Package ‘rCNV’

October 26, 2022

Type Package
Title Detect Copy Number Variants from SNPs Data
Version 1.1.0
Date 2022-10-26
Maintainer Piyal Karunarathne <piyalkarumail@yahoo.com>
Description Functions in this package will import filtered variant call format (VCF) files of SNPs data and generate data sets to detect copy number variants, visualize them and do downstream analyses with copy number variants (e.g. Environmental association analyses).
License AGPL (>= 3)
Imports data.table, graphics, colorspace, R.utils, qgraph, stringr
Encoding UTF-8
LazyData true
RoxygenNote 7.2.1
Depends R (>= 3.6.0)
Suggests rmarkdown, knitr, testthat (>= 3.0.0), covr
Config/testthat/edition 3
URL https://piyalkarum.github.io/rCNV/, https://cran.r-project.org/package=rCNV
BugReports https://github.com/piyalkarum/rCNV/issues
NeedsCompilation no
Author Piyal Karunarathne [aut, cre] (<https://orcid.org/0000-0002-1934-145X>), Qiujie Zhou [aut] (<https://orcid.org/0000-0001-7351-2371>), Klaus Schliep [aut] (<https://orcid.org/0000-0003-2941-0161>), Pascal Milesi [aut] (<https://orcid.org/0000-0001-8580-4291>)
Repository CRAN
Date/Publication 2022-10-26 16:12:37 UTC
### R topics documented:

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ad.correct</td>
<td>2</td>
</tr>
<tr>
<td>ADnorm</td>
<td>3</td>
</tr>
<tr>
<td>ADtable</td>
<td>4</td>
</tr>
<tr>
<td>allele.freq</td>
<td>4</td>
</tr>
<tr>
<td>allele.info</td>
<td>5</td>
</tr>
<tr>
<td>alleleINF</td>
<td>7</td>
</tr>
<tr>
<td>cnv</td>
<td>8</td>
</tr>
<tr>
<td>cpm.normal</td>
<td>9</td>
</tr>
<tr>
<td>depthVsSample</td>
<td>11</td>
</tr>
<tr>
<td>dup.plot</td>
<td>12</td>
</tr>
<tr>
<td>dup.validate</td>
<td>12</td>
</tr>
<tr>
<td>dupGet</td>
<td>13</td>
</tr>
<tr>
<td>exportVCF</td>
<td>15</td>
</tr>
<tr>
<td>get.miss</td>
<td>15</td>
</tr>
<tr>
<td>gt.format</td>
<td>16</td>
</tr>
<tr>
<td>h.zygosity</td>
<td>17</td>
</tr>
<tr>
<td>hetTgen</td>
<td>18</td>
</tr>
<tr>
<td>maf</td>
<td>19</td>
</tr>
<tr>
<td>norm.fact</td>
<td>20</td>
</tr>
<tr>
<td>power.bias</td>
<td>21</td>
</tr>
<tr>
<td>readVCF</td>
<td>22</td>
</tr>
<tr>
<td>relatedness</td>
<td>22</td>
</tr>
<tr>
<td>sig.hets</td>
<td>23</td>
</tr>
<tr>
<td>sim.als</td>
<td>24</td>
</tr>
<tr>
<td>vcf.stat</td>
<td>25</td>
</tr>
<tr>
<td>vst</td>
<td>26</td>
</tr>
</tbody>
</table>

### Index

28

---

#### ad.correct

**Correct allele depth values**

**Description**

A function to correct depth values with odd number of coverage values due to sequencing anomalies or miss classification where genotype is homozygous and depth values indicate heterozygosity. The function adds a value of one to the allele with the lowest depth value for when odd number anomalies or make the depth value zero for when miss-classified. The genotype table must be provided for the latter.

**Usage**

```r
ad.correct(het.table, gt.table = NULL, odd.correct = TRUE, verbose = TRUE)
```
Arguments

- `het.table`: allele depth table generated from the function `hetTgen`
- `gt.table`: genotype table generated from the function `hetTgen`
- `odd.correct`: logical, to correct for odd number anomalies in AD values. Default `TRUE`
- `verbose`: logical, show progress. Default `TRUE`

Value

Returns the coverage corrected allele depth table similar to the output of `hetTgen`

Author(s)

Piyal Karunarathne

Examples

```r
## Not run: adc<-ad.correct(ADtable)
```

Description

Normalized example SNPs data of Chinook Salmon from Larson et al. 2014. The data has been normalized with TMM

Usage

data(ADnorm)

Format

An object of class `list` of length 2.

