Running Structure-like Population Genetic Analyses with R

Olivier François
olivier.francois@grenoble-inp.fr

June 2016

Summary

This short tutorial explains how population structure analyses reproducing the results of the widely-used computer program STRUCTURE can be performed using commands in the R language. The method works for any operating systems, and it does not require the installation of STRUCTURE or additional computer programs. The R program allows running population structure inference algorithms, choosing the number of clusters, and showing admixture coefficient bar-plots using a few commands. The methods used by R are fast and accurate, and they are free of standard population genetic equilibrium hypotheses. In addition, these methods allow their users to play with a large panel of graphical functions for displaying pie-charts and interpolated admixture coefficients on geographic maps.

Keywords: Population genetics, Population structure, Admixture coefficients, Graphical displays, maps, R language.

O. François (2016) Running Structure-like Population Genetic Analyses with R. R Tutorials in Population Genetics, U. Grenoble-Alpes, pages 1-9.

1 Introduction

Estimating and visualizing population genetic structure using the Bayesian computer algorithm STRUCTURE, the spatial clustering program TESS, or the maximum-likelihood algorithm ADMIXTURE are commonly performed with several stages (Pritchard et al. 2000, Chen et al. 2007, Alexander et al. 2011). The typical steps of a population structure analysis include running inference algorithms, choosing the number of clusters, showing admixture bar-plots and displaying pie-charts on geographic maps. Each of these steps requires the user to install and run independent computer packages.

This tutorial explains how those analyses can be performed in a simple way and within a single framework by using the R computer package (R Core Team 2016). Population genetic structure analyses using R are illustrated through the detailed description of two examples. A first example concerns a population-based analysis of simulated allelic markers for two hybridizing (sub-)species (Durand et al. 2009). The second example concerns single nucleotide polymorphism data for 1,096 European accessions of the plant species *A. thaliana* (52k SNPs from chromosome 1, Horton et al. 2012).

The next paragraphs will guide R users through each step of a population structure analysis, making the operations easily reproducible within their computing environment (R or Rstudio). The inference algorithms used by R are based on a fast version of STRUCTURE available from the R package LEA (Frichot and François 2015). The algorithms for estimating population structure implemented in LEA differ from those of STRUCTURE dramatically, but the estimates of ancestry coefficients are similar to STRUCTURE for out-crossing species, and the estimates can be more accurate than those of STRUCTURE in the presence of inbreeding (Frichot et al. 2014). Choosing the number of clusters is based on cross-validation and on an information theoretic measure, the cross-entropy criterion (Alexander et al. 2011, Frichot et al. 2014). Several R packages for geographic analysis and spatial statistics enable showing admixture bar-plots and displaying pie-charts on geographic maps.

2 Using R to run population structure analyses

2.1 Preliminaries: Installing R packages

Assuming that R version >3.2.1 is installed on the computer system (https://cran.r-project.org/), loading a few additional packages and codes is necessary before starting any population structure analysis. From the R command line, the following instructions install the fields package, which contains tools for spatial data and spatial statistics, RColorBrewer, mapplots and the LEA package for population genetic analyses.

```
install.packages(c("fields","RColorBrewer","mapplots"))
source("http://bioconductor.org/biocLite.R")
biocLite("LEA")
```

If the biocLite command fails (because the version of R is too recent), the installation of LEA can be done from the source codes available from the Bioconductor repository https://www.bioconductor.org/. Download the .tar.gz file to a local directory and type the following R command:

```
install.packages("LEA_1.4.0_tar.gz", repos = NULL, type ="source")
```

The next commands load additional functions to import input files from the STRUCTURE format, and to display nice geographic representations of ancestry coefficients with maps.

```
source("http://membres-timc.imag.fr/Olivier.Francois/Conversion.R")
source("http://membres-timc.imag.fr/Olivier.Francois/POPSutilities.R")
```

2.2 Importing input files

We developed the struct2geno function which imports files from the STRUCTURE or TESS 2.3 format to the formats ".geno" and ".lfmm" used by LEA. The struct2geno function takes any input.file in the STRUCTURE or TESS formats. Among its options, FORMAT = 2 means that markers are encoded using two rows of data for each individual, FORMAT = 1 must be used when markers are encoded using one row of data for each individual. Extra columns include all the columns that do not contain genotypic information (flag TESS = FALSE) or genotypic and geographic information (flag TESS = TRUE). The extra.row and extra.col variables are integers indicating the number of extra rows and columns. The path to the output file is "./genotype.geno".

```
struct2geno(file = input.file, TESS = FALSE, diploid = TRUE, FORMAT = 2,
                extra.row = 0, extra.col = 0, output = "./genotype.geno")
```

