Package ‘prewas’

April 2, 2021

Type Package

Title Data Pre-Processing for Bacterial Genome-Wide Association Studies

Version 1.1.1

Description Standardize the pre-processing of genomic variants before performing a bacterial genome-wide association study (bGWAS). 'prewas' creates a variant matrix (where each row is a variant, each column is a sample, and the entries are presence - 1 - or absence - 0 - of the variant) that can be used as input for bGWAS tools. When creating the binary variant matrix, 'prewas' can perform 3 pre-processing steps including: dealing with multiallelic SNPs, (optional) dealing with SNPs in overlapping genes, and choosing a reference allele. 'prewas' can output matrices for use with both SNP-based bGWAS and gene-based bGWAS. This method is described in Saund et al. (2020) <doi:10.1099/mgen.0.000368>. 'prewas' can also provide gene matrices for variants with specific annotations from the 'SnpEff' software (Cingolani et al. 2012).

URL https://github.com/Snitkin-Lab-Umich/prewas

BugReports https://github.com/Snitkin-Lab-Umich/prewas/issues

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 3.5.0)

RoxygenNote 7.1.1

Imports ape (>= 5.3), future (>= 1.15.1), future.apply (>= 1.3.0), phangorn (>= 2.5.5), stats (>= 3.5.0), vcfR (>= 1.8.0), utils (>= 3.5.0), methods (>= 3.5.0)

Suggests testthat (>= 2.2.1), knitr (>= 1.24), rmarkdown (>= 1.15)

VignetteBuilder knitr

Date 2021-04-01

NeedsCompilation no

Author Katie Saund [aut, cre] (<https://orcid.org/0000-0002-6214-6713>),
Zena Lapp [aut] (<https://orcid.org/0000-0003-4674-2176>),
Stephanie Thiede [aut] (<https://orcid.org/0000-0003-0173-4324>)

Maintainer Katie Saund <katiephd@umich.edu>

Repository CRAN

Date/Publication 2021-04-02 12:20:05 UTC

R topics documented:

`collapse_snps_into_genes` *Collapse SNPs into the gene(s) which they are from*

Description

Collapse SNPs into the gene(s) which they are from

Usage

```
collapse_snps_into_genes(bin_mat, gene_vec)
```

Arguments

`bin_mat`	Matrix.
`gene_vec`	Character. Vector of gene names.

Value

gene_mat: Matrix.

collapse_snps_into_genes_by_impact *Collapse SNPs into the gene(s) which they are from and by snpeff impcat*

Description

Collapse SNPs into the gene(s) which they are from and by snpeff impcat

Usage

```
collapse_snps_into_genes_by_impact(
  bin_mat,
  gene_vec,
  predicted_impact,
  snpeff_grouping
)
```

Arguments

`bin_mat`	Matrix.
`gene_vec`	Character. Vector of gene names.
`predicted_impact`	Character. Vector of predicted functional impacts.
`snpeff_grouping`	Character. Vector or single string of impacts of interest.

Value

a list of gene_mats, collapsed by gene and snpeff impact

gff *GFF3 file for example genomes*

Description

An example of GFF3 formatted genome information.

Usage

```
gff
```

Format

A character matrix with 110 rows and 9 columns:

Column 1 Chromosome

Column 2 Data source

Column 3 Feature type

Column 4 Feature start position

Column 5 Feature stop position

Column 6 Score

Column 7 Strand

Column 8 Phase

Column 9 Locus ID

`outgroup` *Name of outgroup in the phylogenetic tree.*

Description

Name of outgroup in the phylogenetic tree.

Usage

```
outgroup
```

Format

Character string.

`prewas` *Preprocess SNPs before bGWAS*

Description

prewas is a tool to standardize the pre-processing of your genomic data before performing a bacterial genome-wide association study (bGWAS). prewas creates a variant matrix (where each row is a variant, each column is a sample, and the entries are presence - 1 - or absence - 0 - of the variant) that can be used as input for bGWAS tools. When creating the binary variant matrix, prewas can perform 3 pre-processing steps including: dealing with multiallelic SNPs, (optional) dealing with SNPs in overlapping genes, and choosing a reference allele. prewas can output matrices for use with both SNP-based bGWAS and gene-based bGWAS. prewas can also provide gene matrices for variants with specific SnpEff annotations (Cingolani et al. 2012).