References

ADtable

Allele Depth (AD) example data

Description

Example SNPs data of Chinook Salmon from Larson et al. et al. 2014. The data contains only a partial snps data set of RadSeq data after filtering.

Usage

data(ADtable)

Format

An object of class data.frame with 3000 rows and 109 columns.

References


allele.freq

Generate allele frequency table for individuals or populations

Description

Get alternative allele frequency across all individuals per SNP from the genotype or allele depth tables

Usage

allele.freq(gtt, f.typ = c("pop", "ind"), verbose = TRUE)

Arguments

gtt
a list or data frame of genotype and/or allele depth table produced from hetTgen (or similar)

f.typ
character. type of allele frequency to be calculated (individual "ind" or population "pop")

verbose
logical. whether to show the progress of the analysis
If the allele frequencies to be calculated for populations from both genotype table and the allele depth table, they must be provided in a list with element names AD for allele depth table and GT for the genotype table. See the examples.

Returns a data frame or a list (if both genotype and allele depth used) of allele frequencies

Piyal Karunarathne

vcf.file.path <- paste0(path.package("rCNV"), "/example.raw.vcf.gz")
vcf <- readVCF(vcf.file.path=vcf.file.path)
het.table<-hetTgen(vcf,"GT")
ad.table<-hetTgen(vcf,"AD")

# for individual based AF
frQ<--allele.freq(het.table,f.typ="ind")

# for population-wise and both allele depth and genotype tables
## Not run: frQ<--allele.freq(list(AD=ad.table,GT=het.table),f.typ="pop")

The function to calculate allele median ratios, proportion of heterozygotes and allele probability values under different assumptions (see details), and their chi-square significance values for duplicate detection

The function to calculate allele median ratios, proportion of heterozygotes and allele probability values under different assumptions (see details), and their chi-square significance values for duplicate detection

allele.info(  
X,  
x.norm = NULL,  
method = c("TMM", "TMMex", "MedR", "QN", "pca"),  
logratioTrim = 0.3,  
sumTrim = 0.05,  
Weighting = TRUE,  
A_cutoff = -1e+10,  
plot.allele.cov = TRUE,  
verbose = TRUE,
Arguments

\( X \)  
allele depth table generated from the function `hetTgen` (non-normalized)

\( x.norm \)  
a data frame of normalized allele coverage, output of `cpm.normal`. If not provided, calculated using \( X \).

\( \text{method} \)  
character. method to be used for normalization (see `cpm.normal` details). Default `TMM`

\( \text{logratioTrim} \)  
numeric. percentage value (0 - 1) of variation to be trimmed in log transformation

\( \text{sumTrim} \)  
numeric. amount of trim to use on the combined absolute levels (“A” values) for method `TMM`

\( \text{Weighting} \)  
logical, whether to compute (asymptotic binomial precision) weights

\( \text{Acutoff} \)  
numeric, cutoff on “A” values to use before trimming

\( \text{plot.allele.cov} \)  
logical, plot comparative plots of allele depth coverage in homozygotes and heterozygotes

\( \text{verbose} \)  
logical, whether to print progress

\( \ldots \)  
further arguments to be passed to `plot`

Details

Allele information generated here are individual SNP based and presents the proportion of heterozygotes, number of samples, and deviation of allele detection from a 1:1 ratio of reference and alternative alleles. The significance of the deviation is tested with Z-score test \( Z = \frac{\sqrt{2} - N_A}{\sigma_c} \), and chi-square test (see references for more details on the method).

Value

Returns a data frame of median allele ratio, proportion of heterozygotes, number of heterozygotes, and allele probability at different assumptions with their chi-square significance

Author(s)

Piyal Karunarathne, Pascal Milesi, Klaus Schliep

References

- Karunarathne et al. 2022 (to be added)
Examples

```r
## Not run: data(ADtable)
AI<-allele.info(ADtable,x.norm=ADnorm)
## End(Not run)
```

### alleleINF

<table>
<thead>
<tr>
<th>alleleINF</th>
<th>Allele info example data</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

#### Description

Semi-randomly generated data from the function dup.snp.info. Data contains depth and proportion values of 2857 snps.