2.3 Example 1: Allelic markers

In this section, we analyze the results of spatially explicit coalescent simulations generating genetic data for populations after secondary contact in Europe (Durand et al. 2009). After an initial phase of divergence, a species started to colonize Europe from two distant southern refugia, one in the Iberian peninsula and the other one in Turkey. Secondary contact occurred in Central Europe, in an area close to Germany. The data consists of 60 population samples of 10 diploid individuals that were genotyped at 100 multi-allelic markers. The data can be downloaded and converted as follows.

```
input.file =  "http://membres-timc.imag.fr/Olivier.Francois/secondary_contact.str"
struct2geno(file = input.file, TESS = TRUE, diploid = TRUE, FORMAT = 2,
                    extra.row = 0, extra.col = 0, output = "secondary_contact.geno")
```

Because the data also contained the geographic information for samples, we used the TESS = TRUE flag. Note that the conversion script exports the geographic coordinates (longitude, latitude) in a .coord file. For warming up, let us run a population structure analysis that assumes $K = 3$ clusters. This can be done by using the snmf function of the LEA package.

```
library(LEA)
obj.snmf = snmf("secondary_contact.geno", K = 3, alpha = 100, project = "new")
qmatrix = Q(obj.snmf, K = 3)
```

The algorithm converges very quickly. At the end of the run, the qmatrix object contains the matrix of ancestry coefficients for each individual and for $K = 3$ clusters. The Q-matrix has 600 rows and 3 columns, and it is traditionally displayed using a barplot representation. For this representation, we just use the barplot function of R (Figure 1).

```
barplot(t(qmatrix), col = c("orange","violet","lightgreen"), border = NA, space = 0,
                xlab = "Individuals", ylab = "Admixture coefficients")
```

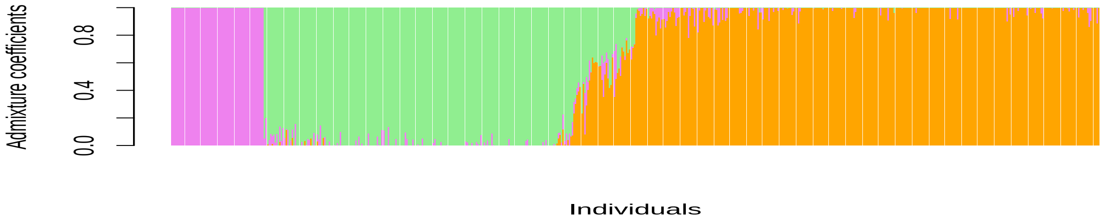

Figure 1. *Secondary contact simulation. Barplot of ancestry coefficients for 600 individuals (allelic data).*

Now we would like to display population estimates of admixture by overlaying pie-charts on a geographic map of Europe. To this goal, we need to read the geographic coordinates of samples and to create sample identifiers. There are 60 population samples with 10 individuals in each population. This can be done as follows.

```
coord = read.table("coordinates.coord")
pop = rep(1:60, each = 10)
```

Population estimates of admixture coefficients can be obtained by taking the mean values for each of the 60 population samples.

```
K = 3
Npop = length(unique(pop))
qpop = matrix(NA, ncol = K, nrow = Npop)
coord.pop = matrix(NA, ncol = 2, nrow =  Npop)
for (i in unique(pop)){
    qpop[i,] = apply(qmatrix[pop == i,], 2, mean)
    coord.pop[i,] = apply(coord[pop == i,], 2, mean)}
```

In this script, the qpop object contains the admixture coefficients for each population sample. The dimension of the matrix is now 60 × 3. A geographic mapping of population admixture proportions can be achieved by using the add.pie function of the mapplots package.

```
library(mapplots)
plot(coord, xlab = "Longitude", ylab = "Latitude", type = "n")
map(add = T, col = "grey90", fill = TRUE)

for (i in 1:Npop){
   add.pie(z = qpop[i,], x = coord.pop[i,1], y = coord.pop[i,2], labels = "",
   col = c("orange","violet","lightgreen"))}
```

