Package 'DetSel'

February 19, 2015

Version 1.0.2

Title A computer program to detect markers responding to selection

Author Renaud Vitalis <vitalis@supagro.inra.fr>.

Maintainer Renaud Vitalis <vitalis@supagro.inra.fr>

Depends R (>= 2.15), ash

Description In the new era of population genomics, surveys of genetic polymorphism (``genome scans'') offer the opportunity to distinguish locus-specific from genome wide effects at many loci. Identifying presumably neutral regions of the genome that are assumed to be influenced by genome-wide effects only, and excluding presumably selected regions, is therefore critical to infer population demography and phylogenetic history reliably. Conversely, detecting locus-specific effects may help identify those genes that have been, or still are, targeted by natural selection. The software package DetSel has been developed to identify markers that show deviation from neutral expectation in pairwise comparisons of diverging populations.

License GPL (>= 2)

NeedsCompilation yes

Repository CRAN

Date/Publication 2013-04-24 09:38:17

R topics documented:

`compute.p.values` *Compute Empirical p-values*

Description

This command compute empirical p-values.

Usage

```
compute.p.values(x.range,y.range,n.bins,m)
```

Arguments

`x.range`	the range of values in the x-axis, respectively, which takes the default values x.range = c(-1,1)
`y.range`	the range of values in the y-axis, respectively, which takes the default values y.range = c(-1,1)
`n.bins`	the size of the 2-dimensional array of n x n square cells used to bin the F_1 and F_2 estimates, which takes the default value n.bins = c(100,100)
`m`	the smoothing parameters of the ASH algorithm, which takes the default value m = c(2,2)

Details

compute.p.values(x.range,y.range,n.bins,m) produces an output file, named 'P-values_i_j.dat', with the *P*-value associated with each observation. To that end, the cumulative distribution function (CDF) is evaluated empirically from the joint distribution of all the pairwise observations (F_1,F_2) within the simulated dataset. Then, the empirical *P*-value for a given marker locus i is calculated as one minus the CDF evaluated at locus i. For multi-allelic markers, the joint distribution of all the pairwise observations (F_1,F_2) within the simulated dataset is computed from a 2-dimensional array, where the (F_1,F_2) pairs are binned, and then smoothed using the Average Shifted Histogram (ASH) algorithm (Scott 1992) as implemented in the "ash" R package. Because the distribution of (F_1,F_2) estimates for bi-allelic markers is discontinuous with many ties, the CDF is computed instead by enumerating all (F_1,F_2) pairs in the simulated data.

Value

The output files are saved in the working directory.

References

Scott, D. W. (1992) Multivariate density estimation: theory, practice, and visualization, John Wiley, New York.

Examples

```
## This is to generate an example file in the working directory.
make.example.files()

## This will read an input file named 'data.dat' that contains co-dominant markers,
## and a maximum allele frequency of 0.99 will be applied (i.e., by removing
## marker loci in the observed and simulated datasets that have an allele with
## frequency larger than 0.99).
read.data(infile = 'data.dat',dominance = FALSE,maf = 0.99)

## The following command line executes the simulations:
run.detsel(example = TRUE)

## This compute empirical \emph{P}-values, assuming a range of values from -1 to 1
## in both dimensions, a grid of 50 x 50 bins, and a smoothing parameter m = 3
## in both dimensions.
compute.p.values(x.range = c(-1,1),y.range = c(-1,1),n.bins = c(50,50),m = c(3,3))
```

DetSel *A R-package to Detect Marker Loci Responding to Selection*

Description

In the new era of population genomics, surveys of genetic polymorphism ('genome scans') offer the opportunity to distinguish locus-specific from genome wide effects at many loci. Identifying presumably neutral regions of the genome that are assumed to be influenced by genome-wide effects only, and excluding presumably selected regions, is therefore critical to infer population demography and phylogenetic history reliably. Conversely, detecting locus-specific effects may help identify those genes that have been, or still are, targeted by natural selection. The software package DetSel has been developed to identify markers that show deviation from neutral expectation in pairwise comparisons of diverging populations. Recently, two major improvements have been made: the analysis of dominant markers is now supported, and the estimation of empirical *P*-values has been implemented. These features, which are described below, have been incorporated into an R package, which replaces the stand-alone DetSel software package.

Author(s)

Renaud Vitalis <vitalis@supagro.inra.fr>

References

Vitalis, R., Dawson, K., and Boursot, P. (2001) Interpretation of variation across marker loci as evidence of selection, *Genetics* **158**, 1811–1823.

Vitalis, R., Dawson, K., Boursot, P., and Belkhir, K. (2003) DetSel 1.0: a computer program to detect markers responding to selection, *Journal of Heredity* **94**, 429–431.

