Package 'LDJump'

August 30, 2019

Type Package

Title Estimating Variable Recombination Rates from Population Genetic Data

Version 0.3.1

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,
pegas,
mgcv

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

Author Philipp Hermann, Andreas Futschik, Fardokhtsadat Mohammadi

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. Moreover, populations can be simulated under user input demographic scenarios in order to train the regression model of constant recombination rates.

License MIT + file LICENSE

Encoding UTF-8

LazyData TRUE

RoxygenNote 6.1.1

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

SystemRequirements Unix Operating System, PhiPack, LDhat (Version 2.2, optional), dos2unix, awk, [For simulations: scrm, ms2dna (Version 1.16)]

R topics documented:

`calcRegMod` *Calculate Regression Model under User Input Demography (Scenario)*

Description

This function computes the regression model for user input demographic scenarios. Moreover, the user is able to handle the sample sizes, lengths, and recombination rates of the simulated populations.

Usage

```
calcRegMod(n = c(10,16,20), len = c(500,1000,2000,3000,5000), thth = 0.01, nsim = 100,
       fr = c(), pathToScrm, scenario, pathToMs2dna, status = T, pathLDhat, pathPhi)
```

Arguments

`n`	A numeric vector containig by default 10, 16, and 20 reflecting the sample sizes of the simulated populations. It can be adapted to any vector.
`len`	A numeric vector containing the lengths of simulated sequences of the populations. By default 0.5, 1, 2, 3, and 5 kb but can be adapted to any integer values.
`thth`	A numeric value for the mutation rate theta under which the populations are simulated. By default 0.01 but can be adapted to any numeric value.
`fr`	A numeric vector containing the recombination rates under which one wants to simulate. By default it is set to an empty vector and uniform random variables are simulated from 5 intervals with `nsim` values per interval.
`nsim`	An integer value for the number of replications (populations) simulated per setup. Setups result from all combinations of sample sizes and sequence lengths. This value can be adapted to any integer value.
`pathToScrm`	A character string containing the path to scrm. This path and the installation of scrm is necessary for the computation of the function.

scenario	A character string containing the demography model (scenario) under which the populations should be simulated. We refer to scrm for details on how to define varying population sizes using the simulation package scrm.
pathToMs2dna	A character string containing the path to ms2dna. This path and the installation of ms2dna is necessary for the computation of the function.
status	an optional logical value: by default TRUE such that the current processing status of the number of simulated populations is printed.
pathLDhat	A character string containing the path to LDhat. This path and the installation of LDhat is necessary for the computation of the package.
pathPhi	A character string containing the path to PhiPack. This path and the installation of PhiPack is necessary for the computation of the package.

Value

regMod	The generalized additive regression model on the box-cox transformed true recombination rates using computed summary statistics from simulated populations under a user defined demography (scenario).
data.all	A data-frame containing all summary statistics per column and simulated samples of populations per row.

Note

This function only works with unix and having PhiPack installed. Optionally when also having LDhat (Auton and McVean (2007)) installed LDJump will compute estimates much faster. Hence, please properly check all paths to PhiPack and in case also LDhat as well as the sequence files. Moreover, the software packages scrm and ms2dna need to be installed for simulating populations under a user input demography (scenario).

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.

Bruen, T. C., Philippe, H., and Bryant, D. (2006). A simple and robust statistical test for detecting the presence of recombination. Genetics, 172(4):2665-2681.

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.

Hermann, P., Heissl, A., Tiemann-Boege, I., and Futschik, A. (2019), LDJump: Estimating Variable Recombination Rates from Population Genetic Data. Mol Ecol Resour. doi:10.1111/1755-0998.12994.

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, doi: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

link{LDJump}, summary_statistics, vcfR_to_fasta, getPhi, get_smuce, smuceR, rq, gam, vcfR2DNAbin, diseq, genotype, readDNAStringSet

Examples

```
##### Do not run these examples                                          #####
##### scenario =  " -eG 0.0 0 -eG 0.42 -100 -eG 0.5 100 "                #####
##### simulatedData = calcRegMod(nsim=100,pathToScrm="/path/To/Scrm/",   #####
#####                 scenario=scenario,pathToMs2dna="/path/To/Ms2dna/", #####
#####                  pathLDhat = "/path/to/LDhat/",                    #####
#####                  pathPhi = "/path/to/Phi/")                        #####
##### regMod = simulatedData[[1]]                                        #####
##### result = LDJump(fileName, alpha = 0.05, segLength = 1000,          #####
#####                  pathLDhat = "/path/to/LDhat/",                    #####
#####                  pathPhi = "/path/to/Phi/",                        #####
#####                  format = "fasta", regMod = regMod)                #####
```