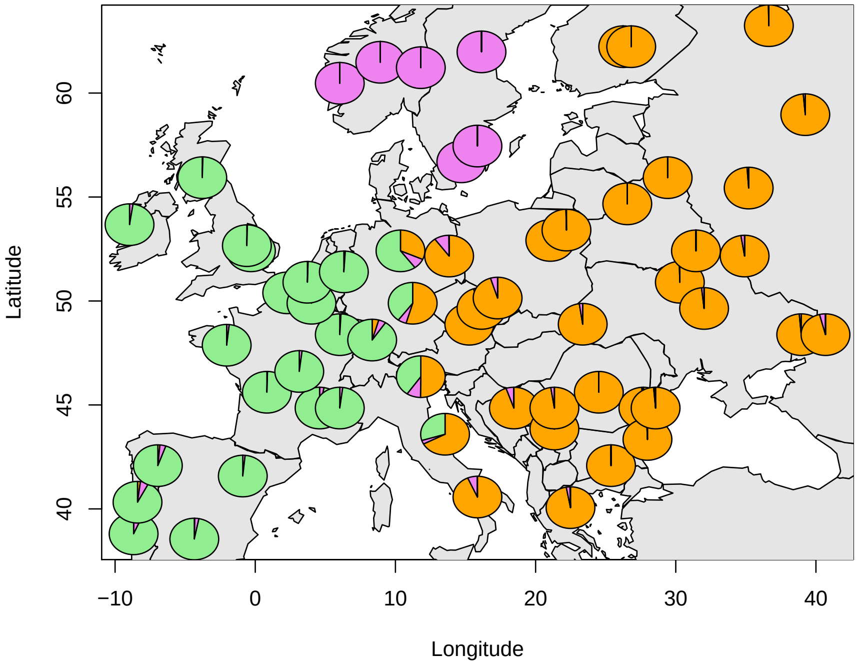

Figure 2. *Map of population admixture estimates using $K = 3$ clusters. Populations of mixed ancestry are identified in Central Europe (contact zone).*

Note that reproducing these commands with your own data will require that you load an addition vector of population labels (one integer for each individual).

```
pop = scan("mypop.txt")
```

2.4 Choosing the number of clusters

In LEA, choosing the number of clusters is based on the cross-entropy criterion. This criterion is also used by the program ADMIXTURE (Alexander et al. 2011). The cross-entropy criterion is based on the prediction of a fraction of masked genotypes (matrix completion), and on the cross-validation approach. Smaller values of the cross-entropy criterion usually mean better runs. We perform runs for 8 values of K, and choose the value of K for which the cross-entropy curve exhibits a plateau ($K = 3$, Figure 3).

```
obj.snmf = snmf("secondary_contact.geno", K = 1:8,  ploidy = 2, entropy = T,
                alpha = 100, project = "new")
plot(obj.snmf, col = "blue4", cex = 1.4, pch = 19)
```

