A quick guide to species distribution modelling with *fuzzySim*

A. Marcia Barbosa, CIBIO/InBIO - University of Evora (Portugal),
barbosa@uevora.pt

updated 7 May 2015

fuzzySim is an R package ultimately designed for calculating fuzzy similarity in species distributions. Meanwhile, it can also produce fuzzy species occurrence data to calculate fuzzy similarity from. Among the best methods to produce such fuzzy occurrence data is **generalized linear modeling of species presence-absence records**, which can provide both **occurrence probability** and **environmental favourability**. This tutorial will explore such modelling and its applications.

Installing and loading *fuzzySim*

The ***fuzzySim*** package works within the free and open-source R statistical software, so you first need to **download, install and open R** (available at http://www.r-project.org). In this tutorial, in monospaced font are the commands that you need to type (or copy and paste) into the R console (and then press the *enter* key to execute them). For commands that generate visible results in R, these are usually shown below them, preceded by hash marks (##). Note that all **commands are case-sensitive**, so you must respect upper- and lower-case letters; that you must always use **straight** (', ") **rather than curly quotes and apostrophes**; and that **R is only ready to receive a new command when there's a prompt sign (>) at the end of the R console**; if not, it's still waiting for an operation to be finished or for you to complete a previous command -- watch out for unclosed parentheses or such.

Install *fuzzySim* by pasting the command below in the R console (when connected to the internet):

```
install.packages("fuzzySim", repos = "http://R-Forge.R-project.org")
```

This should work if you have the **latest version of R**; otherwise, it may either fail (producing a message like "*package* '*fuzzySim*' *is not available for your R version*") or show a warning and install an older version of *fuzzySim*. To **check the version that you have actually installed**, type citation(package="fuzzySim"). To install the latest version of the package, you can either upgrade R *or* download the compressed *fuzzySim* package **source files** to your disk -- *.zip* for Windows or *.tar.gz* for Linux and Mac, available at the package development page or at this Dropbox folder, and then install the

package locally, e.g. with R menu "*Packages - Install packages from local zip files*" (Windows), or "*Packages & Data - Package installer, Packages repository - Local source package*" (Mac), or "*Tools - Install packages - Install from: Package Archive File*" (RStudio).

You only need to install the package once (unless a new version becomes available), but you need to **load it every time you open a new R session** in which you intend to use *fuzzySim* (no need for an internet connection anymore), by getting it from your R library with the following command in R:

```
library(fuzzySim)
```

Preparing and modelling the data

For species distribution modelling, you'll need a table with species presences and absences in *wide* format, i.e., one species per column and their presences and absences as ones and zeros; if your data are in *long* format, with all species in the same column, check out help(splist2presabs) for a way to convert them. You'll also need the values of a set of predictor variables to use in the model(s). For an example of how your data should be organised, look at the *rotif.env* sample dataset that comes with *fuzzySim* (a global dataset of rotifer distribution records published with this article). The following command will load this dataset in your R session:

```
data(rotif.env)
```

You can get more information on this dataset with the following command, which should open an R Documentation window:

```
help(rotif.env)
```

Now **look at the first rows** of this dataset:

```
head(rotif.env)
```

```
##    TDWG4   LEVEL_NAME          REGION_NAME       CONTINENT      Area
## 1 ABT-OO      Alberta        Western_Canada NORTHERN_AMERICA 663485.40
## 2 AFG-OO  Afghanistan          Western_Asia   ASIA-TEMPERATE 641921.77
## 3 AGE-BA Buenos_Aires Southern_South_America SOUTHERN_AMERICA 306187.95
## 4 AGE-CH       Chaco Southern_South_America SOUTHERN_AMERICA  99203.11
## 5 AGE-CN   Corrientes Southern_South_America SOUTHERN_AMERICA  88614.06
## 6 AGE-ER   Entre_Rios Southern_South_America SOUTHERN_AMERICA  78071.93
##   Altitude AltitudeRange HabitatDiversity HumanPopulation  Latitude
## 1   769.07         3346              12        3461492  54.95520
## 2  1797.41         6347              13       32755566  33.78802
## 3    92.66         1092              12       15548773 -36.64692
## 4   115.57          230               7        1090382 -26.38870
## 5    67.60          195               9        1029757 -28.75806
## 6    44.22          125               9        1296896 -32.03426
```

```
##    Longitude Precipitation PrecipitationSeasonality TemperatureAnnualRange
## 1 -114.45960       454.96                  52.23               454.56
## 2   65.98809       309.59                  92.11               403.11
## 3  -60.54985       813.76                  29.92               272.70
## 4  -60.76430       935.89                  57.13               257.05
## 5  -57.78881      1292.63                  28.18               236.53
## 6  -59.20174      1059.91                  31.85               256.20
##   Temperature TemperatureSeasonality UrbanArea Abrigh Afissa Apriod Bangul
## 1       0.429              11465.98     1085      0      0      1      1
## 2      11.728               8812.06      790      1      0      1      1
## 3      15.055               5040.31        0      1      1      0      1
## 4      21.847               4147.56        0      0      0      0      1
## 5      20.720               4192.44        0      0      0      0      0
## 6      18.215               4637.56        0      0      0      0      0
##   Bcalyc Bplica Bquadr Burceo Cgibba Edilat Flongi Kcochl Kquadr Ktropi
## 1      0      0      0      0      0      0      1      1      1      0
## 2      1      1      1      1      1      0      0      1      1      1
## 3      1      1      1      1      0      1      1      1      1      1
## 4      0      0      1      0      1      0      1      0      0      1
## 5      1      1      1      0      0      1      1      0      1      0
## 6      1      0      1      0      0      1      1      1      0      1
##   Lbulla Lclost Lhamat Lluna Llunar Lovali Lpatel Lquadr Mventr Ppatul
## 1      0      0      0      1      1      1      0      0      0      1
## 2      1      1      1      1      0      1      1      1      0      1
## 3      1      1      1      1      1      1      0      1      1      1
## 4      1      1      1      0      1      1      0      1      1      1
## 5      0      0      0      1      0      1      0      0      1      1
## 6      1      0      0      0      0      0      0      0      0      1
##   Pquadr Pvulga Specti Tpatin Tsimil Ttetra
## 1      0      1      1      0      0      1
## 2      1      1      0      1      1      0
## 3      1      1      0      1      0      1
## 4      1      1      0      0      1      1
## 5      1      1      0      0      1      0
## 6      1      0      0      1      0      0
```

