.080
0.5	0.042	0.067	0.082	0.059
$1/\sqrt{\pi}$	0.054	0.082	0.102	0.074

Summary of discretized kernel

Use the `plot` and `summarize` methods to display the properties of discretized kernels. The `plot` method was used earlier. `spotHeight(ke, dec = 3)` may be used to pick out values from particular cells. Here we generate and summarize two kernels (full and sparse):

```
kfull20 <- make.kernel("BVE", kernelradius = 20, spacing = 10, move.a = 30, clip = TRUE,
    sparsekernel = FALSE)
ksparse20 <- make.kernel("BVE", kernelradius = 20, spacing = 10, move.a = 30, clip = TRUE,
    sparsekernel = TRUE)
summary(kfull20)
```

```
## Kernel radius (cells)     :  20
## Spacing (side of cell)    :  10  (m)
## Number of cells           :  1313
## Movement model            :  BVE
## Parameter(s)              :  move.a = 30
## Proportion truncated      :  0.008438546
## Movement as truncated at edge of kernel
## Empirical mean distance   :  58.46007  (m)
## Expected distance         :  58.28644  (m)
## 50th percentile (median)  :  49.88476  (m)
## 90th percentile           :  113.5084  (m)
## Movement, untruncated kernel
```

[2] You set r0 with something like `openCR.fit(..., details = list(r0 = 0.5))`

```
## Expected distance         :  60  (m)
## 50th percentile (median)  :  50.35041  (m)
## 90th percentile           :  116.6916  (m)
```

```
summary(ksparse20)
```

```
## Kernel radius (cells)     :  20
## Spacing (side of cell)    :  10  (m)
## Number of cells           :  137
## Movement model            :  BVE
## Parameter(s)              :  move.a = 30
## Proportion truncated      :  0.008438546
## Movement as truncated at edge of kernel
## Empirical mean distance   :  57.13869  (m)
## Expected distance         :  58.28644  (m)
## 50th percentile (median)  :  49.88476  (m)
## 90th percentile           :  113.5084  (m)
## Movement, untruncated kernel
## Expected distance         :  60  (m)
## 50th percentile (median)  :  50.35041  (m)
## 90th percentile           :  116.6916  (m)
```

The attribute 'distribution' of a discretized kernel from `openCR::make.kernel()` is the cumulative sum of cell-specific probabilities ordered by distance from the centre. This is a convenient way to compare kernels, as in the following code.

```
x <- 0:200
par(mfrow = c(1,2), mar = c(5,5,3,3), mgp = c(2.2, 0.7,0), cex = 0.8)
plot(0,0, type = "n", xlim=c(0,200), ylim = c(0,1),
    xlab = "Distance moved  m", ylab = "Cumulative distribution  F(r)")
lines (x, pkernel(x, "BVE", 30), col = "black", lwd = 1)
lines (x, pkernel(x, "BVE", 30, truncate = 200), col = "red", lwd = 2, lty = 2)
lines(attr(kfull20, "distribution"), col = "red")
mtext("Full", side=3, cex=0.7)

plot(0,0, type = "n", xlim=c(0,200), ylim = c(0,1),
    xlab = "Distance moved  m", ylab = "Cumulative distribution  F(r)")
lines (x, pkernel(x, "BVE", 30), col = "black", lwd = 1)
lines (x, pkernel(x, "BVE", 30, truncate = 200), col = "red", lwd = 2, lty = 2)
lines(attr(ksparse20, "distribution"), col = "red")
mtext('Sparse', side = 3, cex = 0.7)
```