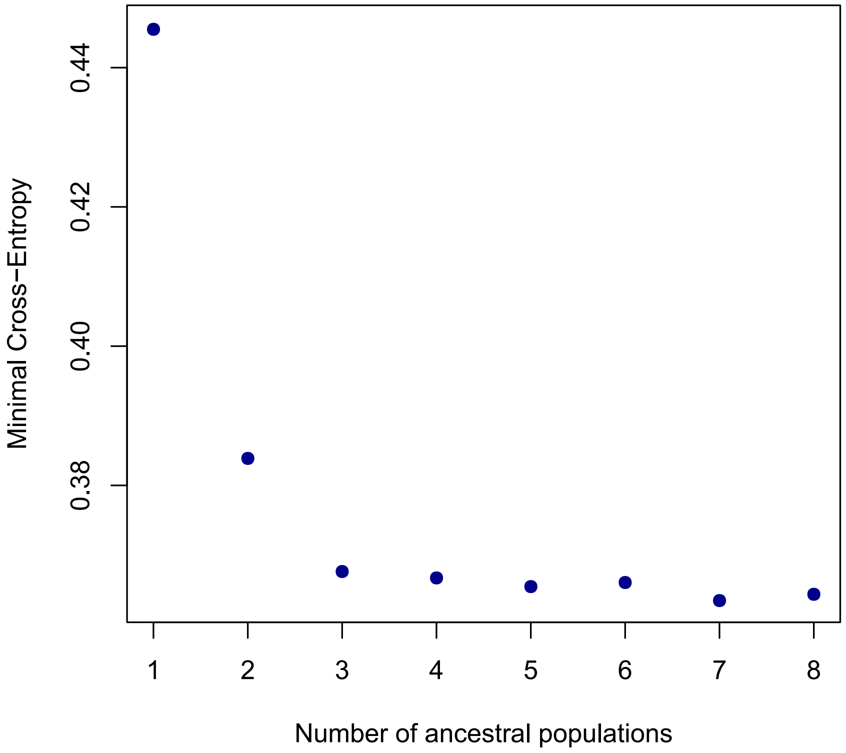

Figure 3. *Cross-entropy plot for the number of cluster* $K = 1$-8*. The retained value of* K *is* $K = 3$.

2.5 Example 2: SNP data

In the second part of this tutorial, we consider SNP data from European ecotypes of the plant species *Arabidopsis thaliana*. The data were extracted from a larger data set published by Horton et al. (2012). In this section, we survey population structure at chromosome 1 for a sample of 1,096 inbred ecotypes (52,001 SNPs). The size of the data set is around 54.4 MB, and we do not want to load those data into the `R` memory. We save the data on the disk, and we also load individual coordinates for each accession.

```
url = "http://membres-timc.imag.fr/Olivier.Francois/Arabidopsis/A_thaliana_chr1.geno"
download.file(url = url, destfile = "./A_thaliana_chr1.geno")
url =  "http://membres-timc.imag.fr/Olivier.Francois/Arabidopsis/at_coord.coord"
download.file(url = url, destfile = "./at_coord.coord")
```

Warning: Breaking the `url` character string causes a program failure. Again, we evaluate population structure using the `snmf` function for $K = 1$-10. Note that we consider the species as a haploid for these data.

```
obj.at = snmf("./A_thaliana_chr1.geno", K = 1:10, ploidy = 1, entropy = T,
                CPU = 1, project = "new")
plot(obj.at, col = "blue4", cex = 1.4, pch = 19)
```

Performing ten runs takes a few minutes (so, it is time for a cup of tea). For the impatient user, runtime could be significantly sped up by using more CPUs of the multi-threated program, for example `CPU = 8`. Here STRUCTURE would take several hours to complete the runs.

Looking at the results, the cross-entropy criterion does not exhibit a minimum value, or a clear plateau (Figure 4). Those results indicate that there are 3 major clusters in Europe, but the program detects much finer population structure than 3 clusters. The population structure reflects a potentially important role for isolation-by-distance processes in having shaped allelic frequencies during the history of the species.

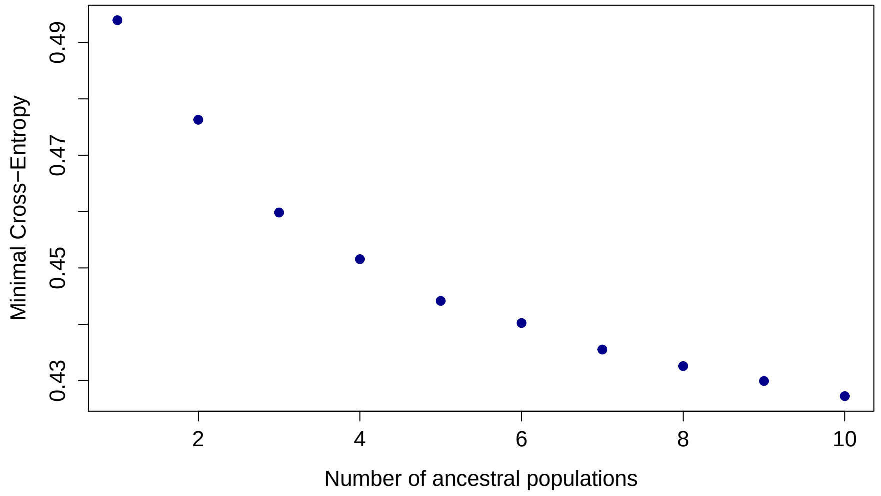

Figure 4. *Cross-entropy plot for the A. thaliana data when the number of cluster ranges between* $K = 1$-10.