Show the **column names** of this dataset, to see which columns contain the species data and which contain the variables:

```
names(rotif.env)
```

```
##  [1] "TDWG4"                  "LEVEL_NAME"
##  [3] "REGION_NAME"            "CONTINENT"
##  [5] "Area"                   "Altitude"
##  [7] "AltitudeRange"          "HabitatDiversity"
##  [9] "HumanPopulation"        "Latitude"
## [11] "Longitude"              "Precipitation"
## [13] "PrecipitationSeasonality" "TemperatureAnnualRange"
## [15] "Temperature"            "TemperatureSeasonality"
```

```
## [17] "UrbanArea"              "Abrigh"
## [19] "Afissa"                "Apriod"
## [21] "Bangul"                 "Bcalyc"
## [23] "Bplica"                "Bquadr"
## [25] "Burceo"                 "Cgibba"
## [27] "Edilat"                "Flongi"
## [29] "Kcochl"                 "Kquadr"
## [31] "Ktropi"                "Lbulla"
## [33] "Lclost"                "Lhamat"
## [35] "Lluna"                 "Llunar"
## [37] "Lovali"                "Lpatel"
## [39] "Lquadr"                 "Mventr"
## [41] "Ppatul"                "Pquadr"
## [43] "Pvulga"                 "Specti"
## [45] "Tpatin"                "Tsimil"
## [47] "Ttetra"
```

You can see that predictor variables are in columns 5 to 17 and species presence/absence data are in columns 18 to 47. You can **get distribution models for multiple species simultaneously** with the *multGLM* function of *fuzzySim*. You must specify the name of the dataset, the index numbers of the columns containing the species data and the variables, and optionally the index number of the column containing the row identifiers. There are a number of additional options on how to select variables for the models; the following command will create an object named *rotif.mods* containing models built for each of these species with the defalt *multGLM* settings:

```
rotif.mods <- multGLM(data = rotif.env, sp.cols = 18:47, var.cols = 5:17, id.col = 1)
```

The object output by *multGLM* is a list containing two elements: another list named *models* (which you can call by typing rotif.mods$models), and a dataframe with the resulting *predictions* (rotif.mods$predictions). Let's check out one of the models, for example the first one in the list:

```
summary(rotif.mods$models[[1]])
```

```
##
## Call:
## glm(formula = Abrigh ~ HabitatDiversity + TemperatureSeasonality +
##     Area + Precipitation + PrecipitationSeasonality + HumanPopulation,
##     family = binomial)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -1.7438  -0.9705  -0.5980   1.0993   1.9609
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)            -2.224e+00  1.003e+00  -2.218 0.026539 *
```

```
## HabitatDiversity         3.558e-01  9.173e-02   3.879 0.000105 ***
## TemperatureSeasonality   -2.197e-04  5.342e-05  -4.112 3.93e-05 ***
## Area                     6.668e-07  2.735e-07   2.439 0.014744 *
## Precipitation           -6.174e-04  2.554e-04  -2.418 0.015611 *
## PrecipitationSeasonality -1.208e-02  4.935e-03  -2.448 0.014350 *
## HumanPopulation          1.141e-08  5.400e-09   2.112 0.034660 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 387.85  on 290  degrees of freedom
## Residual deviance: 344.63  on 284  degrees of freedom
## AIC: 358.63
##
## Number of Fisher Scoring iterations: 4
```

You can also call a model by the name of the species in the dataset (result will be the same as above):

```
summary(rotif.mods$models[["Abrigh"]])
```