Vitalis R. (2012) DETSEL: a R-package to detect marker loci responding to selection, *in* Data Production and Analysis in Population Genomics (Pompanon F. and Bonin A., eds), pp. 277–293 *Methods in Molecular Biology, vol. 888*, Humana Press, USA.

DetSel-internal *Internal DetSel Function*

Description

This function is not intended to be called by the user.

draw.detsel.graphs *Plot Graphical Outputs*

Description

This command plots graphical outputs for DetSel analyses.

Usage

```
draw.detsel.graphs(i,j,x.range,y.range,n.bins,m,alpha,pdf,outliers)
```

Arguments

i	population index
j	population index
x.range	the range of values in the x-axis, respectively, which takes the default values x.range = c(-1,1)
y.range	the range of values in the y-axis, respectively, which takes the default values y.range = c(-1,1)
n.bins	the size of the 2-dimensional array of n x n square cells used to bin the F_1 and F_2 estimates, which takes the default value n.bins = c(100,100)
m	the smoothing parameters of the ASH algorithm, which takes the default value m = c(2,2)
alpha	the alpha-level (hence 1 - alpha is the proportion of the distribution within the plotted envelope), which takes the default value alpha = 0.05
pdf	a logical variable, which is TRUE if the user wants graphics to be plotted in a pdf file
outliers	an optional vector that represents a list of candidate outliers, defined by the user

Details

Once the run.detsel and compute.p.values command lines have been executed, the function draw.detsel.graphs can be used to plot graphs with an estimation of the density of F_1 and F_2 estimates, as detailed in the appendix in Vitalis et al (2001). Note that if the arguments i and j are missing, then all the population pairs are plotted. It is noteworthy that our estimation of the density of the F_1 and F_2 estimates might be discontinuous, because of the discrete nature of the data (the allele counts). This is particularly true when the number of alleles upon which the distribution is conditioned is small. The command line draw.detsel.graphs produces as many conditional distributions per population pair as there are different allele numbers in the pooled sample. All the observed data points are plotted in each graph. The outlier loci are plotted with a star symbol. For the latter, the locus number (i.e., its rank in the data file) is provided on the graph. If the user choses not to provide a pre-defined list of outliers, then the outlier represent all the markers for which the empirical *P*-value is below the threshold alpha-level,

Value

The pdf files are created in the current directory.

References

Vitalis, R., Dawson, K., and Boursot, P. (2001) Interpretation of variation across marker loci as evidence of selection, *Genetics* **158**: 1811–1823.

Examples

```
## This is to generate an example file in the working directory.
make.example.files()

## This will read an input file named 'data.dat' that contains co-dominant markers,
## and a maximum allele frequency of 0.99 will be applied (i.e., by removing
## marker loci in the observed and simulated datasets that have an allele with
## frequency larger than 0.99).
read.data(infile = 'data.dat',dominance = FALSE,maf = 0.99)

## The following command line executes the simulations:
run.detsel(example = TRUE)

## This compute empirical P-values, assuming a range of values from -1 to 1
## in both dimensions, a grid of 50 x 50 bins, and a smoothing parameter m = 3
## in both dimensions.
compute.p.values(x.range = c(-1,1),y.range = c(-1,1),n.bins = c(50,50),m = c(3,3))

## This plots (on the screen) the 99% confidence regions corresponding to the
## pair of populations 1 and 2, using a 50 x 50 2-dimensions array.
draw.detsel.graphs(i = 1,j = 2,n.bins = c(50,50),alpha = 0.01,pdf = FALSE)
```

`genepop.to.detsel` *Convert Input File*

Description

This command converts a data file in Genepop format (see Raymond and Rousset 1995; Rousset 2007), which name can be specified using the infile argument into a data file in DetSel format, which name can be specified using the outfile argument.

Usage

```
genepop.to.detsel(infile,outfile = 'data.dat')
```

Arguments

`infile`	An input file in Genepop format.
`outfile`	An output file in DetSel format.

Details

This command is only available for co-dominant data. The output file is a space-delimited ASCII text file. The first line is a 0 / 1 indicator. '0' indicates that the data matrix for each locus is a populations x alleles matrix; '1' indicates that the data matrix for each locus is an alleles x populations matrix. The second line contains the number of populations. The third line contains the number of loci. Then, the data for each locus consists in the number of alleles at that locus, followed by the data matrix at that locus, with each row corresponding to the same allele (if the indicator variable is 1) or to the same population (if the indicator variable is 0). For dominant data, the data consists in the number of genotypes, not the number of alleles. It is important to note that the frequency of the homozygote individuals for the recessive allele appear first in either the rows or columns of the data matrix. In the following example, the data consists in 2 populations and 2 loci, with 5 alleles at the first locus and 8 alleles at the second locus.

