

movement in rtsetse

Andy South

2016-07-22

rtsetse is an R package that simulates tsetse fly populations on a spatial grid of different vegetation types to enable investigation of tsetse control.

Movement of tsetse flies is represented to replicate the methods used in Hat-trick an existing Excel based tsetse simulation.

This document shows how the rtsetse code works and can be used to represent movement at different levels of detail. You may ignore the grey code boxes if you want to see what the code can do rather than how it does it.

Flies are moved from each cell to it's 4 cardinal neighbours (NESW) and the following mechanisms are represented (ordered by increasing detail). These will be covered as separate sections in this document.

1. A set proportion move per time step
2. Reflecting boundaries (of the whole area)
3. Avoidance of 'no-go' areas
4. Movement rates based on vegetation type
5. Movement to poorer vegetation reduced
6. Adding all the movement mechanisms together

First load the rtsetse package after installing from github if necessary.

```
#require(devtools)  
#install_github('AndySouth/rtsetse', build_vignettes=TRUE)  
require(rtsetse)  
#these packages are used for plots in this document  
require(raster)  
require(sp)  
require(RColorBrewer)
```

The movement functions in rtsetse, accept a spatial matrix (y,x) defining the population of tsetse flies (of a particular age and gender). The function can be called repeatedly for different ages and sexes, allowing the movement rates to be age and sex dependent. For simplicity the examples shown here will just be for a single age and sex.

1. A set proportion move per time step

This example shows setting up a small test grid and running it for a few days with first a high and then a low movement proportion.

```
#set dimensions of the spatial grid  
nY <- nX <- 5  
nDays <- 3  
#create array to store grids over time  
a <- array(0, dim=c(nY, nX, nDays))  
#populate central cell of starting grid on day1  
a[3,3,1] <- 1
```

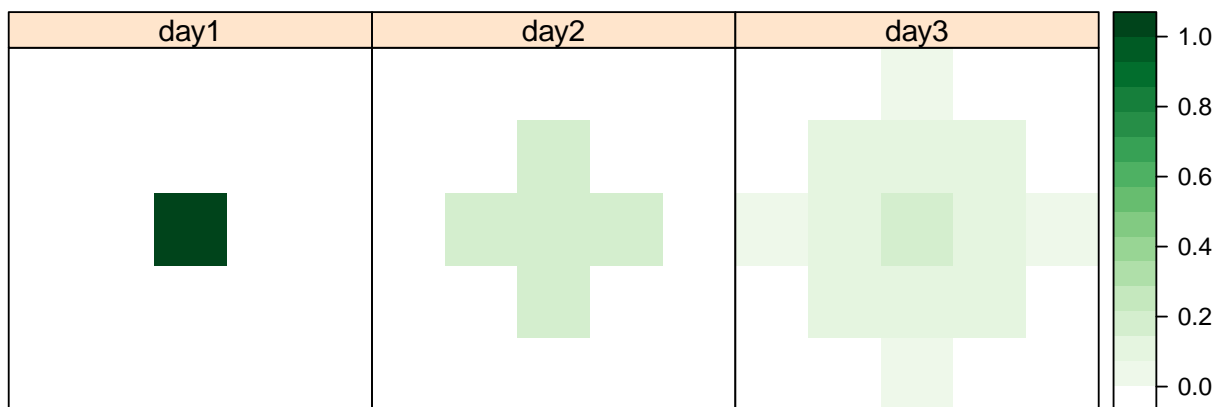
```

#set proportion moving
pMove <- 0.8

for(day in 2:nDays)
{
  #a[, ,day] <- rtMoveReflectNoGoVeg(a[, ,day-1], pMove=pMove)
  a[, ,day] <- rtMove(a[, ,day-1], pMove=pMove)
}

#quick way of displaying population spread over time
spplot( raster::brick(a, xmx=ncol(a), ymx=nrow(a)),
         names.attr=c(paste0("day",1:nDays)),
         col.regions=c("white",colorRampPalette(brewer.pal(9,"Greens"))(99)) )

```



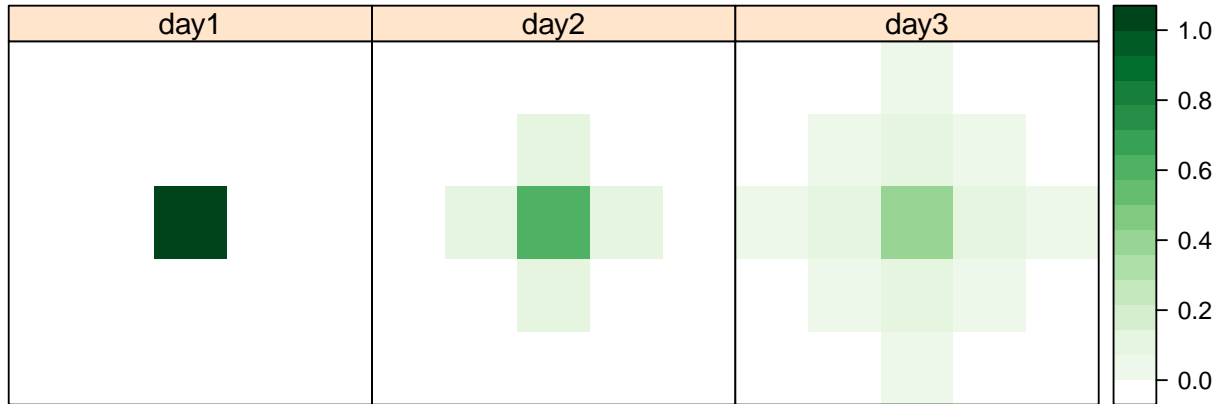
```

#repeat with lower movement proportion
pMove <- 0.4

for(day in 2:nDays)
{
  #a[, ,day] <- rtMoveReflectNoGoVeg(a[, ,day-1], pMove=pMove)
  a[, ,day] <- rtMove(a[, ,day-1], pMove=pMove)
}

#quick way of displaying population spread over time
spplot( raster::brick(a, xmx=ncol(a), ymx=nrow(a)),
         names.attr=c(paste0("day",1:nDays)),
         col.regions=c("white",colorRampPalette(brewer.pal(9,"Greens"))(99)) )

```



2. Reflecting boundaries (of the whole area)

Increasing the number of days we get to see what happens with flies at the boundaries of the simulated area. Each cell receives back the same number of movers as it gives out. This represents a likely situation where the vegetation and fly population neighbouring the boundary cells is similar to that in them. There is also an option within `rtsetse` to represent ‘island’ movement where no movers come back in from outside the area.