Now check out the first rows of the *predictions* dataframe included in *rotif.mods* (results not shown here):

```
head(rotif.mods$predictions)
```

Additional modelling options

Let's now take a closer look at the different modelling options available in *multGLM*. If you type help(multGLM), under Usage, you can see which are the default parameters:

```
multGLM(data, sp.cols, var.cols, id.col = NULL, family = "binomial", test.sample = 0,
FDR = FALSE, step = TRUE, trace = 0, start = "null.model", direction = "both",
Y.prediction = FALSE, P.prediction = TRUE, Favourability = TRUE, group.preds = TRUE,
trim = TRUE)
```

So, the command above should produce the same results as the *multGLM* command executed before, where most of these arguments were not speficied explicitly. The first three arguments (*data*, *sp.cols* and *var.cols*) do not have default values, so they always need to be specified by the user; but the remaining parameters have their default values set, so for example you can keep *id.col* as *NULL* if you don't have or don't want to use an ID column. The *family* argument currently has only one option available in *multGLM*, so the function will produce an error message if you try to specify a different one.

The ***test.sample*** argument is 0 by default, but it can be increased if you want part of the data to be reserved for further testing of the model, and thus not

used for model building. You can specify either a value between 0 and 1, for a **proportion** of the data to choose randomly (e.g. 0.2 for 20%); an integer number, for a particular **number of cases** to choose randomly among the rows in data; a vector of integers, for the **index numbers** of the particular rows to set aside; or **"Huberty"**, for his **rule of thumb** on how many data should be set aside based on the number of variables.

The ***FDR*** argument, which is FALSE by default, indicates whether there should be a pre-selection of variables based on the significance of their bivariate relationship with the species' occurrence. If you set it to TRUE, the *FDR* function is called automatically -- see help(FDR) for more info on the procedure, though you don't need to use this function directly.

The ***step*** argument, which is TRUE by default, defines whether variables should be included in the models with a stepwise selection procedure based on Akaike's Information Criterion (AIC), using the *step* function of R. The three following arguments are relevant only when *step* = *TRUE*: *trace* shows (or not, if FALSE) the intermediate results of the stepwise inclusion of model variables; ***start*** defines whether the inclusion of the variables should start forward (with "null.model") or backward (with "full.model)"; and *direction* specifies in which direction the variable selection should proceed ("forward", "backward", or "both"; see help(step) for more info.)

Arguments ***Y.prediction***, ***P.prediction*** and ***Favourability*** define the type of predictions you want in the output *predictions* table. *Y* (FALSE by default) is the prediction in the scale of the predictor variables (i.e. the logit equation); *P* is the prediction in the scale of the response variable (i.e. probability, varying between 0 and 1); and *Favourability* is the prevalence-independent version of probability (also between 0 and 1), which can be directly compared across species (see Details in help(Fav) for more info).

Further steps

You can **analyse these models and evaluate their performance** with the ***modEvA*** **R package**, which is also available on R-Forge together with another short tutorial.

You can **apply all these models at once to a different dataset** containing the same variables (with the same names) but for another region or time period: see the Examples in help(getPreds) for how to do this.

If you've calculated Favourability, which is directly comparable among species, you can then use **fuzzy logic** to **combine different models**: for example, minimum favourability is the favourability for the simultaneous occurrence of all species in a given set; maximum favourability is the favourability for occurrence of at least one of the species in the given set (see e.g. this article

for illustrated details). The following commands will calculate these values and add them to the *predictions* table:

```
names(rotif.mods$predictions)  # see which are the favourability (_F) columns; 32:61 in this case
rotif.mods$predictions$Fav_all <- min(rotif.mods$predictions[ , 32:61])
rotif.mods$predictions$Fav_any <- max(rotif.mods$predictions[ , 32:61])
```

Saving your results

You can save your model predictions to disk, for example in CSV format, with the following command:

```
write.csv(rotif.mods$predictions, file = "predictions.csv")
```

Modelling your own data

If you want to try this out with your own species data, get the data into R -- e.g. save your **table in a text file**, with **column names in the first row** and columns **separated by tabulators**, name the file *mydata.txt*, save it in your R working directory (type getwd() to find out where it is), and then import it to R with the following command:

```
mydata <- read.table("mydata.txt", header = TRUE, sep = "\t")
```

Then reproduce the operations above, but replacing *rotif.env* with *mydata* (or whatever name you've assigned in the above command) and specifying your column index numbers accordingly. If you use *fuzzySim* in publications, please **cite the paper**:

Barbosa A.M. (2015) fuzzySim: applying fuzzy logic to binary similarity indices in ecology. *Methods in Ecology and Evolution*, in press (DOI: 10.1111/2041-210X.12372).

That's it! You can e-mail me with any suggestions or concerns, but first remember to check for updates to the package or this tutorial at http://fuzzysim.r-forge.r-project.org. This tutorial was built with *RStudio* + *rmarkdown* + *knitr*. Thanks!