Usage

```
prewas(
  dna,
  tree = NULL,
  outgroup = NULL,
  gff = NULL,
  anc = TRUE,
  snpeff_grouping = NULL,
  grp_nonref = FALSE
)
```

Arguments

`dna`	‘Character‘ or ‘vcfR‘. Required input. Path to VCF4.1 file or ‘vcfR‘ object.
`tree`	‘NULL‘, ‘character‘, or ‘phylo‘. Optional input. Ignored if ‘NULL‘. If ‘character‘ it should be a path to a .tree file. Defaults to ‘NULL‘.
`outgroup`	‘NULL‘ or ‘character‘. Optional input. If ‘character‘ it should be either a string naming the outgroup in the tree or a path to a file containing only the outgroup name. Ignored if ‘NULL‘. Defaults to ‘NULL‘.
`gff`	‘NULL‘, ‘character‘, ‘matrix‘, or ‘data.frame‘. Optional input. If ‘NULL‘ it is ignored. If ‘character‘ it should be a path to a GFF3 file. If a ‘matrix‘ or ‘data.frame‘ it should be the GFF information stored in 9 columns with the genes as rows. Defaults to ‘NULL‘.
`anc`	‘Logical‘. Optional input. When ‘TRUE‘ prewas performs ancestral reconstruction. When ‘FALSE‘ prewas calculates the major allele. Defaults to ‘TRUE‘.
`snpeff_grouping`	‘NULL‘, ‘character‘. Optional input. Only used when a SnpEff annotated multivcf is inputted. Use when you want to group SNPs by gene and SnpEff impact. If ‘NULL‘ no custom-grouped gene matrix will be generated. Options for input are a vector combination of ’HIGH’, ’MODERATE’, LOW’, ’MODIFER’. Must write the impact combinations in all caps (e.g. c(’HIGH’, ’MODERATE’)). Defaults to ‘NULL‘.
`grp_nonref`	‘Logical‘. Optional input. When ‘TRUE‘ prewas collapses all non-reference alleles for multi-allelic sites. When ‘FALSE‘ prewas keeps multi-allelic sites separate. Defaults to ‘FALSE‘.

Value

A list with the following items:

allele_mat ‘matrix‘. An allele matrix, created from the vcf where each multiallelic site will be on its own line. The rowname will be the position of the variant in the vcf file. If the position is triallelic, there will be two rows containing the same information. The rows will be labeled "pos" and "pos.1". If the position is quadallelic, there will be three rows containing the same information. The rows will be labeled "pos", "pos.1", and "pos.2"

bin_mat ‘matrix‘. A binary matrix, where 0 is the reference allele and 1 indicates a variant. The dimensions may not match the ‘allele_mat‘ if the gff file is provided, because SNPs in overlapping genes are represented on multiple lines in ‘bin_mat‘; in that case both position and locus tag name are provided in the rowname.

ar_results ‘data.frame‘. This data.frame records the alleles used as the reference alleles. Rows correspond to variant loci. If ‘anc = TRUE‘ the data.frame has two columns which contain the ancestrally reconstructed allele and the probability of the reconstruction. If ‘anc = FALSE‘ there is only one column which contains the major allele.

dup ‘integer‘. A vector of integers. It’s an index that identifies duplicated rows. If the index is unique (appears once), that means it is not a multiallelic site. If the index appears more than once, that means the row was replicated ‘x‘ times, where ‘x‘ is the number of alternative alleles. Note: the multiple indices indicates multiallelic site splits, not overlapping genes splits.

gene_mat ‘NULL‘ or ‘matrix‘ or ‘list‘. ‘NULL‘ if no gene information provided (‘gff = NULL‘ and no SnpEff annotation provided in VCF). If gene information is provided by a gff then a gene matrix is generated where each row is a gene and each column is a sample. If gene information is provided by a SnpEff annotated vcf then a list of up to six gene matrices are returned. The first matrix, ‘gene_mat_all‘ is a gene matrix for all variants.‘gene_mat_modifier‘ is a gene matrix for only variants annoted as MODIFIER by SnpEff. Similarly there is a ‘gene_mat_low‘, ‘gene_mat_moderate‘, and ‘gene_mat_high.‘ If the user asks for a combination SnpEff annotations the final ‘gene_mat_custom‘ will contain that matrix.