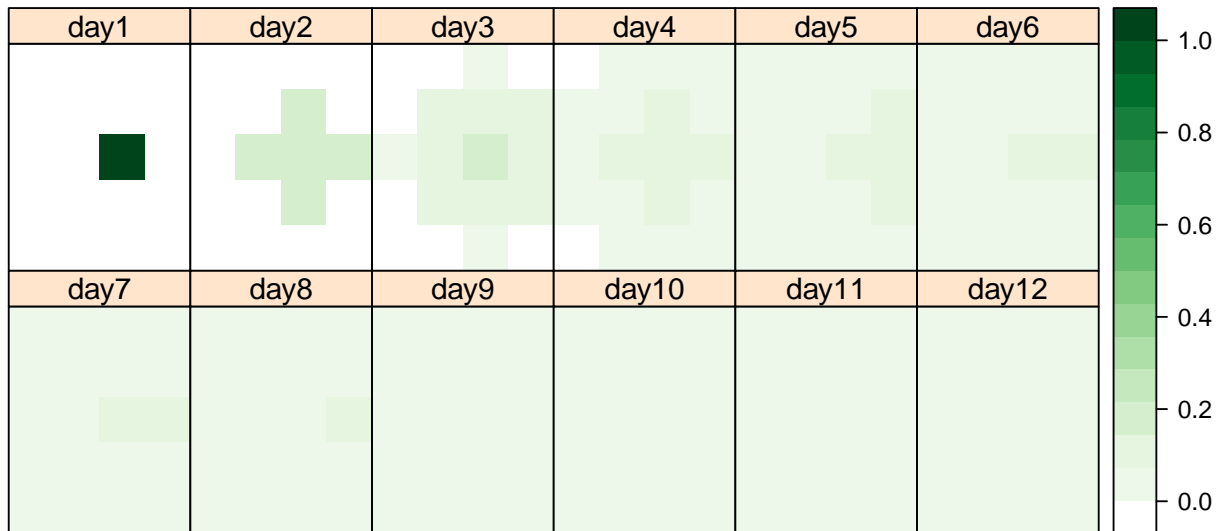
```
#set dimensions of the spatial grid
nY <- 5
nX <- 4
nDays <- 12
#create array to store grids over time
a <- array(0, dim=c(nY, nX, nDays))

#populate central cell of starting grid on day1
a[3,3,1] <- 1
#set proportion moving
pMove <- 0.8

for(day in 2:nDays)
{
  #a[, , day] <- rtMoveReflectNoGoVeg(a[, , day-1], pMove=pMove)
  a[, , day] <- rtMove(a[, , day-1], pMove=pMove)
}

#quick way of displaying population spread over time
spplot( raster::brick(a, xmx=ncol(a), ymx=nrow(a)),
```

```
names.attr=c(paste0("day",1:nDays)),
col.regions=c("white",colorRampPalette(brewer.pal(9,"Greens"))(99)) )
```



You can see that the reflecting movement results in the population becoming distributed equally over the grid.

3. Avoidance of ‘no-go’ areas

To avoid ‘no-go’ areas the code is identical to the above with the addition of a matrix for no-go areas. In this example the top row is set as a no-go area. You should be able to see that row remains unoccupied in the spread maps below.

```
nY <- 5
nX <- 4
nDays <- 5

#create array to store grids over time
a <- array(0, dim=c(nY, nX, nDays))

#creating empty no-go matrix
mNog <- array(1, dim=c(nY,nX))
#set first row to nogo (0)
mNog[1,] <- 0

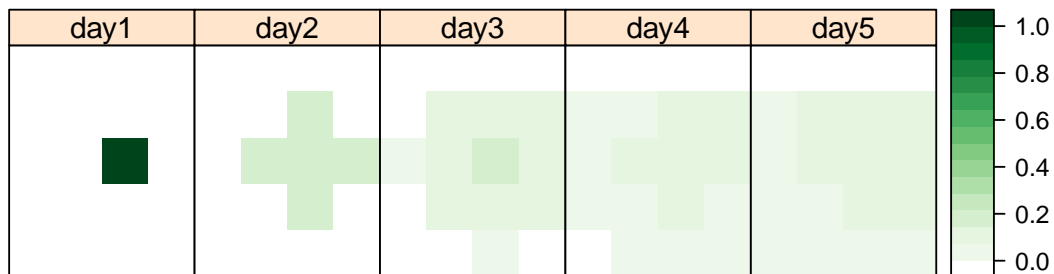
#populate central cell of starting grid on day1
a[3,3,1] <- 1
#set proportion moving
pMove <- 0.8
```

```

for(day in 2:nDays)
{
  #a[, ,day] <- rtMoveReflectNoGoVeg(a[, ,day-1], pMove=pMove, mNog=mNog)
  a[, ,day] <- rtMove(a[, ,day-1], pMove=pMove, mNog=mNog)
}

#display population spread over time
spplot( raster::brick(a, xmx=ncol(a), ymx=nrow(a)),
        names.attr=c(paste0("day",1:nDays)),
        col.regions=c("white",colorRampPalette(brewer.pal(9,"Greens"))(99)) )

```



4. Movement rates based on vegetation type

Within the Hat-trick tsetse simulation flies are assumed to move at different rates in different vegetation types. Flies move more quickly out of cells containing the vegetation types associated with faster movement. In rtsetse this is accommodated by allowing a movement multiplier to be applied to each cell. Where the movement multiplier is greater than 1 flies will move out faster, where it is less than 1 (but still greater than 0) flies will move out more slowly.

