Package 'GARCOM'

October 16, 2020

Type Package

Title Gene and Region Counting of Mutations (``GARCOM'')

Version 1.2.0

Description Gene and Region Counting of Mutations (GARCOM) package computes mutation (or alleles) counts per gene per individuals based on gene annotation or genomic base pair boundaries. It comes with features to accept data formats in plink(.raw) and VCF. It provides users flexibility to extract and filter individuals, mutations and genes of interest.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Imports data.table (>= 1.12.8), stats, vcfR (>= 1.12.0)

Suggests testthat

Depends R (>= 2.10)

NeedsCompilation no

Author Sanjeev Sariya [aut, cre, cph],
Giuseppe Tosto [aut, cph]

Maintainer Sanjeev Sariya <ss5505@cumc.columbia.edu>

Repository CRAN

Date/Publication 2020-10-16 13:40:02 UTC

R topics documented:

genecoord 2
gene_annot_counts 2
gene_pos_counts 4
recodedgen 6
snpgene 6
snppos 7
vcf_counts_annot 7
vcf_counts_SNP_genecoords 8

Index **10**

`genecoord` *genecoord*

Description

sample data for gene base pair boundaries

Usage

```
genecoord
```

Format

An object of class `data.frame` with 5 rows and 3 columns.

Details

genecoord is example data provided with the GARCOM package. It has 3 columns and 5 rows. Column names are GENE, START and END where GENE column contains gene name, START and END indicate start BP and end BP respectively.

`gene_annot_counts` *gene annotation counts*

Description

The function returns a matrix with allelic counts per gene per individual for annotated SNPs

Usage

```
gene_annot_counts(dt_gen,dt_snpgene,keep_indiv=NULL,
extract_SNP=NULL,filter_gene=NULL,
impute_missing=FALSE,impute_method="mean")
```

Arguments

`dt_gen`	a dataframe for genetic data that follows PLINK format (.raw)
`dt_snpgene`	a dataframe that contains SNP and annotated gene with SNP and GENE as column name
`keep_indiv`	an option to specify individuals to retain. Mutation counts will be provided for individuals included in the list only. Default is all individuals. Provide list of individuals in a vector.

extract_SNP	an option to specify SNPs for which mutation counts are needed. Mutation counts will be provided for SNPs provided in the list only. Default all SNPs are used. Provide list of SNPs in a vector.
filter_gene	an option to filter in a list of Genes. Mutation counts will be provided for genes specifed in the list only. Default is all genes. Provide list of genes in a vector.
impute_missing	an option to impute missing genotypes. Default is FALSE.
impute_method	an option to specify imptuation method. Default method is imputation to the mean. Alternatively imputation can be carried out by median. Function accepts method in quotes: "mean" or "median". Data are rounded to the second decimal places (e.g. 0.1234 will become 0.12).

Details

Inputs needed are recoded genetic data formatted in PLINK format (.raw) and SNP-gene annotation data. The first six columns of the input genetic data follow standard PLINK .raw format. Column names as FID, IID, PAT, MAT, SEX and PHENOTYPE followed by SNP information as recoded by the PLINK software. SNP-gene data has two columns: GENE and SNP names. The function returns allelic counts per gene per sample (where each row represents a gene and each column represents an individual starting with the second column where first column contains gene information).

Value

Returns an object of data.table class as an output with allelic gene counts within each sample where each row corresponds to gene and column to individual IDs from column second. The first column contains gene names.

Author(s)

Sanjeev Sariya

Examples

```
#Package provides sample data that are loaded with package loading.

data(recodedgen) #PLINK raw formatted data of 10 individiduals with 10 SNPs

data(snpgene) #SNP and its respective GENE annotated.
#Here 10 SNPs are shown annotated in five genes.
#A SNP can be annotated in multiple genes.

gene_annot_counts(recodedgen,snpgene) #run the function

#subset Genes
gene_annot_counts(recodedgen,snpgene,filter_gene=c("GENE1","GENE2"))

#Subset individuals
gene_annot_counts(recodedgen, snpgene,keep_indiv=c("IID_sample1","IID_sample8"))

#subset with genes and samples
```

```
gene_annot_counts(recodedgen,snpgene,filter_gene=c("GENE1","GENE2"),
keep_indiv=c("IID_sample1","IID_sample8"))

#impute missing using default method.

gene_annot_counts(recodedgen,snpgene,impute_missing=TRUE)

#Subset on individuals and impute for missing values. Default as mean
gene_annot_counts(recodedgen,snpgene,impute_missing=TRUE,
keep_indiv=c("IID_sample1","IID_sample2","IID_sample10"))

#impute using median method
gene_annot_counts(recodedgen,snpgene,impute_missing=TRUE,impute_method="median")

#end not RUN
```