---

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, accept, format)
```

Arguments

seqName	A character string containing the full path and the name of the sequence file in fasta of vcf format. It is necessary to add the extension ("fileName.fa", "fileName.fasta", "fileName.vcf") in order to run LDJump. In case that format equals to DNABin the seqName equals to the name of the DNABin-object (without any extension).
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.

accept — an optional logical value: by default FALSE and LDJump checks for segments with less than 2 SNPs and requires user input to proceed. If set to TRUE, the user accepts that the rates for these segments (<= 1 SNP) are estimated via imputation.

format — A character string which can be fasta, vcf, or DNAStringSet. 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. For the last format the seqName must equal to the DNABin-object which contains the sequences.

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)                       #####
```

---

getPhi — *Summary Statistics to estimate recombination from PhiPack*

Description

This functions calls the PhiPack software and extracts the four summary statistics MaxChi, NSS and the mean and the variance of Phi.

Usage

```
getPhi(seqName, pathPhi, out, rm)
```

Arguments

seqName	A character string containing the full path and the name of the sequence file in fasta of vcf format. It is necessary to add the extension ("fileName.fa", "fileName.fasta", "fileName.vcf") in order to run LDJump. In case that format equals to DNABin the seqName equals to the name of the DNABin-object (without any extension).
pathPhi	A character string containing the path to PhiPack. This path and the installation of PhiPack is necessary for the computation of the package.
out	an optional character string: by default an empty string "". Can be set to any user-defined string in order to rename all output files used within LDJump and PhiPack. This parameter enables to run LDJump from the same directory without creating interfering files in the working directory.
rm	an optional logical value: by default TRUE such that the internally produced fasta file as well as the output file are deleted shortly before finishing the function. This option is added in order to avoid deleting a file of interest when running the function gethi outside LDJump.

Value

A vector is returned containing the four summary statistics MaxChi, NSS and the mean and the variance of Phi.

References

Bruen, T., Phillipe, H. and Bryant, D. 2006. A quick and robust statistical test to detect the presence of recombination. Genetics 172, 2665–2681.

See Also

LDJump, vcfR_to_fasta, get_smuce

Examples

```
## The function is currently defined as
##getPhi(seqName = seqName, pathPhi = pathPhi)
```

---

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, demography, regMod)
```

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.
demography	an optional character value: by default an empty string ("") indicates that the recombination rate estimation is estimated under neutrality. If "b" the regression model estimated based on samples from populations under a bottleneck is used. If "g" the regression model estimated based on samples from populations under population growth is used. If "d", the regression model estimated based on samples from populations under demography (combination of samples of under growth and bottleneck) is used.
regMod	an optional character string: for the default empty string "" *LDJump* uses an existing regression model (constant population size or simple demography example, depending on demography). In oder to use the regression model estimated by user input demography, then this variable should equal to the name of the regression object. Please see the examples for more details.

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.

Hermann, P., Heissl, A., Tiemann-Boege, I., and Futschik, A. (2019), LDJump: Estimating Variable Recombination Rates from Population Genetic Data. Mol Ecol Resour. doi:10.1111/1755-0998.12994.

See Also

LDJump, vcfR_to_fasta, getPhi, 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, quant = 0.35, segLength = 1000, pathLDhat = "",
      pathPhi = "", format = "fasta", refName = NULL, start = NULL, constant = F,
       rescale = F, status = T, polyThres = 0, cores = 1, accept = F,
       demography = F, regMod = "", out = "", lengthofseq = NULL, chr = NULL,
       startofseq = NULL, endofseq = NULL)
```

Arguments

seqName	A character string containing the full path and the name of the sequence file in fasta of vcf format. It is necessary to add the extension ("fileName.fa", "fileName.fasta", "fileName.vcf") in order to run LDJump. In case that format equals to DNABin the seqName equals to the name of the DNABin-object (without any extension).
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.
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 which is the default value.
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.
pathPhi	A character string containing the path to PhiPack. This path and the installation of PhiPack is necessary for the computation of the package.
format	A character string which can be fasta, vcf, or DNAStringSet. 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. For the last format the seqName must equal to the DNABin-object which contains the sequences.