We can demonstrate this with a first simple example where a vegetation matrix is created with lower movement in the 2nd row, and the simulation is started with a uniformly populated grid. In the spread maps we can see this row increases in population while those around it decrease.

```

#create array to store grids over time
a <- array(0, dim=c(nY, nX, nDays))

#creating empty vegetation matrix
mVegMove <- array(1, dim=c(nY,nX))
#set 2nd row to lower movement
mVegMove[2,] <- 0.8

#fill starting grid on day1
a[, ,1] <- 1
#set proportion moving
pMove <- 0.8

for(day in 2:nDays)
{
  #a[, ,day] <- rtMoveReflectNoGoVeg(a[, ,day-1], pMove=pMove, mVegMove=mVegMove)
  a[, ,day] <- rtMove(a[, ,day-1], pMove=pMove, mVegMove=mVegMove)
}

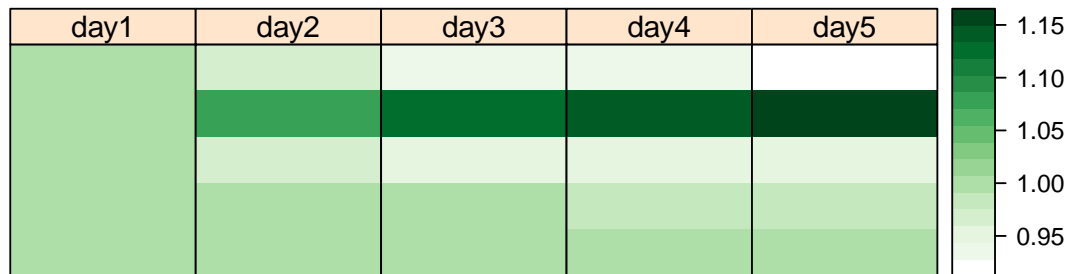
```

```

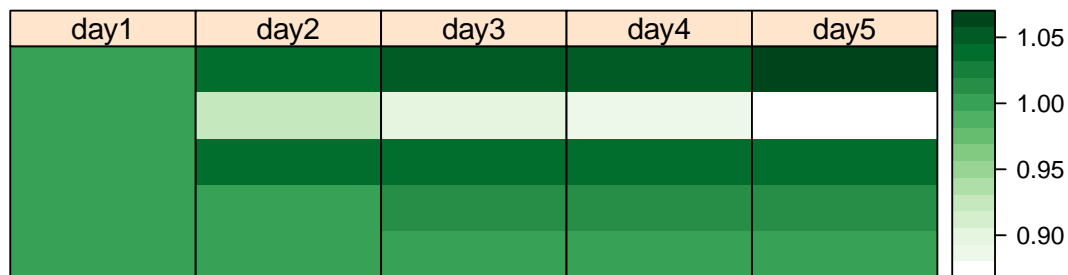
}

#display population spread over time
spplot( raster::brick(a, xmx=ncol(a), ymx=nrow(a)),
  names.attr=c(paste0("day",1:nDays)),
  col.regions=c("white",colorRampPalette(brewer.pal(9,"Greens"))(99)) )

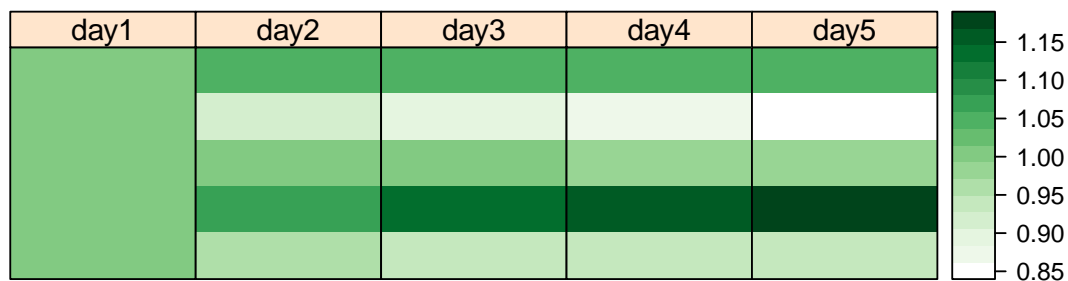
```



In a second example we can increase movement from the 2nd row and in the maps see that the population now becomes lower in that row.



In a third example we can keep the higher movement in the 2nd row and add lower movement in the 4th row. As expected the population declines in the former and increases in the latter.

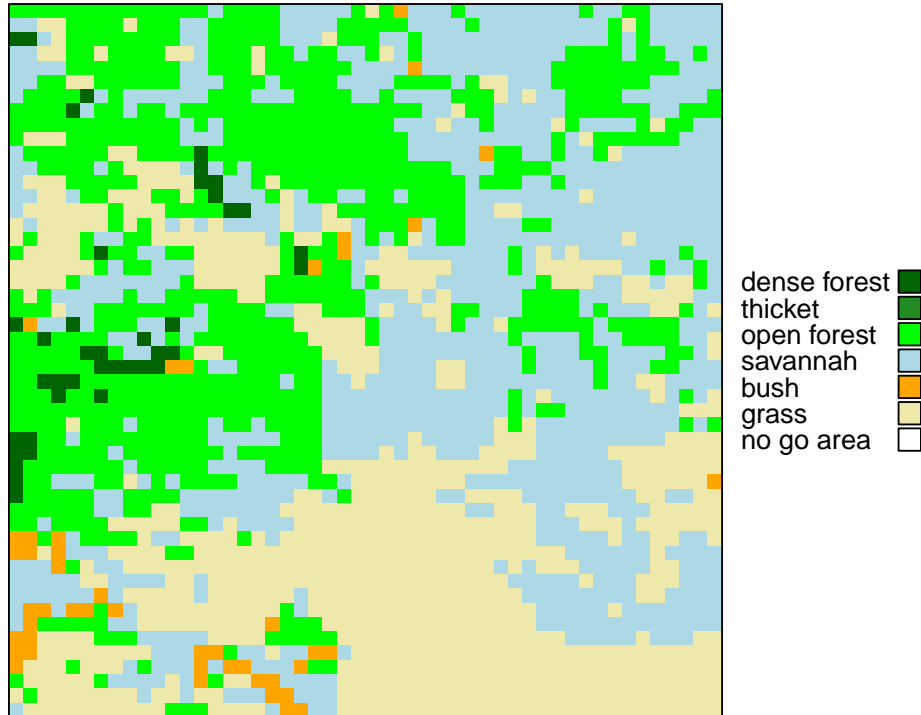


Now what happens if we apply this movement to the representation of a real landscape ? Lets first use the Serengeti map dataset stored in rtsetse (copied from Hat-trick). The cell size in this map is 1km, 80% of the flies are simulated as moving to one of the four neighbouring cells per day, thus representing a reasonable mean displacement of 800m per day seen in Savannah tsetse (Vale et al. 1984).

```

#get the vegetation map
mVegCats <- rtReadMapVeg( system.file("extdata","vegTanzaniaSerengetiTorrik1km.txt", package="rtsetse"))
#plot veg map
rtPlotMapVeg(mVegCats)