#### Usage

```r
data(alleleINF)
```

#### Format

An object of class `data.frame` with 2857 rows and 28 columns.

#### Source

Chinook Salmon sequence reads McKinney et al. 2017

#### References


#### Examples

```r
data(alleleINF)
with(alleleINF,plot(medRatio~propHet))
```
Find CNVs from deviants

Description

Categorize deviant and non-deviant into "singlets" and "duplicates" based on the statistical approaches specified by the user. The intersection of all the stats provided will be used in the categorization. If one would like to use the intersection of at least two stats, this can be specified in the n.ints.

Usage

cnv(
  data,
  test = c("z.het", "z.05", "z.all", "chi.het", "chi.05", "chi.all"),
  filter = c("intersection", "kmeans"),
  ft.threshold = 0.05,
  plot = TRUE,
  verbose = TRUE,
  ...
)

Arguments

data A data frame of allele information generated with the function allele.info

test vector of characters. Type of test to be used for significance. See details

filter character. Type of filter to be used for filtering CNVs. default kmeans. See details.

ft.threshold confidence interval for filtering default = 0.05

plot logical. Plot the detection of duplicates. default TRUE

verbose logical. show progress

... other arguments to be passed to plot

Details

SNP deviants are detected with both excess of heterozygosity according to HWE and deviant SNPs where depth values fall outside of the normal distribution are detected using the following methods:

- **Z-score test** \( Z_x = \sum_{i=1}^{n} Z_i; Z_i = \frac{(N_i \times p - N_{Ai})}{\sqrt{N_i \times p \times (1-p)}} \)

- **chi-square test** \( X^2_x = \sum_{i=1}^{n} X_i^2; X_i^2 = \frac{(N_i \times p - N_{Ai})^2}{N_i \times p} + \frac{(N_i \times (1-p) - (N_i - N_{Ai}))^2}{N_i \times (1-p)} \)

See references for more details on the methods.

Users can pick among Z-score for heterozygotes (z.het, chi.het), all allele combinations (z.all, chi.all) and the assumption of no probe bias \( p=0.5 \) (z.05, chi.05).
filter will determine whether the intersection or kmeans clustering of the provided tests should be used in filtering CNVs. The intersection uses threshold values for filtering and kmeans use unsupervised clustering. Kmeans clustering is recommended if one is uncertain about the threshold values.

Value

Returns a data frame of SNPs with their detected duplication status

Author(s)

Piyal Karunarathne

Examples

## Not run: data(alleleINF)
DD<cnv(alleleINF)
## End(Not run)

---

`cpm.normal` *Calculate normalized depth for alleles*

**Description**

This function outputs the normalized depth values separately for each allele, calculated using normalization factor with trimmed mean of M-values of sample libraries, median ratios normalization or quantile normalization, See details.

**Usage**

```r
cpm.normal(
  het.table, 
  method = c("TMM", "TMMex", "MedR", "QN", "pca"),
  logratioTrim = 0.3,
  sumTrim = 0.05,
  Weighting = TRUE,
  Acutoff = -1e+10,
  verbose = TRUE,
  plot = TRUE
)
```

**Arguments**

- **het.table**: allele depth table generated from the function `hetTgen`
- **method**: character. method to be used (see details). Default `TMM`
- **logratioTrim**: numeric. percentage value (0 - 1) of variation to be trimmed in log transformation
**Details**

This function converts an observed depth value table to an effective depth value table using several normalization methods;

1. TMM normalization (See the original publication for more information). It is different from the function `normz` only in calculation of the counts per million is for separate alleles instead of the total depth. The TMMex method is an extension of the TMM method for large data sets containing SNPs exceeding 10000
2. The method MedR is median ratio normalization;
4. PCA - a modified Kaiser’s Rule applied to depth values: Sample variation of eigen values smaller than 0.7 are removed (i.e., the first eigen value < 0.7) to eliminate the effect of the library size of samples

**Value**

Returns a list with (AD), a data frame of normalized depth values similar to the output of `hetTgen` function and (outliers) a list of outlier sample names

**Author(s)**

Piyal Karunarathne, Qiujie Zhou

**References**


**Examples**

```r
## Not run: data(ADtable)
ADnormalized<-cpm.normal(ADtable)
## End(Not run)
```
depthVsSample

Simulate median allele ratios for varying number of samples and depth values

Description

This function will simulate the expected median allele ratios under HWE for given ranges of no. of samples and depth coverage values. This is useful if you need to find the cutoff values of allele ratios for different no. of samples and depth of coverage values in your data set.

