A tutorial for implementing Step Selection Function in R

P. Antkowiak[*]   H. Tripke[†]   C. Wilhelm[‡]

November 28, 2014

Contents

[*]M.Sc. programme "GIS und Umweltmodellierung" at University of Freiburg
[†]M.Sc. programme "Wildlife, Biodiversity and Vegetation" at University of Freiburg
[‡]M.Sc. programme "Wildlife, Biodiversity and Vegetation" at University of Freiburg

1   Introduction

1.1   Purpose and applications of SSFs

In addition to Resources Selection Functions (RSF) another powerful tool for evaluating data on animal movements and habitat selection are Step Selection Functions (SSF). The latter are used to estimate resource selection by comparing observed habitat use with available structures. Given GPS locations of a collared individual are connected by a linear segment. These segments are considered as steps. The fix rate of the GPS locations that influence the step length should be choosen carefully (i.e. by conducting a pilot study) to meet the requirements of the study questions, the target species and its behaviour. Then random steps are calculated by taking measured angle and distance along steps and using the observed positions as starting points. These alternative steps represent the available habitat, which could have been chosen, within a realistic step length of the observed positions. Finally, we can compare spatial attributes on both and test for effects that explain habitat selection by animals [2].

So far, SSF models were mainly done using Geospatial Modelling Environment (GME) that works with a GIS[1]. Moreover, many packages for analyzing animal movements are provided in R. None of these packages is designed for doing a SSF only or is at least missing fundamental steps. For example, the `adehabitatLT` has many features for analyzing your telemetry data (note that reading the help from 2011 is very much recommendend by us [1]!) but is missing any function to convert random steps. Another problem we faced when exploring the available packages in R was that some of them are lacking detailed descriptions on how to use their examples. The data for these examples is well prepared but for an user, who is freshly starting with SSF, it is hard to understand the structure of these data.

Therefore, the aim of this tutorial is to collect all these functions necessary to conduct a SSF and order and describe them in a way that intuitively makes you understand how to run a SSF with your own data. Each step will be explained using an exemplary dataset of GPS locations collected from seven Cougars (*Puma concolor*) in the year 2010 (in the following adressed as "xmpl"). Since the study design is fundemental to successfully implement SSFs, we recommend to read Thurfjell et al. (2014) [2], who provide different options and suggestions for conducting SSFs, before starting your own study.

1.2   Our SSF workflow in R

Figure 2 provides an overview of all necessary steps and potential options to conduct a SSF. This tutorial will guide you through each step and gives brief instructions on how to implement the functions and what to consider beforehand. To conduct a SSF using this tutorial we need you to store your initial data in two independant datasets:

1. A raster file of your spatial attributes (*Raster data*) and

2. GPS locations of your individuals assigned with a time stamp (*Waypoint data*).

We will start with the *Waypoint data* because these need to be transformed a couple of times to be able to work with them. You can find the single steps on the right site of Figure 2. While there are many options to adjust your *Waypoint data* the *Raster data* describing your spatial attributes needs not much of change. Once you created random steps for your observed positions you can extract the spatial attributes for each of those positions by using the function `extract`. At this point, *Waypoint* and *Raster data* will be combined and your final model can be written.
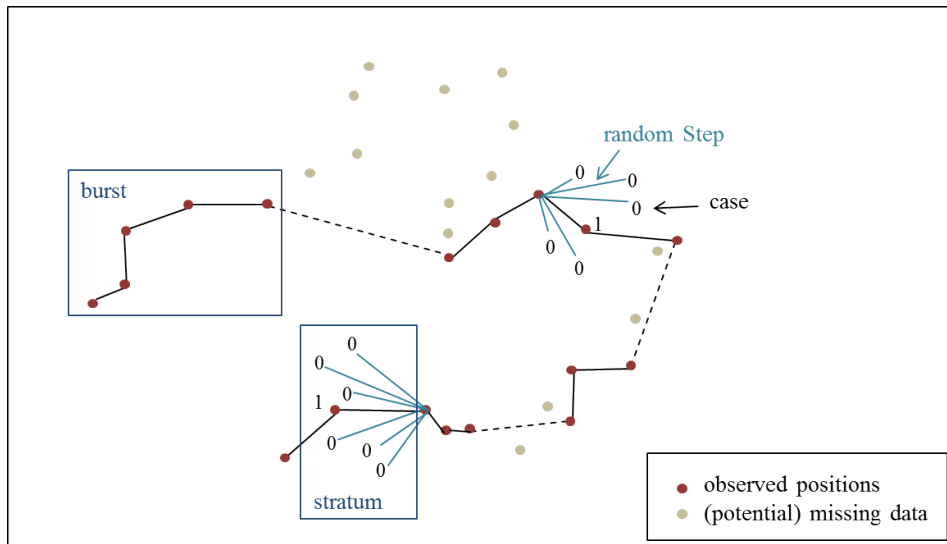
---

[1]www.spatialecology.com/gme/

Figure 1: Animal movements in an environment. Theses data are already separated into bursts of not more than 3h 10min intervals. This prohibits to combine positions uncared of missing data points. These bursts provide the angle and distance to calculate random steps from. Thereby, strata and case have to be defined.