```



We can convert the vegetation categories to movement multipliers using the suggestions from Hat-trick for flies preferring Savannah and run the simulation. Note that the percentage of flies moving out of cells per day seems very high (88% in grass, 80% in savannah and 68% in woodland). Is this reasonable ?

```

#convert vegetation categories to movement multiplier values
dfMoveByVeg <- data.frame(code=c("D","T","O","S","B","G","N"),move=c(0.85, 0.9, 0.95, 1, 1.05, 1.1, 0))
mVegMove <- rtSetGridFromVeg( mVegetation=mVegCats, dfLookup=dfMoveByVeg )

nDays <- 12

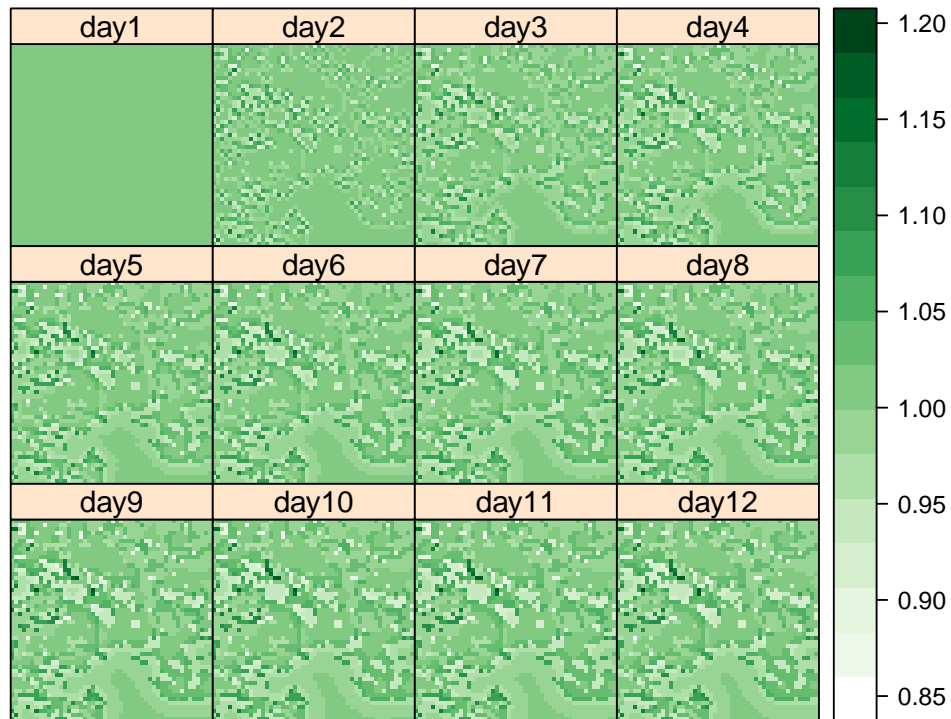
#create array to store grids over time, set dimensions from the veg grid
a <- array(0, dim=c(dim(mVegMove), nDays))

#fill starting grid on day1
a[,1] <- 1
#set proportion moving
pMove <- 0.8

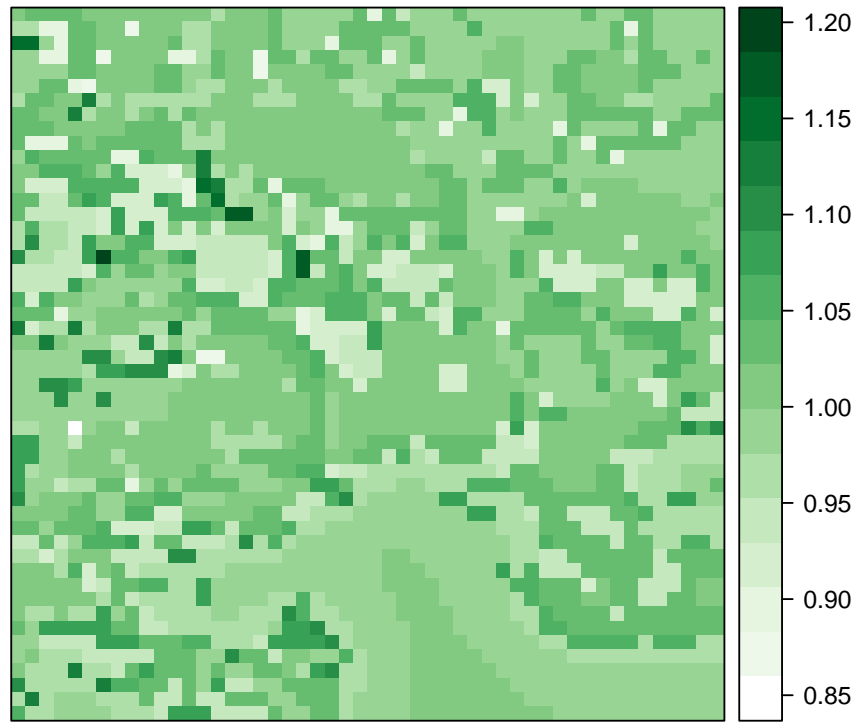
for(day in 2:nDays)
{
  #a[,day] <- rtMoveReflectNoGoVeg(a[,day-1], pMove=pMove, mVegMove=mVegMove)
  a[,day] <- rtMove(a[,day-1], pMove=pMove, mVegMove=mVegMove)
}

```

```
#display population spread over time
spplot( raster::brick(a, xmx=ncol(a), ymx=nrow(a)),
  names.attr=c(paste0("day",1:nDays)),
  col.regions=c("white",colorRampPalette(brewer.pal(9,"Greens"))(99)) )
```



```
#to look more closely at the final day
spplot( raster::raster(a[,nDays], xmx=ncol(a), ymx=nrow(a)),
  names.attr=c(paste0("day",nDays)),
  col.regions=c("white",colorRampPalette(brewer.pal(9,"Greens"))(99)) )
```

Remember that the maps above are just for a single age and sex class with movement only and no births and deaths.

5. Movement to poorer vegetation reduced

The results above also do not include any change in behaviour at vegetation boundaries. Hat-trick allows for movements to less preferred vegetation types to be decreased. The six potential vegetation types in Hat-trick are ordered by decreasing density of vegetation.