0
2
2

5
1 0 4 10 5
0 1 13 0 6

8
3 1 1 0 0 0 1 14
6 0 2 1 2 5 2 2

Value

The output file is saved in the working directory.

References

Raymond, M., and Rousset, F. (1995) Genepop (version 1.2): population genetics software for exact tests and ecumenicism, *Journal of Heredity* **86**: 248–249.

Rousset, F. (2008) genepop'007: a complete re-implementation of the genepop software for Windows and Linux, *Molecular Ecology Notes* **8**: 103–106.

Examples

```
## This is to generate an example file in the working directory.
make.example.files()
## This is to convert the example file in genepop format named 'genepop.dat',
## into a file in DetSel format named 'converted_data.dat'
genepop.to.detsel(infile = 'data.gen',outfile = 'data-converted.dat')
```

---

make.example.files *Generate an Example File in the Working Directory*

Description

This command will copy two example files into the user's working directory. The file "data.dat" is in DetSel format, and the file "data.gen" is in Genepop format (see Raymond and Rousset 1995; Rousset 2007).

Usage

```
make.example.files()
```

Value

The example file is saved in the current directory.

References

Raymond, M., and Rousset, F. (1995) Genepop (version 1.2): population genetics software for exact tests and ecumenicism, *Journal of Heredity* **86**: 248–249.

Rousset, F. (2008) genepop'007: a complete re-implementation of the genepop software for Windows and Linux, *Molecular Ecology Notes* **8**: 103–106.

Examples

```
## This is to generate an example file in the working directory.
make.example.files()
```

`read.data` *Read Data*

Description

Read the data file in DetSel format.

Usage

```
read.data(infile,dominance,maf,a,b)
```

Arguments

`infile`	An input file in DetSel format.
`dominance`	A a logical variable, which is FALSE if co-dominant data are considered (e.g., microsatellite markers, SNPs, etc.), or TRUE, if bi-allelic dominant data are considered (e.g., AFLPs).
`maf`	The maximum allele frequency (the frequency of the most frequent allele over the full sample) to be considered in both the input file and the simulated data.
`a,b`	The parameters for the beta prior distribution, used in Zhivotovsky's (1999) Bayesian method to compute the underlying allele frequencies. The default values are a = b = 0.25, as suggested by Mark A. Beaumont in the DFdist manual, yet the user may alternatively chose to use Zhivotovsky's equation (13) to compute estimates of a and b from the data. Note that neither the parameter a nor the parameter b are not needed if dominance = FALSE.

Details

The input file should be a space- or tab-delimited ASCII text file. The first line is a 0 / 1 indicator. '0' indicates that the data matrix for each locus is a populations x alleles matrix; '1' indicates that the data matrix for each locus is an alleles x populations matrix. The second line contains the number of populations. The third line contains the number of loci. Then, the data for each locus consists in the number of alleles at that locus, followed by the data matrix at that locus, with each row corresponding to the same allele (if the indicator variable is 1) or to the same population (if the indicator variable is 0). For dominant data, the data consists in the number of genotypes, not the number of alleles. It is important to note that the frequency of the homozygote individuals for the recessive allele appear first in either the rows or columns of the data matrix. In the following example, the data consists in 2 populations and 2 loci, with 5 alleles at the first locus and 8 alleles at the second locus.

0
2
2

5
1 0 4 10 5

0 1 13 0 6

8
3 1 1 0 0 0 1 14
6 0 2 1 2 5 2 2

Spaces and blank lines can be included as desired.

For dominant data, it is important to note that the frequency of the homozygote individuals for the recessive allele appears first in either the rows or columns of the data matrix.

The command line `read.data` creates a file named 'infile.dat', a file named 'sample_sizes.dat' and a set of files named 'plot_i_j.dat' where i and j correspond to population numbers, so that each file 'plot_i_j.dat' corresponds to the pairwise analysis of populations i and j. In the file infile.dat, each line corresponds to the pairwise analysis of populations i and j. Each line contains (in that order): the name of the output simulation file, the numbers i and j, the multi-locus estimates of F_1 and F_2, and Weir and Cockerham's (1984) estimate of F_{ST}. The file sample_sizes.dat contains sample sizes information, for internal use only. In the files 'plot_i_j.dat', each line corresponds to one locus observed in the data set. Each line contains (in that order): the locus-specific estimates of F_1 and F_2, Weir and Cockerham's (1984) estimate of F_{ST}, Nei's heterozygosity (H_e), the number of alleles at that locus in the pooled sample, and the rank of the locus in the data set.

Value

The output files are saved in the current directory.

References

Weir, B. S., and Cockerham, C. C. (1984) Estimating F-statistics for the analysis of population structure, *Evolution* **38**: 1358–1370.

Zhivotovsky, L. A. (1999) Estimating population structure in diploids with multilocus dominant DNA markers, *Molecular Ecology* **8**, 907–913

Examples

```
## This is to generate an example file in the working directory.
make.example.files()

## This will read an input file named 'data.dat' that contains co-dominant markers,
## and a maximum allele frequency of 0.99 will be applied (i.e., by removing
## marker loci in the observed and simulated datasets that have an allele with
## frequency larger than 0.99).
read.data(infile = 'data.dat',dominance = FALSE,maf = 0.99)
```

`run.detsel` *Create Simulated Data*

Description

This runs the simulated data, using a coalescent-based algorithm.

Usage

```
run.detsel(example)
```

Arguments

`example`	a logical variable, which is TRUE if the user wants to run a toy example with the example file.

Details

Once the `read.data` command line has been executed, `run.detsel` executes the simulations. The user is first asked to provide the total number of simulations for the entire set of parameter values (default: 500000). For bi-allelic data, I recommend to run no less than 1000000 simulations to estimate correctly the *P*-values for each empirical locus. With less than a million simulations, indeed, simulation tests have shown that the *P*-values may be biased. The user is then asked to provide the average mutation rate, and the mutation model: type '0' for the infinite allele model, where each mutation creates a new allelic state in the population; type '1' for the stepwise mutation model, where each mutation consists in increasing or decreasing by one step, the size of the current allele; and type any integer k (with $k > 1$) for a k-allele model, where each mutation consists in drawing randomly one allele among k possible states, provided it is different from the current state. For example, for SNP data, type '2'. Finally, the user is asked to provide the number of distinct sets of nuisance parameters (the default is a single set of parameters). Because of the uncertainty in the nuisance parameter values, it is recommended to perform simulations using different combinations of values for the ancestral population size, divergence time and bottleneck parameters. Then, the user is asked to provide as many sets of parameters as he/she indicated. Each set comprises four parameters that should be given in the following order: t, N_0, t_0 and N_e. Here, the user must chose parameter values, including the mutation model, that correspond to his/her knowledge of the biological model.

The command line `run.detsel` creates a list of files named 'Pair_i_j_ni_nj.dat', where i and j are the indices of populations pairs, and ni and nj are the sample sizes of populations i and j, respectively. Because some marker loci may have missing data, several 'Pair_i_j_ni_nj.dat' files may be created for a given pair of populations. Simulating the exact sample size for each locus is required to precisely calculate the empirical *P*-values, especially for bi-allelic markers. Note that if negative multi-locus Fi estimates are observed for a pairwise comparison, then the simulations will not be run for that pair. Each line of the 'Pair_i_j_ni_nj.dat' files contains the locus-specific estimates of F_1 and F_2, Weir and Cockerham's estimate of F_{ST} (Weir and Cockerham 1984), Nei's heterozygosity (H_e), and the number of alleles at that locus in the pooled sample. The command line run.detsel() also creates a file named out.dat that contains the estimates of the above statistics averaged over all the simulated data. In the file out.dat, each line corresponds to the pairwise

analysis of populations i and j with sample sizes and ni and nj. Each line contains (in that order): the name of the output simulation file ('Pair_i_j_ni_nj.dat'), the multi-locus estimates of F_1 and F_2, and Weir and Cockerham's estimate of F_{ST} (Weir and Cockerham 1984). An important point to consider is to make sure that for each pairwise comparison, the and estimates averaged over the simulated data (in the file out.dat) closely match to the observed values in the real dataset (in the file infile.dat). If not, this suggests that the simulated datasets do not fit to the observed data, which urges to choose other parameter values for the nuisance parameters.

Value

The output files are saved in the current directory.

References

Weir, B. S., and Cockerham, C. C. (1984) Estimating F-statistics for the analysis of population structure, *Evolution* **38**: 1358–1370.

Examples

```
## This is to generate an example file in the working directory.
make.example.files()

## This will read an input file named 'data.dat' that contains co-dominant markers,
## and a maximum allele frequency of 0.99 will be applied (i.e., by removing
## marker loci in the observed and simulated datasets that have an allele with
## frequency larger than 0.99).
read.data(infile = 'data.dat',dominance = FALSE,maf = 0.99)

## The following command line executes the simulations:
run.detsel(example = TRUE)
```

Index