`gene_pos_counts` *gene position counts*

Description

Function returns matrix with allelic counts per gene per individual for SNP and gene coordinates as inputs

Usage

```
gene_pos_counts(dt_gen,dt_snp,dt_gene,keep_indiv=NULL,
extract_SNP=NULL,filter_gene=NULL,
impute_missing=FALSE,impute_method="mean")
```

Arguments

`dt_gen`	a dataframe for genetic data that follows PLINK format (.raw)
`dt_snp`	a dataframe for SNP information with SNP BP as column names.
`dt_gene`	a dataframe for gene boundaries with CHR START END GENE as column names. Where CHR should be integer 1-22. START and END column should be integer. GENE column contains gene names
`keep_indiv`	an option to specify individuals to retain. Mutation counts will be provided for individuals provided in the list only. Default is all individuals.
`extract_SNP`	an option to specify SNPs for which mutation counts are needed. Mutation counts will be provided for SNPs included in the list only. Default is all SNPs.
`filter_gene`	an option to filter in Genes. Mutation counts will be provided for genes included in the list only. Default is all genes.
`impute_missing`	an option to impute missing genotypes. Default is FALSE.
`impute_method`	an option to specify method to specify imptuation method. Default method is impute to the mean. Alternatively imputation can be carried out by median. Function accepts method in quotes: "mean" or "median". Data are rounded to the second decimal places (e.g. 0.1234 will become 0.12.).

Details

Inputs needed are: recoded genetic data formatted in PLINK format, SNP name with BP (position) and gene name with START and END position. The first six columns of the input genetic data follow standard PLINK .raw format. Column names as FID, IID, PAT, MAT, SEX and PHENOTYPE followed by SNP information as recoded by the PLINK software. The function returns allelic counts per gene per sample (where each row represents a gene and each column represents an individual starting with the second column where first column contains gene information).

Value

Returns an object of data.table class as an output with allelic gene counts within each sample where each row corresponds to gene and column to individual IDs from column second. The first column contains gene names.

Author(s)

Sanjeev Sariya

Examples

```
#Package provides sample data that are loaded with package loading.
#not RUN
data(recodedgen) #PLINK raw formatted data of 10 individiduals with 10 SNPs

data(genecoord) #gene coordinates with START, END, CHR and GENE names.
#Five genes with start and end genomic coordinates

data(snppos) #SNP and BP column names with SNP names and SNP genomic location in BP.
#10 SNPs with genomic location

gene_pos_counts(recodedgen, snppos, genecoord) #run the function

#subset individuals
gene_pos_counts(recodedgen, snppos, genecoord,keep_indiv=c("IID_sample2","IID_sample4"))

#subset genes
gene_pos_counts(recodedgen,snppos,genecoord,filter_gene=c("GENE1","GENE2"))

#subset genes and individual iids
gene_pos_counts(recodedgen,snppos,genecoord,filter_gene=c("GENE1","GENE2"),
keep_indiv=c("IID_sample10","IID_sample4"))

##impute by mean
gene_pos_counts(recodedgen,snppos,genecoord,impute_missing=TRUE,impute_method="mean")

#end not RUN
```

recodedgen *recodedgen*

Description

sample genetic data

Usage

```
recodedgen
```

Format

An object of class `data.frame` with 10 rows and 16 columns.

