Package 'GenomeAdapt'

October 12, 2022

Type Package

Version 1.0.0

Date 2021-11-01

Title Detecting Signatures of Local Adaptation Based on Ancestry Trajectories

Description Portable, scalable and highly computationally efficient tool for detecting signatures of local adaptation based on multidimensional ancestry map (_n_ X _n_ ancestry genetic trajectories, _n_ is the number of individuals). If n samples are included in the analysis, there will be n dimensional spaces that represent the common ancestry maps based on the identity-by-descent (IBD). The package calculates the correlations of loci with the common ancestry genetic maps adopting the Genomic Data Structure (GDS, Zheng et al., 2012) <doi:10.1093/bioinformatics/bts606> and suitable for millions of SNP data. Loci sharing a greater level of most recent common ancestor (MRCA) (large Z-scores) indicates a large number of individuals descend from recent common ancestors, which signals the rapid increase in frequency of a beneficial allele due to recent positive selection. The rationale underlying this package is somewhat analogous to KLFDAPT (Qin, 2021) <doi:10.1101/2021.05.15.444294> (<https://xinghuq.github.io/KLFDAPC/articles/Genome_scan_KLFDAPC.html>). It combines the concept of IBD-based genome scan (Albrechtsen et al., 2010) <doi:10.1534/genetics.110.113977>, iHS (Voight, 2006) <doi:10.1371/journal.pbio.0040072>, and eigenanalysis of SNP data with an identity by descent interpretation (Zheng & Weir, 2016) <doi:10.1016/j.tpb.2015.09.004>. It can also be interpreted as spatial varying selection as ancestry genetic maps reflect geographic origins. Besides the detection of local adaptation, this package also estimates the population admixtures and plots its geographic genetic structure.

biocViews

Depends R (>= 3.3)

License GPL-3

URL https://github.com/xinghuq/GenomeAdapt

BugReports https://github.com/xinghuq/GenomeAdapt/issues

Imports qvalue,robust,stats,SNPRelate,gdsfmt,graphics

VignetteBuilder knitr

NeedsCompilation no

RoxygenNote 6.1.1

Suggests knitr,testthat,rmarkdown

Author Xinghu Qin [aut, cre, cph] (<https://orcid.org/0000-0003-2351-3610>)

Maintainer Xinghu Qin <qinxinghu@gmail.com>

Repository CRAN

Date/Publication 2021-11-11 19:40:08 UTC

R topics documented:

`AdmixProp` *Estimate ancestral proportions from the eigen vectors*

Description

Estimate ancestral (admixture) proportions based on the eigen-analysis

Usage

```
AdmixProp(x, groups, bound = FALSE)
```

Arguments

`x`	an object from GenomeAdapt
`groups`	A list of sample IDs, such like groups = list(CEU = c("NA0101", "NA1022", ...), YRI = c("NAxxxx", ...), Asia = c("NA1234", ...))
`bound`	if TRUE, the estimates are bounded so that no component < 0 or > 1, and the sum of proportions is one

Details

Estimate ancestral (admixture) proportions based on the eigen-analysis, results give a dataframe of individual probability from the ancestries.

Value

A dataframe of individual probability from the ancestries.

References

Zheng X, Weir BS. Eigenanalysis on SNP Data with an Interpretation of Identity by Descent. Theoretical Population Biology. 2015 Oct 23. pii: S0040-5809(15)00089-1. doi: 10.1016/j.tpb.2015.09.004.

Zheng, X., Levine, D., Shen, J., Gogarten, S. M., Laurie, C., & Weir, B. S. (2012). A high-performance computing toolset for relatedness and principal component analysis of SNP data. Bioinformatics, 28(24), 3326-3328.

Examples

```
##---- Do genome scan and get the ancestry proportation for individuals ----
# Scan genomes (HapMap)

HapmapScan=GenomeAdapt.gds(genfile =SNPRelate::snpgdsExampleFileName(),
method="EIGMIX",num.thread = 1L, autosome.only=TRUE,
remove.monosnp=TRUE, maf=0.01, missing.rate=0.1)


# get population information

genofile <- SNPRelate::snpgdsOpen(SNPRelate::snpgdsExampleFileName())
pop_code <- gdsfmt::read.gdsn(gdsfmt::index.gdsn(genofile, "sample.annot/pop.group"))

# get sample id
samp.id <- gdsfmt::read.gdsn(gdsfmt::index.gdsn(genofile, "sample.id"))
SNPRelate::snpgdsClose(genofile)

# define groups
groups <- list(CEU = samp.id[pop_code == "CEU"],
    YRI = samp.id[pop_code == "YRI"],
    CHB = samp.id[is.element(pop_code, c("HCB", "JPT"))])

### estimate the ancestry proportion

Admixpro=AdmixProp(HapmapScan,groups=groups,bound=TRUE)

PlotAdmix(Admixpro,group=as.factor(pop_code),multiplot = FALSE)
```

`GenomeAdapt` *Detecting signatures of local adaptation based on multidimensional ancestry genetic map*

Description

This function implements genome scan to identify the signatures of local adaptation. See details.

Usage

```
GenomeAdapt(genfile, method = "EIGMIX", sample.id = NULL, snp.id =
                    NULL, autosome.only = TRUE, remove.monosnp = TRUE, maf
                    = NaN, missing.rate = NaN, num.thread = 1L, out.fn =
                    NULL, out.prec = c("double", "single"), out.compress =
                    "LZMA_RA", with.id = TRUE, verbose = TRUE,...)

## S3 method for class 'GenomeAdapt.bed'
GenomeAdapt.bed(genfile, method = "EIGMIX", sample.id = NULL, snp.id =
                    NULL, autosome.only = TRUE, remove.monosnp = TRUE, maf
                    = NaN, missing.rate = NaN, num.thread = 1L, out.fn =
                    NULL, out.prec = c("double", "single"), out.compress =
                    "LZMA_RA", with.id = TRUE, verbose = TRUE, ...)

## S3 method for class 'GenomeAdapt.vcf'
GenomeAdapt.vcf(genfile, method = "EIGMIX", sample.id = NULL, snp.id =
                    NULL, autosome.only = TRUE, remove.monosnp = TRUE, maf
                    = NaN, missing.rate = NaN, num.thread = 1L, out.fn =
                    NULL, out.prec = c("double", "single"), out.compress =
                    "LZMA_RA", with.id = TRUE, verbose = TRUE, ...)

## S3 method for class 'GenomeAdapt.gds'
GenomeAdapt.gds(genfile, method = "EIGMIX", sample.id = NULL, snp.id =
                    NULL, autosome.only = TRUE, remove.monosnp = TRUE, maf
                    = NaN, missing.rate = NaN, num.thread = 1L, out.fn =
                    NULL, out.prec = c("double", "single"), out.compress =
                    "LZMA_RA", with.id = TRUE, verbose = TRUE, ...)
```

Arguments

genfile	Genotype file containg sample ID and SNP ID. Genotype format can be plink, vcf, or GDS file.
method	The method used to measure IBD. Default is "EIGMIX" according to Zheng, X., & Weir, B. S. (2016). "GCTA" - genetic relationship matrix defined in CGTA; "Eigenstrat" - genetic covariance matrix in EIGENSTRAT; "EIGMIX" - two times coancestry matrix defined in Zheng & Weir (2015), "Weighted" - weighted GCTA, as the same as "EIGMIX", "Corr" - Scaled GCTA GRM (dividing each i,j element by the product of the square root of the i,i and j,j elements), "Indiv-Beta" - two times individual beta estimate relative to the minimum of beta.
sample.id	a vector of sample id specifying selected samples; if NULL, all samples are used
snp.id	a vector of snp id specifying selected SNPs; if NULL, all SNPs are used
autosome.only	use autosomal SNPs only; if it is a numeric or character value, keep SNPs according to the specified chromosome
remove.monosnp	remove monomorphic SNPs
maf	filter SNPs with ">= maf" only; if NaN, no MAF threshold
missing.rate	filter the SNPs with "<= missing.rate" only; if NaN, no missing threshold

num.thread	the number of (CPU) cores used; if NA, detect the number of cores automatically
out.fn	NULL for no GDS output, or a file name
out.prec	double or single precision for storage
out.compress	the compression method for storing the GRM matrix in the GDS file
with.id	if TRUE, the returned value with sample.id and sample.id
verbose	if TRUE, show information
...	passing to other SNP filtering parameters

Details

The method estimates the z-score of each locus/allele relating to the multidimensional ancestry spaces. If there are **n** samples, there will be **n X n** ancestry genetic trajectories, with an eigen decomposition, producting **n** dimensional spaces that represent the common ancestry maps. This method was conceived combining the idea of KLFDAPT (Qin, 2021) https://xinghuq.github.io/KLFDAPC/articles/Genome_scan_KLFDAPC.html and IBD-based genome scan (Albrechtsen et al., 2010). With an eigenvector decomposition of IBD (Zheng & Weir 2016), we can estimate the population ancestry propotion. It is competitive to pcadapt (Luu, 2016), as it considers **n** latent genetic spaces (which is different from pcadapt that chooses **k** components from **p** eigenverctors).

Value

A GenomeAdapt class, containing the loci z-scores and the eigen analysis results of IBD representing the genetic structure.

zscores	The locus Z-scores relating to n latent ancestry genetic spaces, n is equal to the number of individuals
eig	Eigen analysis of IBD represent ancestry or population genetic structure
chr	Chromosomes

Author(s)

qin.xinghu@163.com

References

Qin, X., Chiang, C. W., & Gaggiotti, O. E. (2021). Kernel Local Fisher Discriminant Analysis of Principal Components (KLFDAPC) significantly improves the accuracy of predicting geographic origin of individuals.bioRxiv.

Zheng, X., & Weir, B. S. (2016). Eigenanalysis of SNP data with an identity by descent interpretation. Theoretical population biology, 107, 65-76.

Albrechtsen, A., Moltke, I., & Nielsen, R. (2010). Natural selection and the distribution of identity-by-descent in the human genome. Genetics, 186(1), 295-308.

Duforet-Frebourg, N., Luu, K., Laval, G., Bazin, E., & Blum, M. G. (2016). Detecting genomic signatures of natural selection with principal component analysis: application to the 1000 genomes data. Molecular biology and evolution, 33(4), 1082-1093.

Examples

```
### using an example dataset (HapMap)to conduct the genome scan

HapmapScan=GenomeAdapt.gds(genfile = SNPRelate::snpgdsExampleFileName(),method="EIGMIX",
num.thread = 1L, autosome.only=TRUE, remove.monosnp=TRUE, maf=0.01, missing.rate=0.1)
```

PlotAdmix *Plot Ancestry Admixture*

Description

Plot admixture proportions based on their ancestries.

Usage

```
PlotAdmix(propmat, group = NULL, col = NULL,
multiplot = TRUE, xlab = "Individuals", ylab = "Ancestry Proportion",
showgrp = TRUE, shownum = TRUE, ylim = TRUE, na.rm = TRUE)
```

Arguments

propmat	a sample-by-ancestry matrix of proportion estimates, returned from AdmixProp()
group	a character vector of a factor according to the samples in propmat
col	specify colors
multiplot	single plot or multiple plots
xlab	The title for the x axis
ylab	The title for the y axis
showgrp	show group names in the plot
shownum	TRUE: show the number of each group in the figure
ylim	TRUE: y-axis is limited to [0, 1]; FALSE: ylim <- range(propmat); a 2-length numeric vector: ylim used in plot()
na.rm	TRUE: remove the sample(s) according to the missing value(s) in group

Details

The minor allele frequency and missing rate for each SNP passed in snp.id are calculated over all the samples in sample.id.

Value

Return to an anscrstry proportion plot/population structure plot that the same as STRUCTURE plot.

References

Zheng X, Weir BS. Eigenanalysis on SNP Data with an Interpretation of Identity by Descent. Theoretical Population Biology. 2015 Oct 23. pii: S0040-5809(15)00089-1. doi: 10.1016/j.tpb.2015.09.004.

Zheng, X., Levine, D., Shen, J., Gogarten, S. M., Laurie, C., & Weir, B. S. (2012). A high-performance computing toolset for relatedness and principal component analysis of SNP data. Bioinformatics, 28(24), 3326-3328.

Examples

```
##---- Do genome scan and get the ancestry proportation for individuals ----
# Scan genomes (HapMap)

HapmapScan=GenomeAdapt.gds(genfile = SNPRelate::snpgdsExampleFileName(),
method="EIGMIX",num.thread = 1L, autosome.only=TRUE,
remove.monosnp=TRUE, maf=0.01, missing.rate=0.1)


# get population information
genofile <- SNPRelate::snpgdsOpen(SNPRelate::snpgdsExampleFileName())
pop_code <- gdsfmt::read.gdsn(gdsfmt::index.gdsn(genofile, "sample.annot/pop.group"))

# get sample id
samp.id <- gdsfmt::read.gdsn(gdsfmt::index.gdsn(genofile, "sample.id"))
SNPRelate::snpgdsClose(genofile)

# define groups
groups <- list(CEU = samp.id[pop_code == "CEU"],
    YRI = samp.id[pop_code == "YRI"],
    CHB = samp.id[is.element(pop_code, c("HCB", "JPT"))])

### estimate the ancestry proportion

Admixpro=AdmixProp(HapmapScan,groups=groups,bound=TRUE)

PlotAdmix(Admixpro,group=as.factor(pop_code),multiplot = FALSE)
```

`plotmanhattan` *Making manhattan plot*

Description

Ploting a Manhattan plot showing the (q)p-values for each SNP.

Usage

```
plotmanhattan(x, ylim=c(0,200),xlab="",
ylab="-log(p-value)",col = x$chr, pch="*",h=10, lcol="blue", ...)
```

Arguments

x	The zscores_qvals object
ylim	The y axis range
xlab	The x title
ylab	The y title
col	The colour used for indicating chromosomes
pch	The shape of of the data points.
h	The cutoff line
lcol	The colour of the cutoff line
...	passing to other parameters of plot

Value

Return to a manhattan plot with (q)p-values for each SNP.

Examples

```
### using Hapmap data
HapmapScan=GenomeAdapt.gds(genfile = SNPRelate::snpgdsExampleFileName(),
method="EIGMIX",num.thread = 1L, autosome.only=TRUE,
remove.monosnp=TRUE, maf=0.01, missing.rate=0.1)

### Not running, it takes a while to finish this
 Hapmapqval=zscores_qvals(HapmapScan)

## plot
plotmanhattan(Hapmapqval$pvals$pvals$p.values,col=Hapmapqval$chr)
```

---

zscores_qvals *Calculating (q)p-values from multiple factors/scores*

Description

Converting the Z-scores to (q)p-values. This function calibrates the p-value/q-values considering multiple scores based on Mahalanobis Distance/componentwise method.

Usage

```
zscores_qvals(x, outlier.method = "mahalanobis",
estim = "pairwiseGK", pval.coret.method = "bonferroni")
```

Arguments

x	The GenomeAdapt object containing the locus scores
outlier.method	The methods used to detect the outliers, "mahalanobis" or "componentwise", default is mahalanobis
estim	Method used to estimate Mahalanobis distance. The choices are : "mcd" for the Fast MCD algorithm of Rousseeuw and Van Driessen, "weighted" for the Reweighted MCD, "donostah" for the Donoho-Stahel projection based estimator, "M" for the constrained M estimator provided by Rocke, "pairwiseQC" for the orthogonalized quadrant correlation pairwise estimator, and "pairwiseGK" for the Orthogonalized Gnanadesikan-Kettenring pairwise estimator. The default "auto" selects from "donostah", "mcd", and "pairwiseQC" with the goal of producing a good estimate in a reasonable amount of time.
pval.coret.method	Correction method for p-values,choices are "holm", "hochberg", "hommel", "bonferroni", "BH", "BY","fdr", "none".The adjustment methods include the Bonferroni correction ("bonferroni") in which the p-values are multiplied by the number of comparisons. Less conservative corrections are also included by Holm (1979) ("holm"), Hochberg (1988) ("hochberg"), Hommel (1988) ("hommel"), Benjamini & Hochberg (1995) ("BH" or its alias "fdr"), and Benjamini & Yekutieli (2001) ("BY"), respectively. A pass-through option ("none") is also included.

Details

Calculating (q)p-values from GenomeAdapt to identify the outlier loci

Value

A dataframe with p-values, adjusted p-values and q-values.

pvals	A dataframe containing 3 columns, including the p-values, q-values, and adjusted p-values for all loci
chr	The chromsomes for the dataset

References

Benjamini, Y., and Hochberg, Y. (1995). Controlling the false discovery rate: a practical and powerful approach to multiple testing. Journal of the Royal Statistical Society Series B, 57, 289-300. http://www.jstor.org/stable/2346101.

R. A. Maronna and R. H. Zamar (2002) Robust estimates of location and dispersion of high-dimensional datasets. Technometrics 44 (4), 307-317.

Capblancq, T., Luu, K., Blum, M. G., & Bazin, E. (2018). Evaluation of redundancy analysis to identify signatures of local adaptation. Molecular Ecology Resources, 18(6), 1223-1233.

Examples

```
##---- Do genome scan ----
HapmapScan=GenomeAdapt.gds(genfile = SNPRelate::snpgdsExampleFileName(),
method="EIGMIX",num.thread = 1L, autosome.only=TRUE,
remove.monosnp=TRUE, maf=0.01, missing.rate=0.1)

## estimating the q-values from genome scan

Hapmapqval=zscores_qvals(HapmapScan)
```

Index