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"`.

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.

accept — an optional logical value: by default `FALSE` and `LDJump` checks for segments with less than 2 SNPs and requires user input to proceed. If set to `TRUE`, it is accepted that the rates for these segments are estimated via imputation.

demography — an optional logical value: by default `FALSE` indicating that the recombination rate estimation is estimated under neutrality. If `TRUE` the regression model estimated based on samples from populations under a bottleneck followed by rapid growth is used.

regMod — an optional character string: for the default empty string "" `LDJump` uses an existing regression model (constant population size or simple demography example, depending on `demography`). In oder to use the regression model estimated by user input demography, then this variable should equal to the name of the regression object. Please see the examples for more details.

out — an optional character string: by default an empty string "". Can be set to any user-defined string in order to rename all output files used within `LDJump`. This parameter enables to run `LDJump` from the same directory without creating interfering files in the working directory.

lengthofseq — an integer value describing the length of the sequence (Only required when running `LDJump` with VCF-Files). It is used to compute the number of segments and to loop through each segment.

chr — either an integer value between 1-22 or a character value "X"/"Y" describing which chromosome is used to run `LDJump` on (Only required when running `LDJump` with VCF-Files). It is required for the `vcftools` system call in order to slice the VCF-File into several segments.

startofseq — an integer value describing at which position the sequence to be analyzed starts (Only required when running `LDJump` with VCF-Files). The starting value is provided to `vcftools` to select the appropriate range for splicing the VCF-File into segments.

endofseq — an integer value describing at which position the sequence to be analyzed ends (Only required when running `vcftools` with VCF-Files).The ending value is provided to `vcftools` to select the appropriate range for splicing the VCF-File into segments.

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 PhiPack installed. We strongly recommend to also install LDhat (Auton and McVean (2007)) in order to decrease the computational cost of estimating recombination maps. Please properly check all paths to PhiPack and in case of LDhat as well as the sequence files. Previous versions (older than v 0.2.1) required lookup tables within the pairwise estimate of LDhat. These files should be located in the path "pathToLDhat/LDhat-master/lk_files". Lookup tables are contained in LDhat, but we still provide several lookup tables here. We strongly recommend to use the most recent version of `LDJump` in order to estimate recombination rates.

Author(s)

Philipp Hermann <philipp.hermann@jku.at>, Andreas Futschik, Fardokhtsadat Mohammadi <fardokht.fm@gmail.c

References

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

Bruen, T. C., Philippe, H., and Bryant, D. (2006). A simple and robust statistical test for detecting the presence of recombination. Genetics, 172(4):2665-2681.

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.

Hermann, P., Heissl, A., Tiemann-Boege, I., and Futschik, A. (2019), LDJump: Estimating Variable Recombination Rates from Population Genetic Data. Mol Ecol Resour. doi:10.1111/1755-0998.12994.

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, doi: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, getPhi, get_smuce, smuceR, rq, gam, vcfR2DNAbin, diseq, genotype, readDNAStringSet, calcRegMod

Examples

```
##### Do not run these examples                                                    #####
##### result = LDJump(fileName, alpha = 0.05, segLength = 1000,                    #####
#####               pathLDhat = getwd(), format = "fasta")                         #####
##### plot(results)                                                                #####
##### results = LDJump("/pathToSample/HatLandscapeN16Len1000000Nrhs15_th0.01_540_1.fa",    #####
##### alpha = 0.05, segLength = 1000, pathLDhat = "/pathToLDhat", pathPhi = "/pathToPhi",  #####
##### format = "fasta", refName = NULL                                             #####
```

---

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)                       #####
```

mod.full.demo — *Regression model to Estimate (constant) Recombination Rates from Population Genetic Summary Statistics under Demography*

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 from populations simulated under demography.

Usage

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

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.demo)
##### Do not run these examples                        #####
##### In get_smuce the function is called as follows   #####
##### pr1 = predict(mod.full.demo,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,
                   pathLDhat, pathPhi, status, polyThres, out, format, startofseq)
```