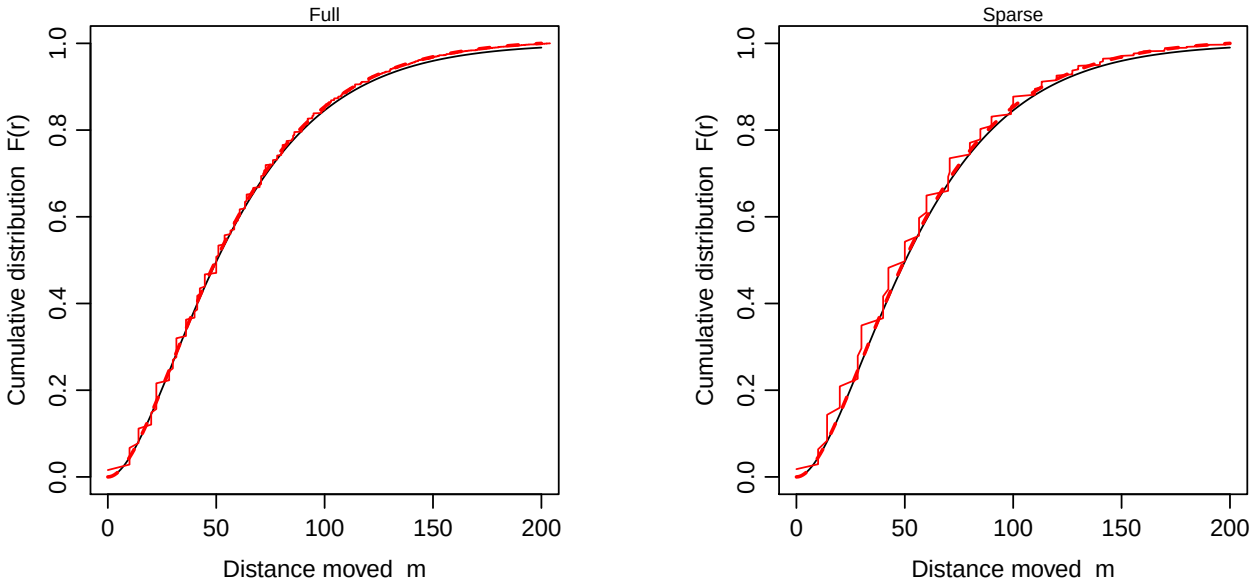


Figure 4. Cumulative distribution of BVE cell probabilities ($\alpha = 30$ m) for full and sparse discretized kernels of radius 200 m. Theoretical cumulative distributions of distance moved $F(r)$ are overlaid, both without truncation at kernel limit (black line) and with truncation (dashed red line).

Modifications

The radially symmetrical kernel is an idealised model for dispersal. Several modifications may be envisaged:

1. Allowance for edge effects
2. Habitat-selective settlement
3. Directional bias
4. Navigation around obstacles

These categories overlap somewhat. We do not wish to overcomplicate the model. Capture–recapture data generally include little information on movement, and other methods such as telemetry should be used if movement is the primary concern. Directional bias and navigation are therefore left aside for now.

Edge effects

Implementations of SECR usually consider a region of habitat with an outer boundary; mapped areas within the outer boundary may also be excluded as non-habitat. Movements drawn from a theoretical distribution will therefore often fall where the model has no way to handle them. To treat all such movements as mortality would be a biological distortion: actively dispersing animals are usually able to recognize and avoid non-habitat (out-of-habitat movements), and animals that cross an outer boundary (out-of-frame movements) may re-enter at a later time.

It is sometimes unclear in published open SECR models how out-of-frame and out-of-habitat transitions have been treated. The default in **openCR** >1.4 is to set the corresponding probabilities to zero (i.e. truncate the kernel at the boundary) and re-normalize the remainder of the discretized kernel (i.e. adjust other probabilities upwards). Internally this is done point by point across the habitat mask, as the neighbourhood of each mask cell is potentially unique. This conserves the relative frequency of destinations within the frame, while reducing the average distance moved and somewhat depleting the density along the boundary.

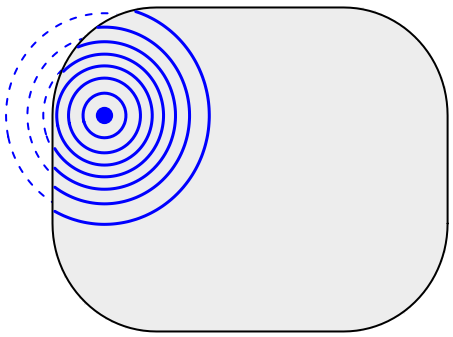


Figure 5. Schematic truncation and renormalization of movement probabilities (blue contours) when kernel overlaps the outer frame of the habitat mask (grey region).

In `openCR::openCR.fit()` the option is set with `edgemethod = "truncate"`, which is the default.

In `secr::sim.popn()` the same effect is achieved with `details = list(edgemethod = "truncate", ...)`[3]. For historical reasons, the default in `secr::sim.popn()` is `edgemethod = "wrap"`; this is unrealistic for open SECR modelling, and `edgemethod` should be set explicitly. Other possible values ('reflect' etc.) are unlikely to be useful; a possible exception in `sim.popn` is 'stop' that leaves potential out-of-frame movers at the point where they would otherwise have exited.

Selective settlement

Selective settlement implies adjustment of the movement kernel for each point of origin according to the values of a spatial covariate. Selective settlement has not been implemented in **secr** and **openCR**, except for potential out-of-habitat moves - see Edge effects. However, selective settlement has been implemented in some state space models (e.g., Bischof et al. 2020). The effect on summary statistics such as median distance moved and the probability of long-distance movement has yet to be explored.

References

Azzalini, A. and Genz, A. (2020) The R package 'mnormt': The multivariate normal and 't' distributions (version 2.0.2). URL http://azzalini.stat.unipd.it/SW/Pkg-mnormt/

Bischof, R. Milleret, C. , Dupont, P. Chipperfield, J., Tourani, M., Ordiza, A., de Valpine, P., Turek, D., Royle, J. A., Gimenez, O., Flagstad, Ø., Åkesson, M., Svensson, L., Brøseth, H., and Kindbergh, J. (2020) Estimating and forecasting spatial population dynamics of apex predators using transnational genetic monitoring. *Proceedings of the National Academy of Sciences, USA*. **117**, 30531–30538.

Clark, J. S., Silman, M., Kern, R., Macklin, E., and HilleRisLambers, J. (1999) Seed dispersal near and far: patterns across temperate and tropical forests. *Ecology* **80**, 1475–1494.

Cousens, R., Dytham, C. and Law, R. (2008) Dispersal in plants. Oxford University Press, Oxford U.K.

Efford, M. G. (submitted) Efficient discretization of movement kernels for spatiotemporal capture–recapture. *Journal of Agricultural, Biological and Environmental Statistics* (In review).

Ergon, T. and Gardner, B. (2014) Separating mortality and emigration: modelling space use, dispersal and survival with robust-design spatial capture–recapture data. *Methods in Ecology and Evolution* **5**, 1327–1336.

Gardner, B., Sollmann, R., Kumar, N. S., Jathanna, D. and Karanth, K. U. (2018) State space and movement specification in open population spatial capture–recapture models. *Ecology and Evolution* **8**, 10336–10344 https://doi.org/10.1002/ece3.4509.

[3] "truncate" here is an alternative to "normalize"; before **secr** 4.4.7 only "normalize" was recognised.

Genz, A., Bretz, F., Miwa, T., Mi, X., Leisch, F., Scheipl, F. and Hothorn, T. (2020) mvtnorm: Multivariate Normal and t Distributions. R package version 1.1-1. URL http://CRAN.R-project.org/package=mvtnorm.

Hofert, M. (2013) On sampling from the multivariate t distribution. *The R Journal* **5**, 129–136.

Hooten, M. B., Johnson, D. S., McClintock, B. T. and Morales, J. M. (2017) *Animal movement: statistical models for telemetry data.* CRC Press, Boca Raton.

Johnson, N. L., Kotz, S. and Balakrishnan, N. 1995. Continuous univariate distributions. 2nd ed. Wiley.

Kotz, S., Kozubowski, T. J., and Podgörski, K. (2001) The Laplace distribution and generalizations : a revisit with applications to communications, economics, engineering, and finance. Springer.

Kotz, S. and Nadarajah S. (2004) Multivariate t distributions and their applications. Cambridge University Press.

Nathan , R., Klein, E., Robledo-Arnuncio, J. J. and Revilla, E. (2012) Dispersal kernels: a review. In: J Clobert et al. *Dispersal Ecology and Evolution*. Oxford University Press. Pp 187–210.

Paquet, M., Arlt, D., Knape, J., Low, M., Forslund, P. and Pärt, T. (2020) Why we should care about movements: Using spatially explicit integrated population models to assess habitat source–sink dynamics. *Journal of Animal Ecology* **89**, 2922–2933.

Reidy, J. L., Thompson, F. R. III, Connette, G. M., and O'Donnell, L. (2018) Demographic rates of Golden-cheeked Warblers in an urbanizing woodland preserve. *Condor* **120**, 249–264.

Appendix 1. List of functions for manipulating movement kernels

Various functions are used to manipulate and summarise movement kernels in **secr** and **openCR**. We distinguish between the underlying continuous and optionally truncated probability distribution and the discretized and necessarily truncated form used for model fitting in **openCR**.

Continuous kernel

Package	Function	Description
openCR	`gkernel`	value(s) of 2-D pdf for specified movement model
	`dkernel`	probability density of distance moved
	`pkernel`	cumulative probability of distance moved
	`qkernel`	quantile function for distance moved
	`matchscale`	find scale parameter (α, move.a) of movement model with known expected distance moved or quantile of distance moved (e.g., median)
	`expected.d`	expected distance moved
	`cumMove`	2-D distribution of location post dispersal
secr	`sim.popn`	simulate multi-session population with turnover and movement
	`extractMoves`	extract between-session movements of surviving animals from popn object

Discretized kernel

Package	Function	Description
openCR	`openCR.fit`	fit open SECR model with movement
	`make.kernel`	construct kernel object (a form of habitat mask) and populate its covariate `kernelp` with probabilities
	`expected.d`	expected distance for kernel object
	`plot`	S3 method for kernel objects

Package	Function	Description
	`summary`	S3 method for kernel objects

Appendix 2. Kernel parameterization

This appendix relates parameterizations used by **openCR** to various sources.