Next, we want to visualize the matrix of ancestry coefficients for $K = 5$ clusters. It is obtained as follows.

```
qmatrix = Q(obj.at, K =5)
```

Only individual data but no population samples are available. Thus we cannot use pie-chart mapping as for our first example. Instead, we compute spatial estimates of admixture coefficients, and we represent the spatial predictions on a geographic map. To this aim, a raster grid representing Europe is required. The raster file can be downloaded from a GIS application or from the web. Note that the `RasterMaps` directory contains raster files for all major regions of the world (e.g. `Asia.asc`, `South_America.asc`, `Africa.asc`, etc).

```
asc.raster="http://membres-timc.imag.fr/Olivier.Francois/RasterMaps/Europe.asc"
grid=createGridFromAsciiRaster(asc.raster)
constraints=getConstraintsFromAsciiRaster(asc.raster, cell_value_min=0)
coord.at = read.table("at_coord.coord")
```

Next, we use the maps function from the POPSutilities.R suite of functions (Jay et al. 2012) to perform the spatial interpolation of ancestry coefficients. The output is a nice colored map of admixture coefficients (Figure 5). Using the max option, only the cluster with the maximal local contribution to ancestry is represented at each geographic point of the map.

```
maps(matrix = qmatrix, coord.at, grid, constraints, method = "max",
        main = "Ancestry coefficients", xlab = "Longitude", ylab = "Latitude", cex = .5)
map(add = T, interior = F)
```

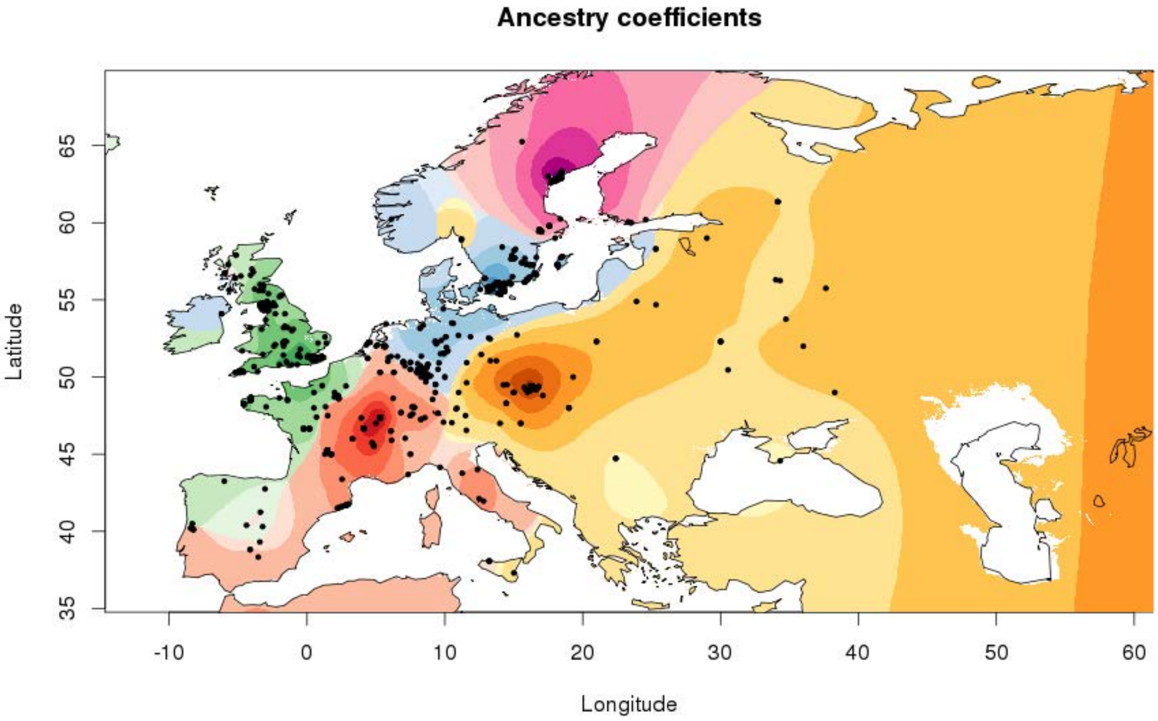

Figure 5. *Interpolated values of admixture coefficients for European A. thaliana ecotypes ($K = 5$).*

Note that the maps function uses 6 colors. To display more than 6 clusters or to change the colors, modify the lColorGradients object (a list object).