1.3 Installing and loading Packages

Before you can actually start using this tutorial for conducting SSF you need to install a bunch of packages in R. For faster processing the `install.packages` function is deactivated.

```r
## for implementing SSF
install.packages("hab")
install.packages("hab", repos = "http://ase-research.org/R/") # regular
install.packages("hab", repos = "http://ase-research.org/R/",
                 type = "source")                    # for self-compiling


install.packages("adehabitatHR") # dealing with home ranges
install.packages("adehabitatHS") # habitat selection
install.packages("adehabitatLT") # trajectories
install.packages("adehabitatMA") # maps

install.packages("tkrplot")

# for handling raster data
install.packages("move")
install.packages("raster")
install.packages("rgdal")

# for analyzing the data
install.packages("mclogit")
install.packages("lme4")
install.packages("effects")
```
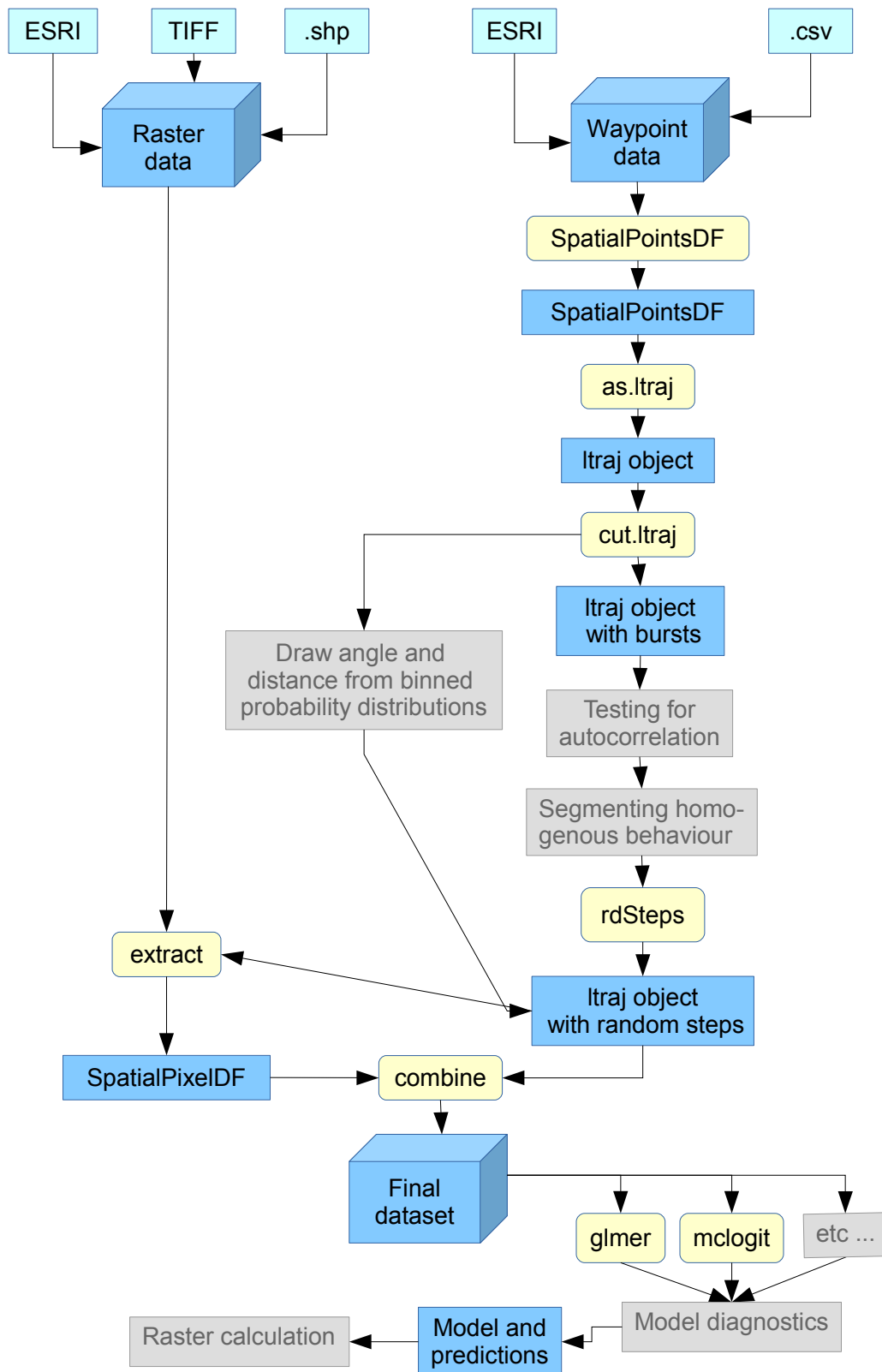
Figure 2: Stepwise conducting a Step Selection Function using existing R-packages. The yellow boxes show the name of the function applied while the blue boxes provide the type of object or data. In light grey optional steps are highlighted that are not implemented in this tutorial.

Keep fingers off the `adehabitat` package! It is outdated and replaced by four different packages designed for different analyses.

Loading packages:

```
library(hab)
library(adehabitatMA)
library(adehabitatHR)
library(adehabitatHS)
library(adehabitatLT)

library(sp)
library(raster)

library(mclogit)
library(lme4)
library(effects)
```

References

[1] Clement Calenge. Analysis of animal movements in r. 2011.

[2] Henrik Thurfjell, Simone Ciuti, and Mark S Boyce. Applications of step-selection functions in ecology and conservation. *Movement Ecology*, 2(1):4, 2014.