Documentation for WOMBSOFT

A R package for wombling analyses.

A. Crida and S. Manel

Janvier 2007

TABLE OF CONTENTS :

WOMBSOFT is a R package that analyses individually geo-referenced multilocus genotypes for the inferences of genetic boundaries between populations. It is based on the wombling approach that does not require genetic assumptions. The wombling method estimates the systemic function by looking for the local variation in allele frequencies. The method applies to codominant or dominant markers, as well as contingency data and allows for missing data. Wombsoft can be downloaded from http://www-leca.ujf-grenoble.fr/logiciels.htm. No background of R knowledge is needed to use WOMBSOFT.

The software R can be downloaded from http://www.r-project.org/.

Installing package WOMBSOFT

Windows

Once R is open, use the menu “Packages” and chose the command “install packages from local zip files”. Then select the file wombsoft_1.0.zip from the folder where it has been downloaded. Then you just need to load WOMBSOFT in R using the following command:

```
> library (wombsoft)
```

You are now ready to use the different functions, and you can change directory to select the folder containing your data files.

Linux and Mac

You first need to install the package from source using the following command in a Linux terminal:

```
R CMD INSTALL –use-zip-data /path/wombsoft_1.0.tar.gz
```

Then in R, you just need to load WOMBSOFT using the following command:

```
> library (wombsoft)
```

You are now ready to use the different functions, and you can change directory to select the folder containing your data files.

Wombling Functions

Each of the following functions has been implemented in three versions (Dominant, Codominant, Contingency), one for each type of data. Dominant data includes for example AFLP, co-dominant can be microsatellites, and the last type is a contingency table. All missing values must be coded as -1.

DataDominant, DataCodominant, DataContingency:

These functions read a text file with coordinates and markers in columns and individuals in rows. To be exact, for dominant or codominant data, each row contains the name of the individual, the two coordinates (either abscissa and ordinates, or longitude and latitude), and the genetic data (one column per allele) (see example page 6). Genetic data can be multilocus genotypes (e.g. microsatellites or RAPD/AFLP). The different alleles of multilocus genotype must be coded by numerical values. For diploid individuals, each locus is described by two consecutives columns. In the case of AFLP the recessive allele is coded by 0 and the dominant one by 1. Where a contingency table is used, each row corresponds to a sampled point, with the name of the point, its coordinates, and the number of individuals for each modality of each variable.

It creates a matrix which only contains the coordinates and converts them into kilometres if they were given in degrees, using the Conversion function (described below) (and writes them in a text file called coord_indiv.txt), and another matrix containing the data. It also computes the grid for which we can choose the resolution on each axis, and the convex hull.

Ex: data<-DataCodominant(“test.txt”, conversion=0, nb_x=50, nb_y=50, ”coord_indiv.txt”)

MirrorDominant, MirrorCodominant, MirrorContingency:

These functions may be used to reduce border effects due to a low density of individuals close to the border. They add some points just outside the convex hull, similar to the points just inside the convex hull.

Ex: data<-MirrorCodominant(data,m=1.2) where 1.2 is the distance from the border where we add individuals.

The value is given in the same format as the value of the Data functions, so that they can both be used in a similar way in further analyses.

WomblingDominant, WomblingCodominant, WomblingContingency:

These functions compute the wombling method (i.e. systemic function). They can be applied to the values of the functions Data or Mirror. The resulting matrix of the systemic function is written in the file "systemic.txt", and the directions of the gradient in "direction.txt". It also produces a coloured map, on which each sampled point is a circle, the background is the systemic function in field colours (green corresponds to low values of the systemic function, and light pink to high values of the systemic function, i.e. zones where the population is heterogeneous). The lines show the direction of the gradient.

Ex: *WomblingCodominant(data,h=1.4,"systemic.txt","direction.txt")*

The binomial test is carried out using two functions in succession:

1) CandidateBoundariesDominant, CandidateBoundariesCodominant, CandidateBoundariesContingency:

These functions count for each point of the grid, the number of candidate boundary elements at the given percentile pB at each point of the grid, and computes the overall gradient.

Ex: *cbe<-CandidateBoundariesCodominant(data,h=1.4,pB=0.3)*

2) BinomialTestDominant, BinomialTestCodominant, BinomialTestContingency:

These functions take the value of the CandidateBoundaries functions as input, and test to see if the number of candidate boundary elements at each point of the grid is significant with regards to the binomial distribution, and the pvalue chosen by the user.

It produces a map, where boundaries are shown in light grey and homogenous zones in green. The lines are still the directions of the gradient. The matrices of the boundaries and the directions are written in the files “boundaries.txt” and “dir_boundaries.txt”.

Ex: *BinomialTestCodominant(data,cbe,pvalue=0.05,”boundaries.txt”,”dir_boundaries.txt”)*

Other functions also available:

These functions are known as other WOMBSOFT functions and the user does not need to use them. But as there are available and can be useful for other applications, we will describe them below.

Conversion:

This function converts degree coordinates, namely (θ, φ), into kilometres. It uses the following formula: $x=R(\theta-\theta_0)\sin(\varphi_0)$, $y=R(\varphi-\varphi_0)$, where R=6378km is the earth’s radius, and θ_0, φ_0 are the mean longitude and latitude of the set of points in radians. This formula is convenient if the sample area is not too wide.

Ex: conv<-Conversion(coordinates), where coordinates is a matrix of 2 columns, the first being the set of longitudes and the second latitudes, all given in decimal degrees.

ConvexHull, InConvexHull:

ConvexHull computes the convex hull of a set of points, and returns its vertices.

Ex: cv<-ConvexHull(coordinates), where coordinates is a matrix of 2 columns containing abscissa and the ordinates of the points.

The value is a matrix with the coordinates of the vertices of the convex hull.

InConvexHull tests if a given point is in a given convex hull.

Ex: cvi<-InConvexHull(convex,x,y), where convex is the set of vertices of the convex hull (result of the previous function) and (x,y) the coordinates of the point.

Input File Format

It is a text file.

Example for microsatellite: the first two lines are as follows (5 microsatellites for diploid individuals) :

Indiv1	-0.252109	10.7890	6	18	8	4	7	6	13	1	18	18
Indiv2	0.522792	4.32660	1	-1	16	12	7	7	9	9	18	20

Example for AFLP data (7 loci) :

Indiv1	11.2956	47.6149	0	0	0	1	1	0	1
Indiv2	11.2957	47.615 0	1	0	0	0	0	1	1