# SSFdata: Obtaining data to run Step Selection Function

**Description**
**File description**
**Input Files**
**Other input information**
**Running the script**
**Requirements**
**Output**
**Notes**
**Suggested Readings**

## Description

The code generates random movement steps representing alternative steps that could have been taken starting at the same origin point**.** A step is defined as a straight line joining two consecutive points. Alternative random steps are generated for each individual based on the frequency distribution of the step lengths and turning angles observed for the *remaining* individuals. The script provides information on the following characteristics along those movement steps: proportion of different habitat types within user-defined buffers, habitat types at start and end points, number of gaps crossed (optional), length of the step that takes place inside each habitat type and distance to a given feature (optional).

The amount of a given habitat type is calculated as percentage of the area of a buffer with user-defined radius surrounding the lines that is covered by different habitat types. Gaps are defined here as "non-habitat" areas rather than "non-habitat" between habitat patches (i.e., an extreme of a line that falls completely in an open area will count as a gap).

The script consists of 7 parts:

1) Generating a file with desired information from the original point file. It also adds information on the step number the point belongs to, the distance to the previous point, and the rotation angle between the lines before and after it.
2) Transform the txt into point shapefiles.
3) Make a summary table with the frequency distribution of the observed inter-step distance and rotation angles.
4) Generate random steps around each point based on the information of the frequency distributions of angles and distances.
5) Calculate the amount of the habitat types inside buffers surrounding the random steps.
6) Get information on gaps and habitat types being crossed.
7) Generate final file with the desired information.

Each of these steps can be skipped (only recommended once the previous steps have been ran).

**File description**

`SSF_v2.1.py` – Main code. This is the file that receives the input from the user
`SSFtools_v1_6.py` – Functions needed to run the main code.
`Trial.csv` – Example location dataset
`Example_Hbt.shp` – Example of habitat shapefile
`Imaginary_Streams.shp` – Example of line shapefile

**Input Files**

1. Csv file with information of locations points. Requirements:
    a. Separator has to be a comma (,)
    b. Input file can't have single records for any given individual
    c. The file has to have the following fields (in this order):
        i. IDA: unique identifier for each record
        ii. IDt: indicator of order of the records (*make sure that the input file is sorted by time and individual before running the code*)
        iii. Ind_ID: unique identifier for each individual animal
        iv. Date: date of observation (format: day-month-year)
        v. TimeArrival: time when the individual got to the location (24h format, e.g.: 16:12:03)
        vi. TimeDeparture: time when the individual left the location. If not existent, write the same value as arrival time
        vii. X: UTM x
        viii. Y: UTM y
2. Polygon shapefile with habitat types. Suggested columns:
    a. -Type of habitat coded with numbers (REQUIRED)
    b. -Type of habitat coded with characters (optional)
    c. -Column indicating which habitat types have to be considered matrix (optional)

**Other input information** (to be modified inside "SSF_v2.1.py"):

# BaseFolder= Folder where the results will go to(Process and Output folders will be created inside it)

e.g. : "C:/temp/SSFtrial/"

# TheInputPath= Indicate location of original point file

e.g.: "C:/temp/SSFtrial/Input/Trial.csv"

# HbtShpPath= Location of habitat type shapefile

e.g.: "C:/temp/SSFtrial/Input/terrain/for_matr_past_20140206.shp"

# UTMzone= Determine UTM zone (WGS 1984)
e.g.: "17N"
# Keyword= Indicate a name to identify the run (if not needed, write "")
e.g.: "SSFcleaning"
# size= Indicate the radius of the buffer that will surround the movement lines (in meters)
e.g.: 15
# RandSteps= Indicate how many random step lines will be produced (per observed step)
# Limited= Indicate if the records will be constrained by time and distance between points (True/False)
# MaxTime= Indicate upper time limit (minutes) Steps that took place within a time interval larger than this won't be used
# MinDist= Indicate lower distance limit (meters). Steps shorter than this won't be used to
# HbtNumeric= Indicate the name of the shapefile column with the numeric indicator of type of habitat
# HbtCharact= Indicate the name of the shapefile column with the names of the different types of habitat [Optional]
# CountGap= Indicate if the user desires to count number of gaps along the steps (True/False)
# MatrixID= Indicate which column in the habitat shp indicates matrix (with "Y" and "N")
# FeatDist= Indicate if the user desires to estimate distance of end point to a particular feature (True/False)
# FeatureShp= Indicate location of the feature shp
# Feature= Indicate which type of feature it is, as reference to assign names later
# WantLines= Indicate if you want shapefiles with the lines connecting the observed locations (True/False). If you select False, you will still get the lines but together with the randomly generated lines.