Bivariate Laplace (BVE) kernel

From e.g. Clark et al. 1999 Eq 5a.

$$g(r) = \frac{1}{2\pi\alpha^2} \exp\left(\frac{-r}{\alpha}\right).$$

For distribution of distance moved, multiply pdf by $2\pi r$:

$$f(r) = \frac{r}{\alpha^2} \exp\left(\frac{-r}{\alpha}\right).$$

The normalisation coefficient can be checked using the standard integral $\int_0^\infty xe^{-ax}\,dx = \frac{1}{a^2}$ with $a = \frac{1}{\alpha}$.

A Gamma distribution with $k = 2$ has

$$f(x) = \frac{x}{\theta^2} e^{-\frac{x}{\theta}}$$

so we can draw random variates from rgamma

Bivariate t (BVT) kernel

Package mvtnorm provides a multivariate t distribution rmvt (see also Hofert 2013). Random draws with this function differ subtly from the naive rt for x,y (rmvt gives larger ratio Q90/Q50 for intermediate df 1.5-5)

From Kotz and Naradjah (2004 Eq. 1.1) see also Hofert (2013 Eq 4):

$$f_X(\mathbf{x}) = \frac{\Gamma(\frac{\nu+d}{2})}{\Gamma(\frac{\nu}{2})(\pi\nu)^{d/2}\sqrt{\det\Sigma}} \left(1 + \frac{(\mathbf{x}-\mu)^\top \Sigma^{-1}(\mathbf{x}-\mu)}{\nu}\right)^{-\frac{\nu+d}{2}}, \mathbf{x} \in R^d$$

As $\Gamma(x+1)/\Gamma(x) = x$, and given $\sqrt{\det\Sigma} = \sigma^2$, for 2 dimensions with $\mu = (0,0)$, $\mathbf{x} = (x,y)$ and $r^2 = x^2 + y^2$, this is

$$f_X(r) = g(r) = \frac{1}{2\pi\sigma^2}\left(1 + \frac{r^2}{\nu\sigma^2}\right)^{-(\frac{\nu}{2}+1)}, \; r \geq 0$$

In **secr**

$$g(r) = \frac{\beta}{\pi\alpha^2}\left(1 + \frac{-r^2}{\alpha^2}\right)^{-(\beta+1)}$$

Clark et al (1999 Eq 8) use

$$f_X(r) = g(r) = \frac{p}{\pi u\left(1 + \frac{r^2}{u}\right)^{p+1}}$$

where $p = \nu/2$

$$f_X(r) = g(r) = \frac{\nu}{2\pi u}\left(1 + \frac{r^2}{u}\right)^{-(\frac{\nu}{2}+1)}$$

which matches above if $u = \nu\sigma^2$.

Cousens et al. (2008) use

$$g(r) = \frac{a}{\pi b}\left(1 + \frac{r^2}{b}\right)^{-a-1}$$

which matches Clark et al. (1999) if $a = p$ and $b = u$.

Nathan et al. (2012) use

$$f_X(r) = g(r) = \frac{b-1}{\pi a^2}\left(1 + \frac{r^2}{a^2}\right)^{-b},\ a > 0,\ b > 1.$$

Using $b = \frac{\nu}{2} + 1$

$$f_X(r) = g(r) = \frac{\nu}{2\pi a^2}\left(1 + \frac{r^2}{a^2}\right)^{-(\frac{\nu}{2}+1)}$$

,

which matches if $a^2 = u = \nu\sigma^2$.

Hence for the MVTnorm and mnormt simulation functions, use $\Sigma = \begin{bmatrix} \text{move.a}^2/\nu & 0 \\ 0 & \text{move.a}^2/\nu \end{bmatrix}$ and $\nu = 2\,\text{move.b}$

Incidentally,

$$F(r) = 1 - \left(\frac{\alpha^2}{\alpha^2 + r^2}\right)^{\beta},$$

and

$$F^{-1}(u) = \sqrt{\alpha^2(u^{\frac{-1}{\beta}} - 1)}.$$