1. Dense Forest
2. Thicket
3. Open Forest
4. Savannah
5. Bush
6. Grass

The default names can be changed, but the ordering by density needs to remain.

The user chooses a preferred ('best') vegetation type for the flies being considered then movement is reduced if it would result in a decline in preference. As explained in Hat-trick :

If the fly is in a certain vegetation type and is headed to the same vegetation type, or to one that is better, ie, the best or a type that is closer to the best type, then the fly will keep going with no adjustment to the evacuation rate. Otherwise, the fly must be headed to a vegetation type that differs more markedly from the best than the one the fly is in. The fly will then tend to turn back, with a probability that is greater the more the vegetation type to which it is headed differs from the best. In deciding whether vegetataion is closer to the best, one that is X categories denser than the best is regarded as no worse than one that is X categories less dense than the best.

Movement resulting in a decline in preference of 1, 2, 3, 4, 5 categories is set to 30%, 10%, 3%, 1%, & 0.3% respectively of what it would have been otherwise.

Here is a first simple test with just 2 vegetation types.

```
#create very simple test veg map
nY <- 5
nX <- 4
# G for Grass, S for Savannah
#vertical split
m2 <- array("G", dim=c(nY,nX))
m2[1:(nY*nX/2)] <- "S"
#horizontal split
m2 <- array("G", dim=c(nY,nX))
m2[1:round(nY/2),] <- "S"

nDays <- 99

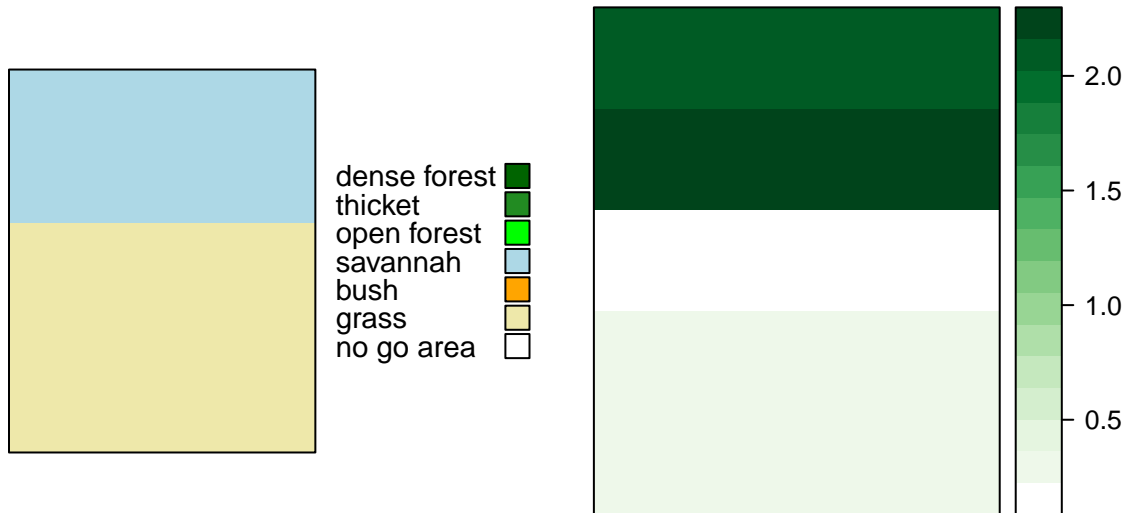
#create array to store grids over time, set dimensions from the veg grid
a <- array(0, dim=c(dim(m2), nDays))

#fill starting grid on day1
a[, ,1] <- 1
#set proportion moving
pMove <- 0.8
#set preferred vegetation type to Savannah(index=4)
iBestVeg <- 4

for(day in 2:nDays)
{
  #because mVegMove is not specified here, only the vegetation difference effect operates
  a[, ,day] <- rtMoveReflectNoGoVegBoundary(a[, ,day-1], pMove=pMove, mVegCats=m2)
  a[, ,day] <- rtMove(a[, ,day-1], pMove=pMove, mVegCats=m2, iBestVeg=iBestVeg)
}

#plot vegetation map
rtPlotMapVeg(m2)

#plot final day
spplot( raster::raster(a[, ,nDays], xmx=ncol(a), ymx=nrow(a)),
  names.attr=c(paste0("day",nDays)),
  col.regions=c("white",colorRampPalette(brewer.pal(9,"Greens"))(99)) )
```



Now trying with more vegetation types.

```
#create very simple test veg map
nY <- 6
nX <- 4

#create horiz stripes of habitat
m2 <- array(c("D","T","O","S","B","G"), dim=c(nY,nX))

#m2[1:round(nY/2),] <- "S"

nDays <- 12

#create array to store grids over time, set dimensions from the veg grid
a <- array(0, dim=c(dim(m2), nDays))

#fill starting grid on day1
a[, ,1] <- 1
#set proportion moving
pMove <- 0.6

for(day in 2:nDays)
{
  #because mVegMove is not specified here, only the vegetation difference effect operates
  #a[, ,day] <- rtMoveReflectNoGoVegBoundary(a[, ,day-1], pMove=pMove, mVegCats=m2)
  a[, ,day] <- rtMove(a[, ,day-1], pMove=pMove, mVegCats=m2, iBestVeg=iBestVeg)
}
```