**To run this script:**

1) Open “SSF_v2.1.py” in a text editor (e.g.: Notepad)
2) Complete required fields in “User input” section and save. Take into account that:
    a. Folder or file paths have to be surrounded "" and have to have /, NOT \ as separators
    e.g.: “C:/temp” but NOT “C:\temp”
    b. The code is space-sensitive, so don’t create unnecessary spaces/tabs (in particular to the left of the existing text)
3) Open a Command Line window (type “cmd” in the Window's search box)
4) Change the directory to the one that contains the current version of Python being used. To do this type “cd” followed by the path to the location of your ArcGIS folder.
    e.g.:

cd C:\Python27\ArcGIS10.1

5) Run the script. This is done by typing "python" followed by the path to the script file, making sure to include the file name and its extension (.py).

   e.g:

   python C:\Temp\SSF_v2.1.py

By the end, the text in the cmd window should look something like this:

```
C:\Users\person>cd C:\Python27\ArcGIS10.1
C:\Python27\ArcGIS10.1>python C:\Temp\SSF_v2.1.py
```

You'll know that the script is running because lines will start to appear in the window. When it's done, a message will appear in the window letting you know.

**Requirements:**

Software:

- ArcGIS 10.1
- Geospatial Modelling Environment (GME) (http://www.spatialecology.com/gme)

Python modules (might be already installed):

- Shapefile.py (http://code.google.com/p/pyshp/)
- Numpy (http://www.numpy.org/)
- Matplotlib (http://matplotlib.org/)

**Output columns**

"..._SSFAllInfo.txt": contains the summary of the characteristics of the steps. It includes the following columns:

IDA = ID of the point at the beginning of the step
Ind_ID = ID of individual
StepID = ID of the step
Length = length of the step
Type = observed (1) or random (0)
[FeatureName]StepID = FID of nearest feature
[FeatureName]Dist = Distance to nearest feature
A_[i] = area of habitat i within buffer
P_[i] = percentage of buffer formed by habitat i
Segments = number of habitat segments that make up the step line
L_[#] = length of the step formed by habitat whose numeric code is "#"
StartMatrix = Indicates if starting point falls in the matrix (Y/N)
EndMatrix = Indicates if end point falls in the matrix (Y/N)

Gaps = Number of times the individual enters the matrix

Folder “SSFLineShp”: contains the line and buffer shapefiles of the observed and random steps.

Folder “MovLineShp” [Optional]: contains the line shapefiles of observed steps

**NOTES**:

- Be PATIENT. This process is VERY time consuming and requires high processing power.
- If you have a lot of data, you might want to sub-divide it and run the process in different multiple machines.
- If there is an error, it will appear in the cmd window and/or in error files in the Output and/or Process folder
- make sure the "-.py" files are in the same folder
- In general, not available data is represented as “NA” in text files and “-9999” in shapefiles
- Even when the code was built for ArcGIS 10.1, it MIGHT work for 10.0 as is. If not, send an email to Noelia Volpe. It will likely need only a few changes to make it functional.
- Make sure GME is installed inside “C:\Program Files (x86)\SpatialEcology\GME\SEGME.exe”. If not, you will have to change the path inside SSFtools_v_1_6.py)
- GME sometimes generates an error message when opening, which pauses the process until the user manually presses “okay” in the pop-up window. This doesn’t prevent obtaining results, but it means the user must keep an eye on it.
- The value “IDA” in the output line shapefiles is slightly different in the one that shows only observed lines (output of Part 2 of the code) and the one showing SSF lines (output of Part 4). In the former, the IDA of a line corresponds to the IDA of the END point of the movement step. In the latter, the IDA of a line corresponds to the IDA of the START point of the movement step.

For questions or problems contact: Noelia Volpe (noelia.volpe@fulbrightmail.org)

**Suggested readings**

Fortin, D., H. L. Beyer, M. S. Boyce, D. W. Smith, T. Duchesne, and J. S. Mao. 2005. Wolves influence elk movements: behavior shapes a trophic cascade in Yellowstone National Park. Ecology 86:1320–1330.

Gillies, C. S., H. L. Beyer, and C. C. St Clair. 2011. Fine-scale movement decisions of tropical forest birds in a fragmented landscape. Ecological Applications 21:944–954.