Usage

```r
depthVsSample(
  cov.len = 100,
  sam.len = 100,
  nsims = 1000,
  plot = TRUE,
  col = c("#1C86EE", "#00BFFF", "#DAA520", "#FF0000")
)
```

Arguments

- `cov.len`: max value of depth of coverage to be simulated
- `sam.len`: maximum no. of samples to be simulated
- `nsims`: numerical. no. of simulations to be done for each combination of samples and depth depth and no. samples ranges
- `plot`: logical. Whether to plot the output (a plot of no. samples vs median depth of coverage colored by median allele ratios)
- `col`: character. Two colors to add to the gradient

Value

A matrix of median allele ratios where rows are the number of samples and columns are depth of coverage values

Author(s)

Pascal Milesi, Piyal Karunarathne

Examples

```r
## Not run: depthVsSample(cov.len=100,sam.len=100)
```
dup.plot

Plot classified SNPs into deviants/CNVs and non-deviants/non-CNVs

Description
The function plots detected deviants/CNVs from functions sig.snps, cnv and dupGet on a median ratio (MedRatio) Vs. proportion of heterozygote (PropHet) plot.

Usage

dup.plot(ds, ...)

Arguments

ds a data frame of detected deviants/cnvs (outputs of functions above)
...
other graphical parameters to be passed to the function plot

Value

Returns no value, only plots proportion of heterozygotes vs allele median ratio separated by duplication status

Author(s)

Piyal Karunarathne

Examples

## Not run: data(alleleINF)
DD<-dupGet(alleleINF,plot=FALSE)
dup.plot(DD)
## End(Not run)

dup.validate

Validate detected duplicates

Description

This function will validate the detected duplicated-SNPs using a moving window approach (see details)

Usage

dup.validate(d.detect, window.size = 100, scaf.size = 10000)
**Arguments**

- `d.detect`: a data frame of detected duplicates or deviants from the outputs of `dupGet` or `cnv` (output of `dupGet`)
- `window.size`: numerical. a single value of the desired moving window size (default 100 bp)
- `scaf.size`: numerical. scaffold size to be checked. i.e. the split size of the fragment to be checked with the specified window size. default=10000

**Details**

Chromosome positions correctly ordered according to a reference sequence is necessary for this function to work properly. Therefore, this function is still in development for non-mapped reference sequences.

**Value**

A data frame of scaffold names and their average presence in the scaffold.

**Author(s)**

Piyal Karunarathne

---

**Description**

Detect deviant SNPs using excess of heterozygotes (alleles that do not follow HWE) and allelic-ratio deviations (alleles with ratios that do not follow a normal Z-score or chi-square distribution). See details.

**Usage**

```r
dupGet(
  data,
  test = c("z.het", "z.05", "z.all", "chi.het", "chi.05", "chi.all"),
  intersection = FALSE,
  method = c("fisher", "chi.sq"),
  plot = TRUE,
  verbose = TRUE,
  ...
)
```
dupGet

**Arguments**

data  data frame of the output of `allele.info`
test  character. type of test to be used for significance. See details
intersection  logical, whether to use the intersection of the methods specified in `test` (if more than one)
method  character. method for testing excess of heterozygotes. Fisher exact test (`fisher`) or Chi-square test (`chi.sq`)
plot  logical. whether to plot the detected singlets and duplicates on allele ratio vs. proportion of heterozygotes plot.
verbose  logical. show progress
...  additional parameters passed on to `plot`

**Details**

SNP deviants are detected with both excess of heterozygosity according to HWE and deviant SNPs where depth values fall outside of the normal distribution are detected using the following methods:

- **Z-score test** $Z_x = \sum_{i=1}^{n} Z_i; Z_i = \frac{(N_i \times p - N_{Ai})}{\sqrt{N_i \times p (1-p)}}$

- **chi-square test** $X^2_x = \sum_{i=1}^{n} X_i^2; X^2_i = \left( \frac{(N_i \times p - N_{Ai})^2}{N_i \times p} + \frac{(N_i \times (1-p) - (N_i - N_{Ai}))^2}{N_i \times (1-p)} \right)$

See references for more details on the methods

Users can pick among Z-score for heterozygotes (`z.het`, `chi.het`), all allele combinations (`z.all`, `chi.all`) and the assumption of no probe bias $p=0.5$ (`z.05`, `chi.05`)

**Value**

Returns a data frame of snps/alleles with their duplication status

**Author(s)**

Piyal Karunarathne

**Examples**

```r
## Not run: data(alleleINF)
DD<-dupGet(alleleINF)
## End(Not run)
```
exportVCF  

Export VCF files

Description

A function to export tables/matrices in VCF format to VCF files

Usage

exportVCF(out.vcf, out.path, compress = TRUE)

Arguments

out.vcf  
a matrix or data frame in vcf file format to be exported

out.path  
a character string of output path for the vcf file; should end in the name as the vcf file and .vcf. See examples

compress  
logical. whether to compress the output file. If TRUE, the file will be .gz compressed

Value

Exports a vcf file to a given destination

Author(s)

Piyal Karunarathne

Examples

## Not run: vcf.file.path <- paste0(path.package("rCNV"), "/example.raw.vcf.gz")
vcf <- readVCF(vcf.file.path)
exportVCF(vcf,"../exVcf.vcf")
## End(Not run)

get.miss  
Get missingness of individuals in raw vcf

Description

A function to get the percentage of missing data of snps per SNP and per sample

Usage

get.miss(data, type = c("samples", "snps"), plot = TRUE, verbose = TRUE)
Arguments

- **data**
  a list containing imported vcf file using `readVCF` or genotype table generated using `hetTgen`

- **type**
  character. Missing percentages per sample “samples” or per SNP “snps”, default both

- **plot**
  logical. Whether to plot the missingness density with ninety five percent quantile

- **verbose**
  logical. Whether to show progress

Value

Returns a data frame of allele depth or genotypes

Author(s)

Piyal Karunaratne

Examples

```r
vcf.file.path <- paste0(path.package("rCNV"), "/example.raw.vcf.gz")
vcf <- readVCF(vcf.file.path=vcf.file.path)
missing<-get.miss(vcf,plot=TRUE)
```

---

**gt.format**

Format genotype for BayEnv and BayPass

Description

This function generates necessary genotype count formats for BayEnv and BayPass with a subset of SNPs

Usage

```r
gt.format(gt, info, format = c("benv", "bpass"), snp.subset = NULL)
```

Arguments

- **gt**
  multi-vector. an imported data.frame of genotypes or genotype data frame generated by `hetTgen` or path to GT.FORMAT file generated from VCFTools

- **info**
  a data frame containing sample and population information. It must have “sample” and “population” columns

- **format**
  character. output format i.e., for BayPass or BayEnv

- **snp.subset**
  numerical. number of randomly selected subsets of SNPs. default = NULL
Value

Returns a list with formatted genotype data: $bayenv - snps in horizontal format - for BayEnv (two lines per snp); $baypass - vertical format - for BayPass (two column per snp); $sub.bp - subsets snps for BayPass $sub.be - subsets of snps for BayEnv

Author(s)

Piyal Karunarathne

Examples

```r
## Not run: vcf.file.path <- paste0(path.package("rCNV"), ","/example.raw.vcf.gz")
vcf <- readVCF(vcf.file.path=vcf.file.path)
het.table<-hetTgen(vcf,"GT")
info<-unique(substr(colnames(het.table)[-c(1:3)],1,8))
GT<-gt.format(het.table,info)
## End(Not run)
```

---

**h.zygosity**

*Determines per sample heterozygosity and inbreeding coefficient*

Description

This function will calculate the heterozygosity on a per-sample basis from vcf files (snps), and most importantly inbreeding coefficient which is used to filter out the samples with bad mapping quality.