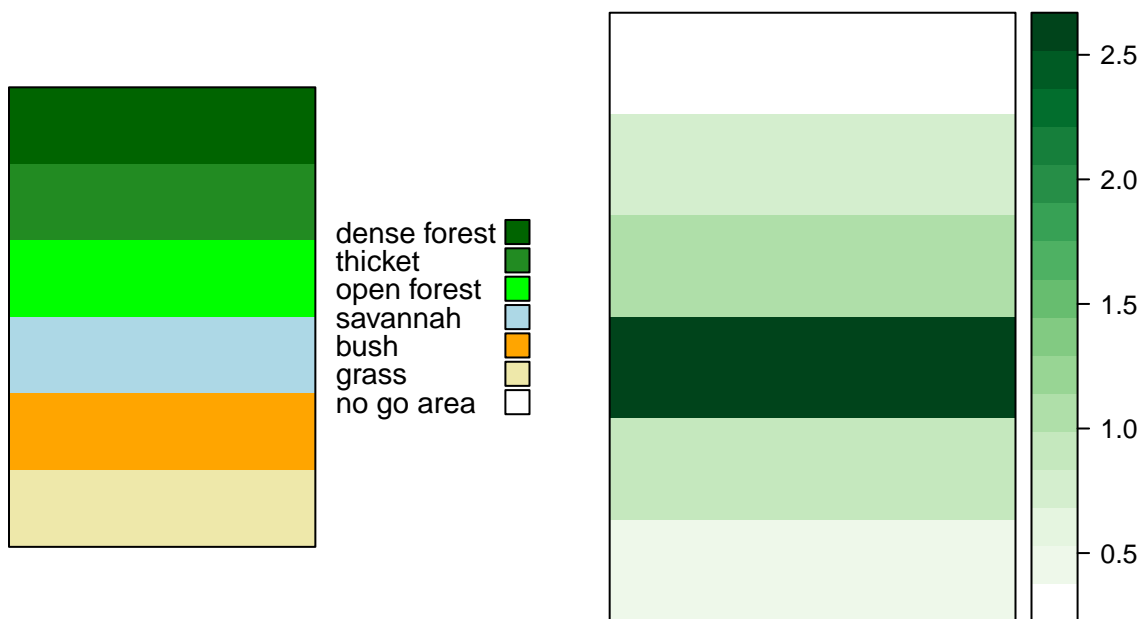
```

}

#plot vegetation map
rtPlotMapVeg(m2)

#plot final day
spplot( raster::raster(a[, , nDays], xmx=ncol(a), ymx=nrow(a)),
  names.attr=c(paste0("day", nDays)),
  col.regions=c("white", colorRampPalette(brewer.pal(9, "Greens"))(99)) )

```



The above plot shows accumulation of flies in the preferred habitat type as expected, and progressively fewer flies in the vegetation types that are more different from the preferred. Changing the order of the rows and the effect remains as expected.

```

#create very simple test veg map
nY <- 6
nX <- 4

#create horiz stripes of habitat
#m2 <- array(c("D", "T", "O", "S", "B", "G"), dim=c(nY, nX))
#change order
m2 <- array(c("O", "T", "D", "S", "G", "B"), dim=c(nY, nX))

nDays <- 12

#create array to store grids over time, set dimensions from the veg grid

```

```

a <- array(0, dim=c(dim(m2), nDays))

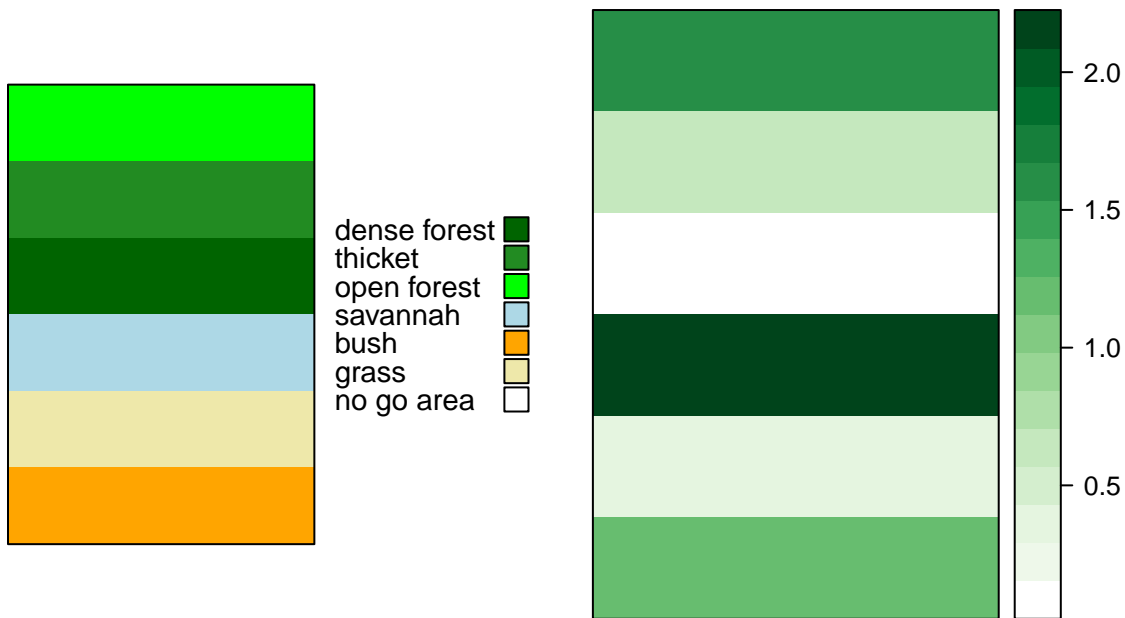
#fill starting grid on day1
a[,1] <- 1
#set proportion moving
pMove <- 0.6

for(day in 2:nDays)
{
  #because mVegMove is not specified here, only the vegetation difference effect operates
  #a[,day] <- rtMoveReflectNoGoVegBoundary(a[,day-1], pMove=pMove, mVegCats=m2)
  a[,day] <- rtMove(a[,day-1], pMove=pMove, mVegCats=m2, iBestVeg=iBestVeg)
}

#plot vegetation map
rtPlotMapVeg(m2)

#plot final day
splot( raster::raster(a[,nDays], xmx=ncol(a), ymx=nrow(a)),
        names.attr=c(paste0("day",nDays)),
        col.regions=c("white",colorRampPalette(brewer.pal(9,"Greens"))(99)) )