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 full path and the name of the sequence file in `fasta` of `vcf` format. It is necessary to add the extension ("fileName.fa", "fileName.fasta", "fileName.vcf") in order to run `LDJump`. In case that `format` equals to `DNABin` the seqName equals to the name of the `DNABin`-object (without any extension).
`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.
`pathLDhat`	A character string containing the path to LDhat. This path and the installation of LDhat is necessary for the computation of the package.
`pathPhi`	A character string containing the path to PhiPack. This path and the installation of PhiPack 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).
out	an optional character string: by default an empty string "". Can be set to any user-defined string in order to rename all output files used within LDJump. This parameter enables to run LDJump from the same directory without creating interfering files in the working directory.
format	a character string describing the format of the used file g.e. "fasta" or "vcf". The default is set to "fasta".
startofseq	an integer value describing at which position the sequence to be analyzed starts (Only required when running LDJump with VCF-Files). The starting value is provided to vcftools to select the appropriate range for splicing the VCF-File into segments. In summary_statistics, the same value is used to loop over each FASTA-segment.

Value

This function returns a concatenated vector of all computed summary statistis as:

hahe	The haplotype heterozygosity of the considered segment. Returned with stats.
tajd	Tajima's D. Only used in the regression model for demography.
haps	The number of haplotypes. Later on it is normalized by sequence length and number of individuals.
apwd	Average pairwise differences. Later it is normalized by sequence length.
vapw	Variance of pairwise differences. Later it is normalized by sequence length.
wath	Watterson's theta. Later it is normalized by sequence length.
phis	A vector containing the four summary statistics obtained from PhiPack as Max-Chi, NSS, mean(Phi) and var(Phi).

Author(s)

Philipp Hermann <philipp.hermann@jku.at>, Andreas Futschik, Fardokhtsadat Mohammadi <fardokht.fm@gmail.c

References

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

Bruen, T. C., Philippe, H., and Bryant, D. (2006). A simple and robust statistical test for detecting the presence of recombination. Genetics, 172(4):2665-2681.

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

Hermann, P., Heissl, A., Tiemann-Boege, I., and Futschik, A. (2019), LDJump: Estimating Variable Recombination Rates from Population Genetic Data. Mol Ecol Resour. doi:10.1111/1755-0998.12994.

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, getPhi, 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, ref= NULL, fa_start = NULL,
              fa_end = NULL, attr_name = NULL)
```

Arguments

seqName	A character string containing the full path and the name of the sequence file. It is necessary to add the extension in order to run LDJump (seqName = "fileName.vcf").
refName	An (optional) full path including file name and extension (".vcf") to the reference sequence for the region of interest downloaded from e.g. http://phase3browser.1000genomes.org/inde 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".
ref	A character string describing the name of the reference sequence. If the working directory is not set to the location of the file, the complete path to the file has to be provided g.e. ref = "/home/LDJump/refseq.fa". The reference sequence is needed as it is used together with the vcfR-package to convert each VCF-segment into a FASTA-file.
fa_start	An integer value used to subset the reference sequence when converting VCF-segments to FASTA. It doesn't have to be provided in the function call, but rather it is initialized and computed inside the function vcf_statistics.
fa_end	An integer value used to subset the reference sequence when converting VCF-segments to FASTA. It doesn't have to be provided in the function call, but rather it is initialized and computed inside the function vcf_statistics.

attr_name — A character string describing the chromosome number of the reference file. For example, we have a FASTA-header ">21 dna:chromosome:GRCh37:21:41000000:41010000:1" in our reference file, which describes our file to be a segment of chromosome 21, ranging from 41000000 to 41010000. In vcf_statistics, we use this information to retrieve the chromosome number "21" for the conversion step.

Value

A print command provides information that the file is converted.

Author(s)

Philipp Hermann <philipp.hermann@jku.at>, Andreas Futschik, Fardokhtsadat Mohammadi <fardokht.fm@gmail.c

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, getPhi, get_smuce, vcfR2DNAbin

Examples

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

---

vcf_statistics *Calculating statistics on VCF-files.*

Description

This function estimates variable recombination rates from population genetic data using VCF-files. 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