Usage

```r
h.zygosity(vcf, plot = FALSE, pops = NA, verbose = TRUE)
```

Arguments

- `vcf` an imported vcf file in in a list using `readVCF` or a data frame of genotypes generated using `hetTgen`
- `plot` logical. Whether to plot a boxplot of inbreeding coefficients for populations. A list of populations must be provided
- `pops` character. A list of population names with the same length and order as the number of samples in the vcf
- `verbose` logical. Show progress

Value

Returns a data frame of expected “E(Hom)” and observed “O(Hom)” homozygotes with their in-breeding coefficients.
hetTgen

Generate allele depth or genotype table

**Description**

hetTgen extracts the read depth and coverage values for each snp for all the individuals from a vcf file generated from readVCF (or GatK VariantsToTable: see details)

**Usage**

```r
hetTgen(
  vcf,
  info.type = c("AD", "AD-tot", "GT", "GT-012", "GT-AB", "DP"),
  verbose = TRUE
)
```

**Arguments**

- `vcf`: an imported vcf file in a list using `readVCF`
- `verbose`: logical. whether to show the progress of the analysis

**Details**

If you generate the depth values for allele by sample using GatK VariantsToTable option, use only `-F CHROM -F POS -GF AD` AD flags to generate the table. Or keep only the CHROM, POS, ID, ALT, and individual AD columns. For info.type GT option is provided to extract the genotypes of individuals by snp.

**Value**

Returns a data frame of allele depth, genotype of SNPs for all the individuals extracted from a VCF file."
maf

Author(s)

Piyal Karunarathne, Klaus Schliep

Examples

vcf.file.path <- paste0(path.package("rCNV"), "/example.raw.vcf.gz")
vcf <- readVCF(vcf.file.path=vcf.file.path)
het.table<-hetTgen(vcf)

maf
Remove MAF allele

Description

A function to remove the alleles with minimum allele frequency and keep only a bi-allelic matrix when loci are multi-allelic

Usage

maf(h.table, AD = TRUE, verbose = TRUE)

Arguments

h.table allele depth table generated from the function hetTgen
AD logical. If TRUE a allele depth table similar to hetTgen output will be returns; If FALSE, individual AD values per SNP will be returned in a list.
verbose logical. Show progress

Value

A data frame or a list of minimum allele frequency removed allele depth

Author(s)

Piyal Karunarathne

Examples

## Not run: mf<-maf(ADtable)
norm.fact

Calculate normalization factor for each sample

Description

This function calculates the normalization factor for each sample using different methods. See details.

Usage

```r
norm.fact(
  df,
  method = c("TMM", "TMMex", "MedR", "QN"),
  logratioTrim = 0.3,
  sumTrim = 0.05,
  Weighting = TRUE,
  Acutoff = -1e+10
)
```

Arguments

- **df**: a data frame or matrix of allele depth values (total depth per snp per sample)
- **method**: character. method to be used (see details). Default TMM
- **logratioTrim**: numeric. percentage value (0 - 1) of variation to be trimmed in log transformation
- **sumTrim**: numeric. amount of trim to use on the combined absolute levels (“A” values) for method TMM
- **Weighting**: logical, whether to compute (asymptotic binomial precision) weights
- **Acutoff**: numeric, cutoff on “A” values to use before trimming

Details

Originally described for normalization of RNA sequences (Robinson & Oshlack 2010), this function computes normalization (scaling) factors to convert observed library sizes into effective library sizes. It uses the method trimmed means of M-values proposed by Robinson & Oshlack (2010). See the original publication and edgeR package for more information. The method MedR is median ratio normalization; QN - quantile normalization (see Maza, Elie, et al. 2013 for a comparison of methods).