```



Below the decreased movement to poorer vegetation types alone is applied to the Serengeti map.

```

nDays <- 12

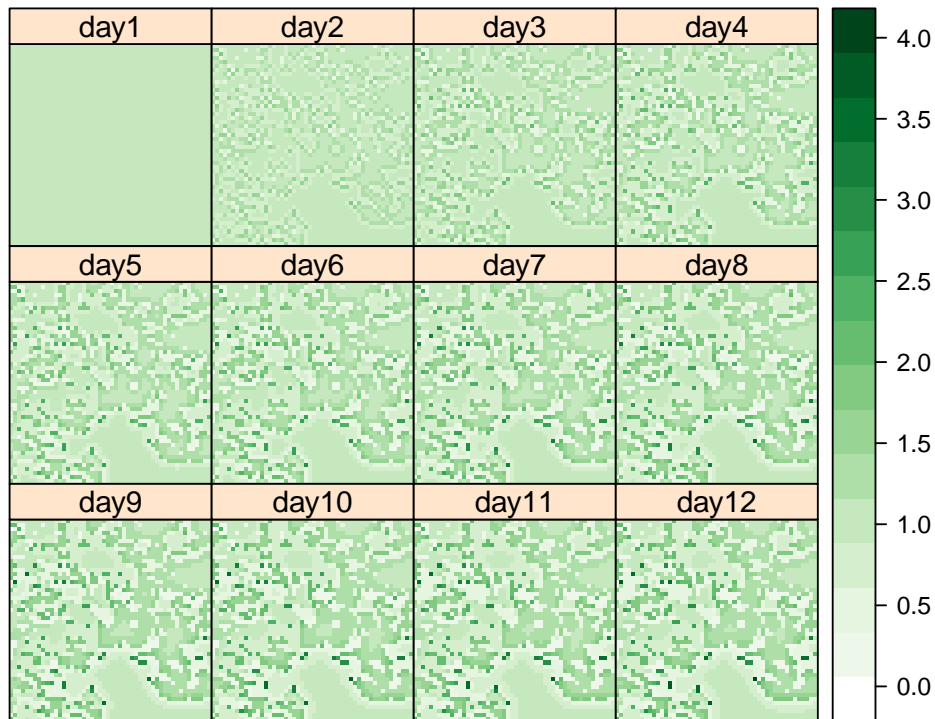
#create array to store grids over time, set dimensions from the veg grid
a <- array(0, dim=c(dim(mVegCats), nDays))

#fill starting grid on day1
a[, ,1] <- 1
#set proportion moving
pMove <- 0.8

for(day in 2:nDays)
{
  #because mVegMove is not specified here, only the vegetation difference effect operates
  #a[, ,day] <- rtMoveReflectNoGoVegBoundary(a[, ,day-1], pMove=pMove, mVegCats=mVegCats)
  a[, ,day] <- rtMove(a[, ,day-1], pMove=pMove, mVegCats=mVegCats, iBestVeg=iBestVeg )
}

#display population spread over time
spplot( raster::brick(a, xmx=ncol(a), ymx=nrow(a)),
        names.attr=c(paste0("day",1:nDays)),
        col.regions=c("white",colorRampPalette(brewer.pal(9,"Greens"))(99)) )

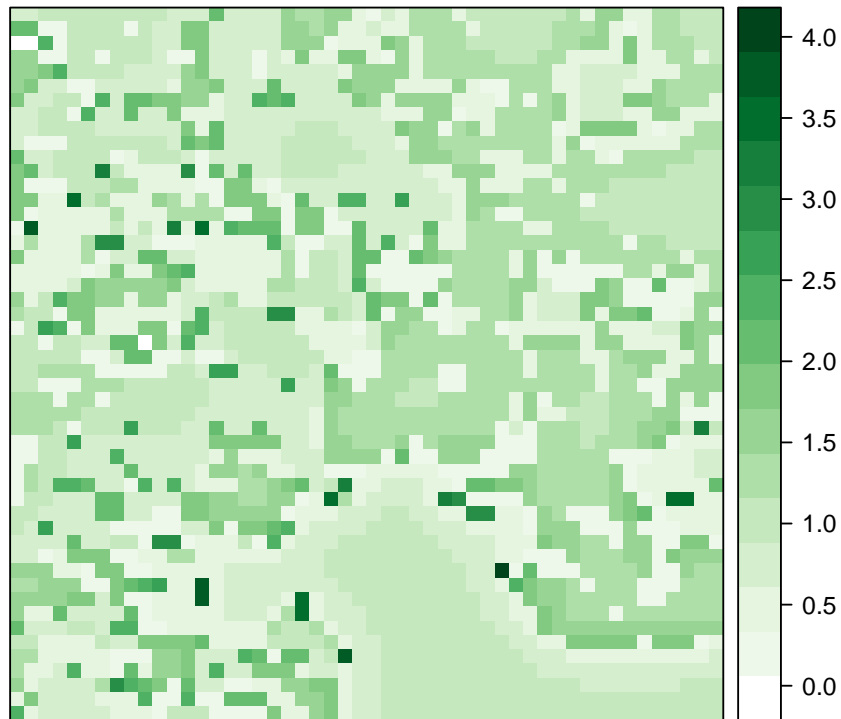
```



```

#to look more closely at the final day
spplot( raster::raster(a[, ,nDays], xmx=ncol(a), ymx=nrow(a)),
        names.attr=c(paste0("day",nDays)),
        col.regions=c("white",colorRampPalette(brewer.pal(9,"Greens"))(99)) )

```



As expected flies accumulate in the preferred habitat (savannah in this case). There is particular accumulation where preferred habitat cells are surrounded by less preferred. Here is a confirmation check of what happens on the lower right corner of the Serengeti map.

```
#get the vegetation map
mVegCats <- rtReadMapVeg( system.file("extdata","vegTanzaniaSerengetiTorri1km.txt", package="rtsetse"))
#subset SE corner
m2 <- mVegCats[40:50,40:50]

nDays <- 99

#create array to store grids over time
#set dimensions from the veg grid
a <- array(0, dim=c(dim(m2), nDays))

#fill starting grid on day1
a[, ,1] <- 1
#set proportion moving
pMove <- 0.8

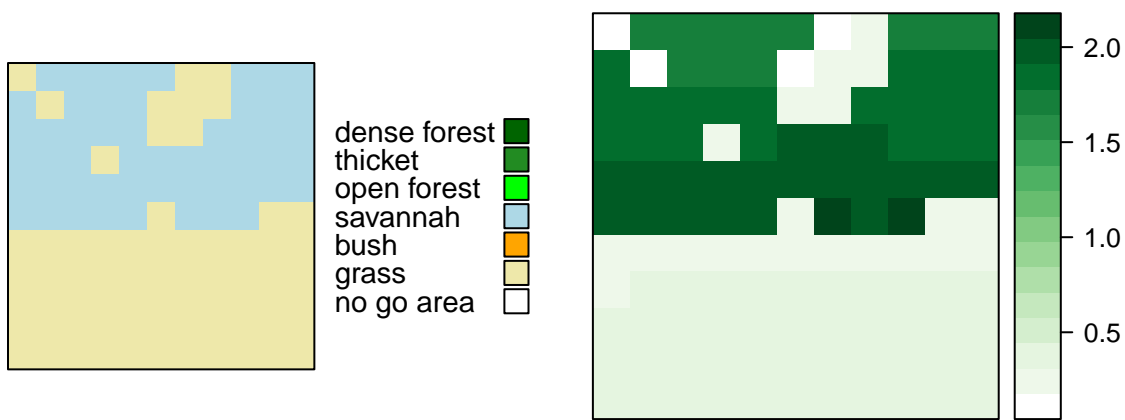
for(day in 2:nDays)
{
  #because mVegMove is not specified here, only the vegetation boundary effect operates
  #a[, ,day] <- rtMoveReflectNoGoVegBoundary(a[, ,day-1], pMove=pMove, mVegCats=m2)
  a[, ,day] <- rtMove(a[, ,day-1], pMove=pMove, mVegCats=m2, iBestVeg=iBestVeg )
}

#plot vegetation map
```

```
rtPlotMapVeg(m2)
```

```
#plot final day
```

```
spplot( raster::raster(a[, , nDays], xmx=ncol(a), ymx=nrow(a)),  
  names.attr=c(paste0("day", nDays)),  
  col.regions=c("white", colorRampPalette(brewer.pal(9, "Greens"))(99)) )
```