```
vcf_statistics(seqName, alpha = 0.05, quant = 0.35, segLength = 1000, pathLDhat = "",
      pathPhi = "", format = "fasta", refName = NULL, start = NULL, constant = F,
       rescale = F, status = T, polyThres = 0, cores = 1, accept = F,
       demography = F, regMod = "", out = "", lengthofseq = NULL,
       chr = NULL, startofseq = NULL, endofseq = NULL)
```

Arguments

seqName	A character string containing the full path and the name of the sequence file in fasta of vcf format. It is necessary to add the extension ("fileName.fa", "fileName.fasta", "fileName.vcf") in order to run LDJump. In case that format equals to DNABin the seqName equals to the name of the DNABin-object (without any extension).
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.
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 which is the default value.
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.
pathPhi	A character string containing the path to PhiPack. This path and the installation of PhiPack is necessary for the computation of the package.
format	A character string which can be fasta, vcf, or DNAStringSet. 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. For the last format the seqName must equal to the DNABin-object which contains the sequences.
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".
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.
accept	an optional logical value: by default FALSE and LDJump checks for segments with less than 2 SNPs and requires user input to proceed. If set to TRUE, it is accepted that the rates for these segments are estimated via imputation.

demography	an optional logical value: by default FALSE indicating that the recombination rate estimation is estimated under neutrality. If TRUE the regression model estimated based on samples from populations under a bottleneck followed by rapid growth is used.
regMod	an optional character string: for the default empty string "" LDJump uses an existing regression model (constant population size or simple demography example, depending on demography). In oder to use the regression model estimated by user input demography, then this variable should equal to the name of the regression object. Please see the examples for more details.
out	an optional character string: by default an empty string "". Can be set to any user-defined string in order to rename all output files used within LDJump. This parameter enables to run LDJump from the same directory without creating interfering files in the working directory.
lengthofseq	an integer value describing the length of the sequence (Only required when running LDJump with VCF-Files). It is used to compute the number of segments and to loop through each segment.
chr	either an integer value between 1-22 or a character value "X"/"Y" describing which chromosome is used to run LDJump on (Only required when running LDJump with VCF-Files). It is required for the vcftools system call in order to slice the VCF-File into several segments.
startofseq	an integer value describing at which position the sequence to be analyzed starts (Only required when running LDJump with VCF-Files). The starting value is provided to vcftools to select the appropriate range for splicing the VCF-File into segments.
endofseq	an integer value describing at which position the sequence to be analyzed ends (Only required when running vcftools with VCF-Files).The ending value is provided to vcftools to select the appropriate range for splicing the VCF-File into segments.

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 PhiPack installed. We strongly recommend to also install LDhat (Auton and McVean (2007)) in order to decrease the computational cost of estimating recombination maps. Please properly check all paths to PhiPack and in case of LDhat as well as the sequence files. Previous versions (older than v 0.2.1) required lookup tables within the pairwise estimate of LDhat. These files should be located in the path "pathToLDhat/LDhat-master/lk_files". Lookup tables are contained in LDhat, but we still provide several lookup tables here. We strongly recommend to use the most recent version of `LDJump` in order to estimate recombination rates.

Author(s)

Philipp Hermann <philipp.hermann@jku.at>, Andreas Futschik, Fardokhtsadat Mohammadi <fardokht.fm@gmail.c

References

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

Bruen, T. C., Philippe, H., and Bryant, D. (2006). A simple and robust statistical test for detecting the presence of recombination. Genetics, 172(4):2665-2681.

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.

Hermann, P., Heissl, A., Tiemann-Boege, I., and Futschik, A. (2019), LDJump: Estimating Variable Recombination Rates from Population Genetic Data. Mol Ecol Resour. doi:10.1111/1755-0998.12994.

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, doi: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`, `getPhi`, `get_smuce`, `smuceR`, `rq`, `gam`, `vcfR2DNAbin`, `diseq`, `genotype`, `readDNAStringSet`, `calcRegMod`

Examples

```
##### Do not run these examples                                                    #####
##### result = LDJump(fileName, alpha = 0.05, segLength = 1000,                    #####
#####               pathLDhat = getwd(), format = "fasta")                         #####
##### plot(results)                                                                #####
##### results = LDJump("/pathToSample/HatLandscapeN16Len1000000Nrhs15_th0.01_540_1.fa",    #####
##### alpha = 0.05, segLength = 1000, pathLDhat = "/pathToLDhat", pathPhi = "/pathToPhi",  #####
##### format = "fasta", refName = NULL                                             #####
```

Index