Value

Returns a numerical vector of normalization factors for each sample

Author(s)

Piyal Karunarathne
power.bias

References


Examples

```r
vcf.file.path <- paste0(path.package("rCNV"), ",example.raw.vcf.gz")
vcf <- readVCF(vcf.file.path)
df<-hetTgen(vcf,"AD-tot",verbose=FALSE)
norm.fact(df)
```

power.bias

Simulate and plot detection power of bias in allele ratios

Description

This function simulates 95% confidence level Z-score based detection power of allele biases for a given number of samples and a range of depths

Usage

```r
power.bias(
  Dlist = c(2, 4, 8, 16),
  sam = 100,
  intensity = 0.005,
  nsims = 1000,
  p = 0.5,
  plot = TRUE
)
```

Arguments

- **Dlist**: numerical. vector of depths values to be tested
- **sam**: numerical. number of samples
- **intensity**: numerical. frequency of bias
- **nsims**: numerical. number of simulations to be done for each sample
- **p**: numerical. expected allele ratio (0.5 for data with known sequencing biases)
- **plot**: logical. plot the output

Value

Returns a list of detection probability values for the given range of samples and depth
**readVCF**

*Import VCF file*

**Description**

Function to import raw single and multi-sample VCF files. The function required the R-package `data.table` for faster importing.

**Usage**

```r
readVCF(vcf.file.path, verbose = FALSE)
```

**Arguments**

- `vcf.file.path`: path to the vcf file
- `verbose`: logical. show progress

**Value**

Returns a list with vcf table in a data frame, excluding meta data.

**Examples**

```r
vcf.file.path <- paste0(path.package("rCNV"), "/example.raw.vcf.gz")
vcf <- readVCF(vcf.file.path)
```

---

**relatedness**

*Determine pairwise relatedness*

**Description**

Relatedness is determined according to genome-wide relationship assessment of Yang et al. 2010 equation 6, on a per sample basis (with itself and others), using SNPs.

**Usage**

```r
relatedness(vcf, plot = TRUE, threshold = 0.5, verbose = TRUE)
```
Arguments

vcf  an imported vcf file in a list using readVCF or a data frame of genotypes generated using hetTgen
plot logical. Whether to plot relatedness of samples against themselves, among themselves and outliers
threshold numerical. A value indicating to filter the individuals of relatedness among themselves. Default \(0.5\) (siblings)
verbose logical. Show progress.

Details

According to Yang et al. (2010), out breeding non-related pairs should have a relatedness value of zero while the individual with itself will have a relatedness value of one. Relatedness value of \(~0.5\) indicates siblings.

Value

A data frame of individuals and relatedness score \(A_{jk}\)

Author(s)

Piyal Karunarathne

References


Examples

```r
vcf.file.path <- paste0(path.package("rCNV"), "/example.raw.vcf.gz")
vcf <- readVCF(vcf.file.path=vcf.file.path)
relate<-relatedness(vcf)
```

---

**sig.hets**

Identify significantly different heterozygotes from SNPs data

Description

This function will recognize the SNPs with a proportion of heterozygotes significantly higher than expected under HWE and plot deviant snps based only on the excess of heterozygotes.
Usage

```r
sig.hets(
  a.info,
  method = c("chi.sq", "fisher"),
  plot = TRUE,
  verbose = TRUE,
  ...
)
```

Arguments

- **a.info**: allele info table generated from filtered vcfs using the function `allele.info` or allele depth table generated from `hetTgen`
- **method**: character. Method for testing significance. Fisher exact test (`fisher`) or Chi-square test (`chi.sq`)
- **plot**: logical. Whether to plot the identified duplicated snps with the expected values
- **verbose**: logical, if TRUE, the progress is shown
- **...**: other arguments passed to `plot`

Value

A matrix of expected heterozygote proportions from the observed data with p-value indicating significance of deviation.

Author(s)

Piyal Karunarathne, Pascal Milesi, Klaus Schliep

Examples

```r
## Not run: data(alleleINF)
AI <- alleleINF
duplicates<-sig.hets(AI,plot=TRUE)
## End(Not run)
```

---

**sim.als**

*Simulate Allele Frequencies*

**Description**

This function simulates allele frequencies of a desired population size under HWE

**Usage**

```r
sim.als(n = 500, nrun = 10000, res = 0.001, plot = TRUE)
```
Arguments

- `n`: desired populations size (set this value same as your actual population size for an accurate simulation)
- `nrun`: number of simulations to run on each allele frequency. The higher this number, the closer the simulations will be to the theoretical values (at the cost of computer power); 10000 is an optimal value.
- `res`: desired resolution of the theoretical allele frequency
- `plot`: logical. Whether to plot the simulation

Value

A list of two matrices:
1. `allele_freqs`: theoretical allele frequency
2. `simulated_freqs`: simulated frequencies at different confidence intervals

Author(s)

Piyal Karunarathne, Pascal Milesi

Examples

```r
## Not run: alleles <- sim.als(n=200,nrun=1000,res=0.001,plot=TRUE)
```

vcf.stat

Get sequencing quality statistics of raw VCF files (with GatK generated vcf files only)

Description

This function will generate a table similar to VariantsToTable option in GatK from raw vcf files for filtering purposes. The function will also plot all the parameters (see details & values).

Usage

```r
vcf.stat(vcf, plot = TRUE, ...)
```

Arguments

- `vcf`: an imported vcf file in data.frame or matrix format using `readVCF`
- `plot`: logical. Whether to plot the (12) parameters
- `...`: other arguments passed on to `plot` (e.g. `col`, `border`)

Details

For more details see instructions of GatK
Value

Returns a data frame with quality parameters from the INFO. field of the vcf

- QUAL: The Phred-scaled probability that a REF/ALT polymorphism exists at this site given sequencing data
- AC: Allele count
- AF: Allele frequency
- DP: unfiltered depth
- QD: QualByDepth - This is the variant confidence (from the QUAL field) divided by the unfiltered depth of non-hom-ref samples
- FS: FisherStrand - This is the Phred scaled probability that there is strand bias at the site
- SOR: StrandOddsRatio - This is another way to estimate strand bias using a test similar to the symmetric odds ratio test
- MQ: RMSMappingQuality - This is the root mean square mapping quality over all the reads at the site
- MQRankSum: MappingQualityRankSumTest - This is the u-based z-approximation from the Rank Sum Test for mapping qualities
- ReadPosRankSum: ReadPosRankSumTest: This is the u-based z-approximation from the Rank Sum Test for site position within reads

Author(s)

Piyal Karunaratne

Examples

```r
vcf.file.path <- paste0(path.package("rCNV"), "/example.raw.vcf.gz")
vcf <- readVCF(vcf.file.path=vcf.file.path)
statistics<-vcf.stat(vcf,plot=TRUE)
```

vst Calculate population-wise Vst

Description

This function calculates Vst (variant fixation index) for populations given a list of duplicated loci

Usage

```r
vst(AD, pops, id.list = NULL, qGraph = TRUE, verbose = TRUE, ...)
```
Arguments

AD: data frame of total allele depth values of (duplicated, if id.list is not provided) SNPs

pops: character. A vector of population names for each individual. Must be the same length as the number of samples in AD

id.list: character. A vector of duplicated SNP IDs. Must match the IDs in the AD data frame

qGraph: logical. Plot the network plot based on Vst values (see details)

verbose: logical. show progress

... additional arguments passed to qgraph

Details

Vst is calculated with the following equation

\[ V_T = \frac{V_S}{V_T} \]

where \( V_T \) is the variance of normalized read depths among all individuals from the two populations and \( V_S \) is the average of the variance within each population, weighed for population size (see reference for more details) See qgraph help for details on qgraph output

Value

Returns a matrix of pairwise Vst values for populations

Author(s)

Piyal Karunarathne

References


Examples

```r
## Not run: data(alleleINF)
data(ADtable)
DD<-dupGet(alleleINF)
ds<-DD[DD$dup.stat=="duplicated",]
ad<-ADtable[match(paste0(ds$CHROM,".",ds$POS),paste0(ADtable$CHROM,".",ADtable$POS)),]
vst(ad,pops=substr(colnames(ad)[-c(1:4)],1,11))
## End(Not run)
```
Index

* datasets
  ADnorm, 3
  ADtable, 4
  alleleINF, 7
ad.correct, 2
ADnorm, 3
ADtable, 4
allele.freq, 4
allele.info, 5
alleleINF, 7
cnv, 8
cpm.normal, 9
depthVsSample, 11
dup.plot, 12
dup.validate, 12
dupGet, 13
exportVCF, 15
get.miss, 15
gt.format, 16
h.zygosity, 17
hetTgen, 18
maf, 19
norm.fact, 20
power.bias, 21
readVCF, 22
relatedness, 22
sig.hets, 23
sim.als, 24
vcf.stat, 25
vst, 26