6. Adding all the movement mechanisms together

```
nDays <- 12
```

```
#create array to store grids over time
```

```
#set dimensions from the veg grid
```

```
a <- array(0, dim=c(dim(mVegCats), nDays))
```

```
#fill starting grid on day1
```

```
a[, , 1] <- 1
```

```
#set proportion moving
```

```
pMove <- 0.8
```

```
for(day in 2:nDays)
```

```
{
```

```
  #a[, , day] <- rtMoveReflectNoGoVegBoundary(a[, , day-1], pMove=pMove, mVegCats=mVegCats, mVegMove=mVegMove)
```

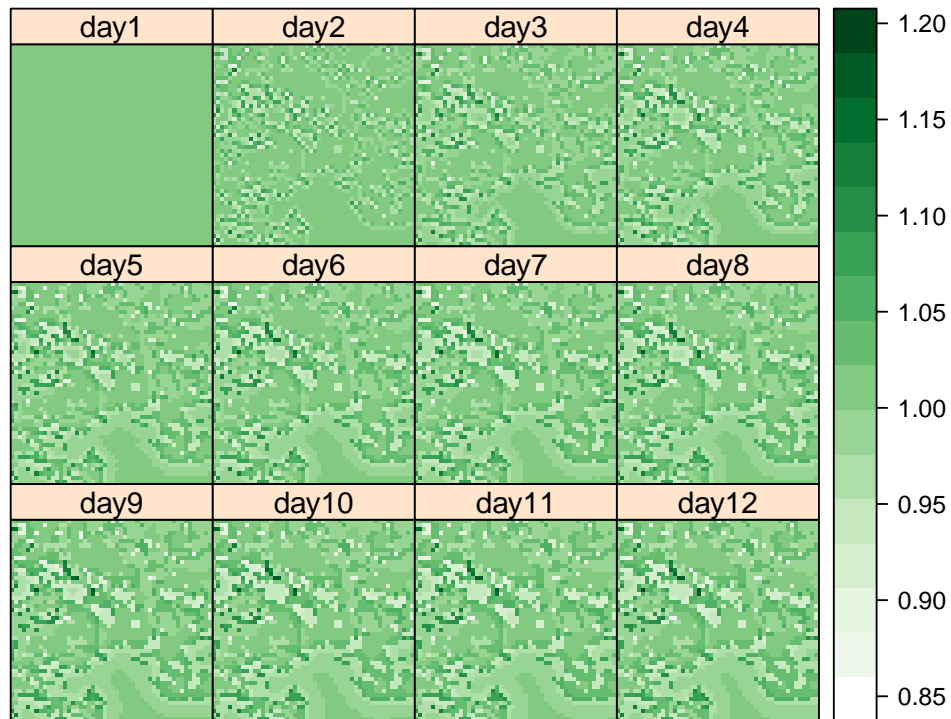


```

a[, , day] <- rtMove(a[, , day-1], pMove=pMove, mVegCats=mVegCats, mVegMove=mVegMove)
}

#display population spread over time
spplot( raster::brick(a, xmx=ncol(a), ymx=nrow(a)),
  names.attr=c(paste0("day", 1:nDays)),
  col.regions=c("white", colorRampPalette(brewer.pal(9, "Greens"))(99)) )

```



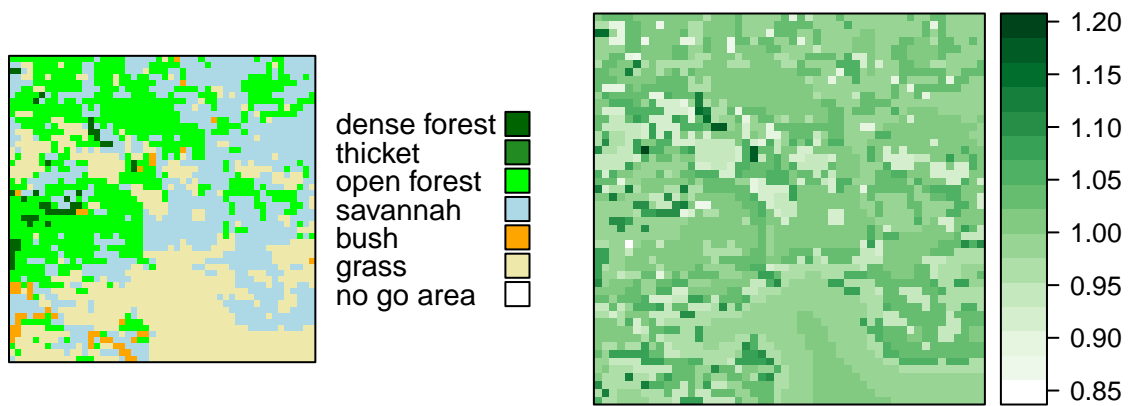
Now to compare the final day with the vegetation map.

```

#plot vegetation map
rtPlotMapVeg(mVegCats)

#plot final day
spplot( raster::raster(a[, , nDays], xmx=ncol(a), ymx=nrow(a)),
  names.attr=c(paste0("day", nDays)),
  col.regions=c("white", colorRampPalette(brewer.pal(9, "Greens"))(99)) )

```



Now that it seems these vegetation movement effects are working as expected I need to add them into the main simulation including births and deaths.