Details

recodedgen is sample genetic data provided with the GARCOM package. It has with 10 rows and 16 columns. Where the first 6 columns are FID, IID, PAT, MAT, SEX and PHENOTYPE which are inherent from the PLINK (recode) output. Columns followed by PHENOTYPE are SNP names which are suffixed with _A, or _C or _T or _G. Each SNP column may have 0, 1, 2 or NA value. Where NA represents missingness.

snpgene *snpgene*

Description

sample SNP-gene annotation data

Usage

```
snpgene
```

Format

An object of class `data.frame` with 10 rows and 2 columns.

Details

snpgene is sample SNP-Gene annotation data provided with the GARCOM package. It has 10 rows and 2 columns. Column names are GENE and SNP, where GENE column contains GENE names and SNP column contains SNP name that is annotated with the GENE

snppos *snppos*

Description

sample data for SNP coordinates

Usage

```
snppos
```

Format

An object of class `data.frame` with 10 rows and 2 columns.

Details

snppos is sample SNP-BP data provided with the GARCOM package. It has 2 columns and 10 rows. Column names are SNP and BP. where SNP columns contains SNP names BP column contains position of the SNP

vcf_counts_annot *gene annotation counts using VCF data*

Description

Function returns a matrix with allelic (reference) counts per gene per individual for SNP-gene annotation

Usage

```
vcf_counts_annot(
  vcf_data,
  df_snpgene,
  keep_indiv = NULL,
  extract_SNP = NULL,
  filter_gene = NULL
)
```

Arguments

vcf_data	an object of vcfR class
df_snpgene	a data frame that contains SNP and annotated gene with SNP and GENE as column name
keep_indiv	an option to specify individuals to retain. Mutation counts will be provided for individuals included in the list only. Default is all individuals. Provide list of individuals in a vector.
extract_SNP	an option to specify SNPs for which mutation counts are needed. Mutation counts will be provided for SNPs included in the list only. Default is all SNPs.
filter_gene	an option to filter in a list of Genes. Mutation counts will be provided for genes specifed in the list only. Default is all genes. Provide list of genes in a vector.

Details

Inputs needed are a vcf data and a data frame of SNP-gene annotation. The function returns a matrix of allelic counts (reference) per gene per sample (where each row represents a gene and each column represents an individual starting with the second column where first column contains gene information).

Value

Returns an matrix of data.table class as an output with allelic (reference) gene counts within each sample where each row corresponds to gene and column to individual IDs from column second. The first column contains gene names.

Author(s)

Sanjeev Sariya

Examples

```
## Not run:
vcf_counts_annot(vcf,df_snpgene_test)

## End(Not run)
```

---

vcf_counts_SNP_genecoords

VCF gene position counts

Description

Function returns a matrix with allelic counts per gene per individual for SNP and gene coordinates as inputs

Usage

```
vcf_counts_SNP_genecoords(
  vcf_data,
  df_snppos,
  df_genecoords,
  keep_indiv = NULL,
  extract_SNP = NULL,
  filter_gene = NULL
)
```

Arguments

vcf_data	an object of vcfR class
df_snppos	a dataframe for SNP information with SNP BP as column names.
df_genecoords	a dataframe for gene boundaries with CHR START END GENE as column names. Where CHR should be integer 1-22. START and END column should be integer. GENE column contains gene names
keep_indiv	an option to specify individuals to retain. Mutation counts will be provided for individuals included in the list only. Default is all individuals. Provide list of individuals in a vector.
extract_SNP	an option to specify SNPs for which mutation counts are needed. Mutation counts will be provided for SNPs included in the list only. Default is all SNPs.
filter_gene	an option to filter in Genes. Mutation counts will be provided for genes included in the list only. Default is all genes.

Value

Returns an matrix of data.table class as an output with allelic (reference) gene counts within each sample where each row corresponds to gene and column to individual IDs from column second. The first column contains gene names.

Author(s)

Sanjeev Sariya

Examples

```
## Not run:
vcf_counts_SNP_genecoords(vcf_data_test,df_snppos_test,df_genecoords_test)

## End(Not run)
```

Index

∗ **datasets**
genecoord, 2
recodedgen, 6
snpgene, 6
snppos, 7

gene_annot_counts, 2
gene_pos_counts, 4
genecoord, 2

recodedgen, 6

snpgene, 6
snppos, 7

vcf_counts_annot, 7
vcf_counts_SNP_genecoords, 8