tree ‘NULL‘ or ‘phylo‘. If ‘anc = FALSE‘ no tree is use or generated and the function returns ‘NULL‘. If ‘anc = TRUE‘ and the user provides a tree but no outgroup: the function returns the tree after midpoint rooting. If ‘anc = TRUE‘ and the user provides both a tree and an outgroup: the function returns a tree rooted on the outgroup and the outgroup is removed from the tree. If the user does not provide a tree and ‘anc = TRUE‘ the function returns the midpoint rooted tree generated.

Examples

```
vcf = prewas::vcf
gff = prewas::gff
tree = prewas::tree
outgroup = prewas::outgroup
output <- prewas(dna = vcf,
                 tree = tree,
                 outgroup = outgroup,
                 gff = gff,
                 anc = FALSE ,
                 grp_nonref = FALSE)
```

---

results *Results from running prewas() on the example data.*

Description

Output from prewas(). results <- prewas::prewas(dna = prewas::vcf, tree = prewas::tree, outgroup = prewas::outgroup, gff = prewas::gff, anc = FALSE)

Usage

```
results
```

Format

List of 5 objects.

allele_mat Matrix. Character matrix of nucleotides (alleles). Multiallelic sites represented on multiple lines in the matrix. Dim: 360 x 13. Rows are genomic loci. Columns are samples. Row names include only genomic position and do not have gene information.

bin_mat Matrix. Binary matrix (nucleotides stored as 0 or 1). Multiallelic sites represented on multiple lines in the matrix. Alleles in overlapping genes are represented on multiple lines in the matrix. Rownames include genomic position and gene. Dim: 1016 x 13. Rows are genomic loci. Columns are samples.

ar_results Data.frame. Dim: 360 x 1. Rows are genomic loci. The column is the major allele at that position. If anc=TRUE, then this object would be a 306 x 2 data.frame where the first column is the ancestral allele at that position inferred from ancestral reconstruction and the second column is the maximum likelihood probability.

dup Integer vector. Length = 360. The number refers to the original genomic loci in the VCF file. The occurrence count of the number is one less than the number of alleles. Ex: the 1st genomic locus (Position "1") occurs once in ‘dup‘ indicating that this is a biallelic site. In contrast, the 5th genomic locus in the vcf (Position 18) occurs twice indicating that this is a triallelic site (represented in two rows: 18 and 18.1)

gene_mat Matrix. Gene-based matrix. Genes with any SNP stored as 1, genes without SNPs stored as 0. Rows are genes. Columns are samples. Dim: 96 x 13.

`snpeff_vcf` *Nucleotide variants in example genome samples with snpeff annotations.*

Description

An example dataset containing 14 variants from 49 genome samples that has been annotated using snpeff.

Usage

```
snpeff_vcf
```

Format

vcfR class object with three sections:

meta The metadata for the VCF file including the file format version number

- **fix** A character matrix with 14 rows and 8 columns. Contains information on chromosome (CHROM), genome position (POS), reference genome allele (REF), and alternative allele (ALT). Information column (INFO) contains a field called "ANN" which provides the snpeff annotation including predicted functional impact of the variant on the protein function.
- **gt** A character matrix with 14 rows and 49 columns. Presence/absence for each variant defined in fix. Colnames are sample IDs.

`tree` *Phylogenetic tree of example genomes*

Description

Example unrooted phylogenetic tree.

Usage

```
tree
```

Format

An ape phylo object with 14 tips.

`vcf` *Nucleotide variants in example genome samples*

Description

An example dataset containing 326 variants from 14 genome samples.

Usage

```
vcf
```

Format

vcfR class object with three sections:

- **meta** The metadata for the VCF file including the file format version number
- **fix** A character matrix with 326 rows and 8 columns. Contains information on chromosome (CHROM), genome position (POS), reference genome allele (REF), and alternative allele (ALT)
- **gt** A character matrix with 326 rows and 15 columns. Presence/absence for each variant defined in fix. Colnames are sample IDs.

Index