3 Additional information

The raster map Europe.asc was downloaded from the following NOAA web site

```
http://maps.ngdc.noaa.gov/viewers/wcs-client/
```

Additional low resolution continental maps are available from the same url by replacing the character string Europe.asc

```
asc.raster="http://membres-timc.imag.fr/Olivier.Francois/RasterMaps/Europe.asc"
```

by one of the following strings (their names speak by themselves)

```
Africa.asc
Asia.asc
Central_Asia.asc
South_America.asc
North_America.asc
Central_America.asc
Australia.asc
Italia.asc
```

For generating personalized maps, go to the NOAA web site, zoom their world map and grab the area of interest by using the xv button on the left of the screen. Save the result as ArcGIS ASCII Grid and eventually modify the header of the ascii file, by adding a sixth row with `NODATA_VALUE -9999`. The resulting header should look like this:

```
NCOLS        1093
NROWS        707
XLLCORNER    61.691666666667
YLLCORNER    2.458333333333
CELLSIZE     0.0833333333334
NODATA_VALUE -9999
```

If you need to change the colors or to add new colors to the maps, use the color names from `RColorBrewer`, and modify the `lColorGradients` object from the R command line. For example, adding a 7th color (say, a purple cluster) could be done as follows:

```
lColorGradients = list(
        c("gray95",brewer.pal(9,"Reds")),
        c("gray95",brewer.pal(9,"Greens")),
        c("gray95",brewer.pal(9,"Blues")),
        c("gray95",brewer.pal(9,"YlOrBr")),
        c("gray95",brewer.pal(9,"RdPu")),
        c("gray95",brewer.pal(9,"Purples")),
        c("gray95",brewer.pal(9,"Greys"))
)
```

Use `display.brewer.all()` to display all color gradients provided by `RColorBrewer`, and play with your own list of colors. This can also be done by using the `colorRampPalette` function of `R` (use 10 values).

4 Extensions

The `R` package `tess3r` implements a version of TESS 2.3 based on accelerating principles similar to those used by LEA (Caye et al. 2016). The commands using the `snmf` function from the LEA package can be replaced by the `tess` function from the `tess3r` package. The `tess` command will run spatially explicit analyses with the objective of ending with better admixture estimates than `snmf` by modelling continuous geographic variation through space. Our next release of `tess3r` will facilitate the use of raster maps, so follow our updates at `@oliviefr`.

References

Alexander DH and Lange K (2011) Enhancements to the ADMIXTURE algorithm for individual ancestry estimation. BMC Bioinformatics 12:246.

Caye K, Deist TM, Martins H, Michel O, François O (2016) TESS3: fast inference of spatial population structure and genome scans for selection. Molecular ecology resources. Molecular Ecology Resources 16 (2), 540-548.

Chen C, Durand E, Forbes F, François O (2007) Bayesian clustering algorithms ascertaining spatial population structure: a new computer program and a comparison study. Molecular Ecology Notes 7: 747-756.

Durand E, Jay F, Gaggiotti OE, François O (2009) Spatial inference of admixture proportions and secondary contact zones. Molecular Biology and Evolution 26:1963-1973.

Frichot E, François O (2015) LEA: an R package for landscape and ecological association studies. Methods in Ecology and Evolution 6:925-929.

Frichot E, Mathieu F, Trouillon T, Bouchard G and François O (2014) Fast and efficient estimation of individual ancestry coefficients. Genetics 196:973-983.

Horton MW, Hancock AM, Huang YS, Toomajian C, Atwell S, Auton A, *et al.* (2012) Genome-wide patterns of genetic variation in worldwide *Arabidopsis thaliana* accessions from the RegMap panel. Nature genetics 44(2): 212-216.

Jay F, Manel S, Alvarez N, Durand EY, Thuiller W et al. (2012) Forecasting changes in population genetic structure of alpine plants in response to global warming. Molecular Ecology 21(10): 2354-2368.

Pritchard JK, Stephens M, Donnelly P (2000) Inference of population structure using multilocus genotype data. Genetics 155:945-959.

R Core Team (2016) R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria.