Package 'LDJump'

February 9, 2018

Type Package

Title Estimating Variable Recombination Rates from Population Genetic Data

Version 0.1.7

Imports adegenet (>= 2.0.1),
ape,
genetics (>= 1.3.8.1),
Biostrings (>= 2.38.4),
stepR (>= 2.0.1),
vcfR (>= 1.5.0),
snow,
data.table

Depends R (>= 2.10),
seqinr (>= 3.1-3),
quantreg

Author Philipp Hermann, Andreas Futschik

Maintainer Philipp Hermann <philipp.hermann@jku.at>

Description This package estimates variable recombination rates from population genetic data. It is a unix based program (with a necessary installation of LDhat), able to estimate the recombination map of sequences in fasta and vcf format. Sequences are divided in short segments of user defined length. The recombination rate is estimated for every segment with a regression model. This set of estimates is fed in a segmentation algorithm (SMUCE) to estimate the breakpoints of the recombination landscape.

License GPL-2

Encoding UTF-8

LazyData TRUE

RoxygenNote 6.0.1

BugReports https://github.com/PhHermann/LDJump/

SystemRequirements Unix Operating System, LDhat (Version 2.2), dos2unix, awk

R topics documented:

`check_continue` *Checks whether there are SNPs in each segment*

Description

This function calculates the number of SNPs per segment. In case that there exist segments with less than 2 SNPs the user is asked for input to continue ("y") or not ("n"). In case that the user wants to continue, the recombination rates for segments without SNPs are estimated via imputation.

Usage

```
check_continue(seqName, segs)
```

Arguments

`seqName`	A character string containing the path and the name of the sequence file in `fasta` of `vcf` format.
`segs`	A (non-negative) integer which reflects the number of segments considered. It is calculated in the program based on the user-defined segment length.

Value

This function returns `TRUE` in case that all segments contain SNPs. It will also return `TRUE` if the user agrees to continue although there exist segments without SNPs. It returns `FALSE` if the user denies to continue due to segments without SNPs.

Author(s)

Philipp Hermann <philipp.hermann@jku.at>, Andreas Futschik

References

Paradis E., Claude J. & Strimmer K. 2004. APE: analyses of phylogenetics and evolution in R language. Bioinformatics 20: 289-290.

See Also

`read.FASTA`, `seg.sites`

Examples

```
##### Do not run these examples                                  #####
##### check_continue(seqName, segs = segs)                       #####
```

fgt_rrate_dpr *Four Gametes Test, R^2 and LD' Calculation*

Description

This helper function calculates three summary statistics of the regression model. Here, the four gametes test, R^2, and LD' are calculated for each pair of sites and returned to its calling function (summary_statistics)

Usage

```
fgt_rrate_dpr(x, y, data1, data2)
```

Arguments

x	Site 1
y	Site 2
data1	Data set to calculate the first two summary statistics
data2	Data set to calculate the four gametes test

Value

A vector is returned containing three values for:

fgt	An indicator value whether the four gametes test indicates a recombination event
R^2	for the pair of sites x and y using diseq, genotype
LD'	for the pair of sites x and y using diseq, genotype

Author(s)

Philipp Hermann <philipp.hermann@jku.at>, Andreas Futschik

References

Gregory Warnes, with contributions from Gregor Gorjanc, Friedrich Leisch and Michael Man. (2013). genetics: Population Genetics. R package version 1.3.8.1.

See Also

LDJump, vcfR_to_fasta, summary_statistics, get_smuce, diseq, genotype

Examples

```
##### Do not run these examples                                          #####
##### In summary_statistics.R the function is called as follows          #####
##### helper = mapply(fgt_rrate_dpr, x = indices[,1], y = indices[,2],   #####
#####                MoreArgs = list(data1 = g2dftemp, data2 = samp))    #####
```

get_impute_data *Data management to obtain data used for imputation*

Description

This function obtains the neighbouring values for every segment without SNPs. The function either merges data of two or four neighbouring values. It is only used in the impute function.

Usage

```
get_impute_data(index, data,two=T,segs)
```

Arguments

index	this is a vector containing the integer number of the segments without SNPs.
data	A data vector containing the estimated recombination rates per segment.
two	A logical parameter indicating whether two neighbouring values (if TRUE) or four neighbouring values (if FALSE) should be provided.
segs	A (non-negative) integer which reflects the number of segments considered. It is calculated in the program based on the user-defined segLength.

Value

The function returns a vector containing the corresponding data which can be used for imputation. Note that the ordering is made that values of distance 1 are listed first in case that four neighbouring values are returned.

Author(s)

Philipp Hermann <philipp.hermann@jku.at>, Andreas Futschik

See Also

impute

Examples

```
##### Do not run these examples                                  #####
##### This command shows how it is used in the impute function   #####
##### sapply(index, get.impute.data, data, two = two)            #####
```

get_smuce *Segmentation Algorithm to Estimate Breakpoints in the Recombination Map*

Description

First, the recombination rates per segment are computed based on the regression model (generalized additive models) as well as the bias correction. Consequently, we apply SMUCE (simultaneous multiscale change-point estimator) of Frick (2014) and Futschik et al. (2014) to estimate locations and breakpoints in the recombination map. Under a specific type-I error probability alpha the number of distinct segments with respect to the recombination rate is not overestimated.

Usage

```
get_smuce(help, segs, alpha, ll, quant = 0.35, rescale, constant)
```

Arguments

help	a matrix containing a set of summary statistics is calculated in the function summary_statistics. These values are used in the regression model to calculate the (constant) recombination rates.
segs	A (non-negative) integer which reflects the number of segments considered. It is calculated in the program based on the user-defined segLength.
alpha	A value from the interval (0,1) for the type-I error probability used in the segmentation algorithm. We recommend to use 0.05. We enabled to estimate the recombination map efficiently (without recalculating all summary statistics) under several type-I errors when LDJump is applied with a vector of type-I error probabilities.
ll	A (non-negative) integer which reflects the total sequence length of the sequences under study.
quant	A value between 0.1 and 0.5 with 0.05 distances in between which reflects the quantile used in the quantile regression. We recommend to use the 0.35 quantile.
rescale	an optional logical value: if TRUE, it rescales the sequence length of the output of SMUCE to a range from 0 to 1.
constant	an optional logical value: by default FALSE estimating variable recombination rates. If TRUE, the constant recombination rate for the full sequence is estimated.

Value

seq.full.cor	The final estimate of the recombination map. Depiction with plot-function of stepR package.
pr.full.cor	A vector of (constant) estimates of the recombination rate per segment.

Author(s)

Philipp Hermann <philipp.hermann@jku.at>, Andreas Futschik

References

Frick, K., Munk, A., and Sieling, H. (2014). Multiscale change-point inference. Journal of the Royal Statistical Society: Series B, 76(3), 495–580.

Futschik, A., Hotz, T., Munk, A., and Sieling, H. (2014). Multiscale DNA partitioning: Statistical evidence for segments. Bioinformatics, 30(16), 2255–2262.

See Also

LDJump, vcfR_to_fasta, fgt_rrate_dpr, summary_statistics, stepFit, rq, gam

Examples

```
##### Do not run these examples                          #####
##### In LDJump.R the function is called as follows      #####
##### get_smuce(help, segs, alpha,ll,list.quantile.regs) #####
```

`impute` *Imputation of estimated recombination rates for segments without SNPs*

Description

This function recursively imputes the recombination rate for missing segments. First it imputes the mean of the two neighbouring segments. In case that one of these segments is also missing, it then imputes the weighted mean of the four neighbouring segments putting higher weights to the closer segments. Exceptions were made for e.g. the first and last segment in the sequence. It also imputes those positions first, where more information is already available.

Usage

```
impute(data, index, two, segs)
```

Arguments

`data`	A data vector containing the estimated recombination rates per segment.
`index`	this is a vector containing the integer number of the segments without SNPs.
`two`	A logical parameter indicating whether two neighbouring values (if `TRUE`) or four neighbouring values (if `FALSE`) should be provided.
`segs`	A (non-negative) integer which reflects the number of segments considered. It is calculated in the program based on the user-defined `segLength`.

Details

The function calls itself after every imputation step trying to impute based on two neighbouring segments.

Value

`data`	This vector contains the estimated recombination rates including the imputed values for the segments without SNPs.

Author(s)

Philipp Hermann <philipp.hermann@jku.at>, Andreas Futschik

See Also

get_impute_data

Examples

```
##### Do not run these examples                                    #####
##### This command shows how it is used in the get_smuce function #####
##### pr.cor.nat = impute(pr.cor.nat, ind, two = T)               #####
```

LDJump *Estimate Variable Recomination Rates from Population Genetic Data*

Description

This function estimates variable recombination rates from population genetic data. Therefore, a segmentation algorithm with specific segment lengths (segLength) and type-I error probability (alpha, α) is applied. The returned object can be plotted with the plot-function of the package stepR.

Usage

```
LDJump(seqName, alpha = 0.05, segLength = 1000, pathLDhat = "",
       format = "fasta", refName = NULL, start = NULL, thth = 0.005,
       constant = F, rescale = F, status = T, polyThres = 0, cores = 1)
```

Arguments

seqName	A character string containing the path and the name of the sequence file in fasta of vcf format.
alpha	A value from the interval (0,1) for the type-I error probability α used in the segmentation algorithm. We recommend to use 0.05. We enabled to estimate the recombination map efficiently (without recalculating all summary statistics) under several type-I errors when LDJump is applied with a vector of type-I error probabilities.
segLength	An integer value for the length of the segments, provided by the user. The default value of 1000 is our recommended value (1kb). The number of resulting segments, based on the sequence length is calculated within the funtion.
pathLDhat	A character string containing the path to LDhat. This path and the installation of LDhat is necessary for the computation of the package.
format	A character string which can be either fasta or vcf. If fasta is used, the package will proceed with the computation of the recombination map. If vcf, the package will convert the data in vcf format to fasta format with the function vcfR_to_fasta and then proceed as in case fasta.
refName	An (optional) path to the reference sequence for the region of interest downloaded from e.g. http://phase3browser.1000genomes.org/index.html. Only to be used in case that format == "vcf".

start	An (optional) integer value which reflects the starting position of the sequences in bp. Only to be used in case that format == "vcf".
thth	A numeric value for θ used in the lookup tables of LDhat.
constant	an optional logical value: by default FALSE estimating variable recombination rates. If TRUE, the constant recombination rate for the full sequence is estimated.
rescale	an optional logical value: by default FALSE the sequence length is not rescaled to 0 and 1. If TRUE this rescaling is performed.
status	an optional logical value: by default TRUE such that the current processing status of the segments is printed.
polyThres	a numeric value between 0 and 1. Used in data manipulation function DNAbin2genind: the minimum frequency of a minor allele for a locus to be considered as polymorphic (default to 0).
cores	a positive integer value which is by default 1. This integer reflects the number of cores to be used. Hence, when setting to an integer larger then one the same number of cores are used to compute the recombination map. For small sequences, we do not recommend to set the number of cores larger than 2 in this moment.

Value

The following list is returned in the case of estimating variable recombination rates (constant == FALSE).

seq.full.cor	The final estimate of the recombination map. Depiction with plot-function of stepR package.
pr.full.cor	The (constant) estimates of the recombination rate per segment.
help	A helper matrix containing the summary statistics per segment used in the regression model.
alpha	The type-I error probability α.
nn	The number of individuals (more precisely sequences) for which the recombination map was estimated.
ll	Total sequence length
segs	The number of segments by which the sequence is divided. Resulting from the user-defined segment length (segLength).

For constant recombination rate estimation across the whole sequences (constant == TRUE), we provide the same list except for seq.full.cor.

Note

This function only works with unix and having LDhat (Auton and McVean (2007)) installed. Please properly check all paths to LDhat as well as the sequence files. The required lookup tables used by LDhat should be located in the path "pathToLDhat/LDhat-master/lk_files". Lookup tables are contained in LDhat, but we also provide several lookup tables here.

Author(s)

Philipp Hermann <philipp.hermann@jku.at>, Andreas Futschik

References

Auton, A. and McVean, G. (2007). Recombination rate estimation in the presence of hotspots. Genome Research, 17(8), 1219–1227.

Frick, K., Munk, A., and Sieling, H. (2014). Multiscale change-point inference. Journal of the Royal Statistical Society: Series B, 76(3), 495–580.

Futschik, A., Hotz, T., Munk, A., and Sieling, H. (2014). Multiscale DNA partitioning: Statistical evidence for segments. Bioinformatics, 30(16), 2255–2262.

Jombart T. and Ahmed I. (2011) adegenet 1.3-1: new tools for the analysis of genome-wide SNP data. Bioinformatics. doi: 10.1093/bioinformatics/btr521

Knaus BJ and Grünwald NJ (2017). VCFR: a package to manipulate and visualize variant call format data in R. Molecular Ecology Resources, 17(1), pp. 44-53. ISSN 757, <URL:http://dx.doi.org/10.1111/1755-0998.12549>.

McVean, G. A. T., Myers, S. R., Hunt, S., Deloukas, P., Bentley, D. R., and Donnelly, P. (2004). The fine-scale structure of recombination rate variation in the human genome. Science, 304(5670), 581–584.

Paradis E., Claude J. & Strimmer K. 2004. APE: analyses of phylogenetics and evolution in R language. Bioinformatics 20: 289-290.

The 1000 Genomes Project Consortium (2015). Aglobal reference for human genetic variation. Nature, 526(7571), 68–74.

Wood, S.N. (2011) Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. Journal of the Royal Statistical Society (B) 73(1):3-36

See Also

summary_statistics, vcfR_to_fasta, fgt_rrate_dpr, get_smuce, smuceR, rq, gam, vcfR2DNAbin, diseq, genotype, readDNAStringSet

Examples

```
##### Do not run these examples                                    #####
##### result = LDJump(fileName, alpha = 0.05, segLength = 1000,  #####
#####                 pathLDhat = getwd(), format = "fasta")          #####
##### plot(results)                                                #####
##### ab <- system(paste("locate LDhat-master |  head -n 1"), intern = T)   #####
##### map.tsi.40 = LDJump("S40example.fa", alpha = 0.05, segLength = 1000,  #####
#####                 pathLDhat = ab, format = "fasta")                      #####
##### plot(map.tsi.40)                                                    #####
```

`list.quantile.regs` *Quantile Regressions for Bias Correction*

Description

This data set contains a list of quantile regression models (rq) for bias correction. In total nine regression models are saved in a list form, where we recommend to use the 0.35 for the correction.

Usage

```
data("list.quantile.regs")
```

Format

A list object of length 9, containing quantiles in sequences between 0.1 and 0.5 with 0.05 distances.

Examples

```
data(list.quantile.regs)
##### Do not run these examples                                          #####
##### In get_smuce the function is called as follows                     #####
##### pr1 = predict(mod.full,help); pr1[is.na(pr1)] = -1/gam;            #####
##### ind.q = which(seq(0.1, 0.5, by = 0.05) == quant)                   #####
##### pr.cor = predict(list.quantile.regs[[ind.q]], data.frame(x = pr1)  #####
```

`mod.full` *Regression model to Estimate (constant) Recombination Rates from Population Genetic Summary Statistics*

Description

This data set contains a generalized additive regression model (gam) which estimates the constant recombination rate for a set of segments based on summary statistics.

Usage

```
data("mod.full")
```

Format

A list containing all information on the regression model such as the coefficients, residuals, among others as usually for generalized additive models (gam) saved as an `R object`.

References

Wood, S.N. (2011) Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. Journal of the Royal Statistical Society (B) 73(1):3-36

Examples

```
data(mod.full)
##### Do not run these examples                        #####
##### In get_smuce the function is called as follows   #####
##### pr1 = predict(mod.full,help)                     #####
```

summary_statistics *Summary Statistics per Segment*

Description

This function computes summary statistics for every segment of the sequence. Sequence files are generated within this function which are then used by LDhat and other packages to estimate all necessary parameters.

Usage

```
summary_statistics(x, s, segLength, segs, seqName, nn, thth,
                   cor = 1, pathLDhat, status, polyThres)
```

Arguments

x	An integer control variable for the considered segment of the DNA sequence.
s	An XStringSet object which is read by readDNAStringSet
segLength	An integer value for the length of the segments, provided by the user. The default value of 1000 is our recommended value (1kb). The number of resulting segments, based on the sequence length is calculated within the funtion.
segs	A (non-negative) integer which reflects the number of segments considered. It is calculated in the program based on the user-defined segLength.
seqName	A character string containing the path and the name of the sequence file in fasta of vcf format.
nn	An integer which reflects the number of individuals (more precisely sequences) of the population to be analyzed. In case of diploid samples this is twice the number of individuals.
thth	A numeric value for θ used in the lookup tables of LDhat.
cor	An integer value which reflects the number of cores on which LDhat should be run. We recommend to keep here 1 core.
pathLDhat	A character string containing the path to LDhat. This path and the installation of LDhat is necessary for the computation of the package.
status	an optional logical value: by default TRUE such that the current processing status of the segments is printed.
polyThres	a numeric value between 0 and 1. Used in data manipulation function DNAbin2genind: the minimum frequency of a minor allele for a locus to be considered as polymorphic (default to 0).

Value

This function returns a concatenated vector of two separately used vectors (scalars) of summary statistis as:

stats	A vector of summary statistics. Returned with the value of hahe.
hahe	The haplotype heterozygosity of the considered segment. Returned with stats.

Author(s)

Philipp Hermann <philipp.hermann@jku.at>, Andreas Futschik

References

Auton, A. and McVean, G. (2007). Recombination rate estimation in the presence of hotspots. Genome Research, 17(8), 1219–1227.

Jombart T. and Ahmed I. (2011) adegenet 1.3-1: new tools for the analysis of genome-wide SNP data. Bioinformatics. doi: 10.1093/bioinformatics/btr521

McVean, G. A. T., Myers, S. R., Hunt, S., Deloukas, P., Bentley, D. R., and Donnelly, P. (2004). The fine-scale structure of recombination rate variation in the human genome. Science, 304(5670), 581–584.

Paradis E., Claude J. & Strimmer K. 2004. APE: analyses of phylogenetics and evolution in R language. Bioinformatics 20: 289-290.

See Also

LDJump, vcfR_to_fasta, fgt_rrate_dpr, get_smuce, readDNAStringSet, DNAbin2genind

Examples

```
##### Do not run these examples                                                  #####
##### In LDJump.R the function is called as follows                              #####
##### sapply(1:segs,summary_statistics,s=s,segs=segs,seqName=seqName,nn=nn,ll = ll)  #####
```

`vcfR_to_fasta` *Conversion of vcf to fasta Format*

Description

This function enables to read vcfR files and convert them to necessary fasta files. Therefore, we recommend to provide a reference sequence from e.g. genome browser and the starting position. The default parameters are those of the vcfR package.

Usage

```
vcfR_to_fasta(seqName, refName = NULL, ext.ind = T, cons = F,
              ext.haps = T, start = NULL)
```

Arguments

`seqName`	A character string containing the path and the name of the sequence file in vcf format.
`refName`	An (optional) path to the reference sequence for the region of interest downloaded from e.g. http://phase3browser.1000genomes.org/index.html. Only to be used in case that `format == "vcf"`.
`ext.ind`	See package **vcfR** for details (`vcfR2DNAbin`, `extract.indels`)
`cons`	See package **vcfR** for details (`vcfR2DNAbin`, `consensus`)
`ext.haps`	See package **vcfR** for details (`vcfR2DNAbin`, `extract.haps`)
`start`	An (optional) integer value which reflects the starting position of the sequences in bp. Only to be used in case that `format == "vcf"`.

Value

A print command provides information that the file is converted.

Author(s)

Philipp Hermann <philipp.hermann@jku.at>, Andreas Futschik

References

Knaus BJ and Grünwald NJ (2017). VCFR: a package to manipulate and visualize variant call format data in R. Molecular Ecology Resources, 17(1), pp. 44-53. ISSN 757, <URL: http://dx.doi.org/10.1111/1755-0998.12549>.

See Also

LDJump, summary_statistics, fgt_rrate_dpr, get_smuce, vcfR2DNAbin

Examples

```
##### Do not run these examples                                #####
##### vcfR_to_fasta (seqName, refName, ext.ind = T, cons = F,  #####
#####                ext.haps = T, start = 1)                  #####
```

Index