Package ‘rCNV’

April 6, 2022

Type Package

Title Detect Copy Number Variants from SNPs Data

Version 1.0.0

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

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

Depends R (>= 3.6.0)

Suggests rmarkdown, knitr, testthat (>= 3.0.0), covr

Config/testthat/edition 3

URL https://piyalkarum.github.io/rCNV/

BugReports https://github.com/piyalkarum/rCNV/issues

NeedsCompilation no

Author Piyal Karunarathne [aut, cre] (<https://orcid.org/0000-0002-1934-145X>), Pascal Milesi [aut] (<https://orcid.org/0000-0001-8580-4291>), Qiujie Zhou [aut] (<https://orcid.org/0000-0001-7351-2371>)

Repository CRAN

Date/Publication 2022-04-06 07:50:02 UTC

R topics documented:

   ad.correct ........................................................................ 2
   ADnorm ........................................................................ 3
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

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

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>het.table</td>
<td>allele depth table generated from the function hetTgen</td>
</tr>
<tr>
<td>gt.table</td>
<td>genotype table generated from the function hetTgen</td>
</tr>
<tr>
<td>odd.correct</td>
<td>logical, to correct for odd number anomalies in AD values. default TRUE</td>
</tr>
<tr>
<td>verbose</td>
<td>logical. show progress. Default TRUE</td>
</tr>
</tbody>
</table>
ADnorm

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)
```

---

**ADnorm**  
*Normalized allele depth example data*

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 (from genotypes)

**Description**

Get alternative allele frequency across all individuals per SNP from the genotype table

**Usage**

```r
allele.freq(gtt, verbose = TRUE)
```

**Arguments**

- `gtt` a genotype table produced from `hetTgen` (or similar)
- `verbose` logical. whether to show the progress of the analysis

**Details**

Use `hetTgen` function to generate the genotype table with the `GT` option

**Value**

Returns a data frame of allele frequencies calculated from genotypes

**Author(s)**

Piyal Karunarathne

**Examples**

```r
vcf.file.path <- paste0(path.package("rCNV"), "/example.raw.vcf.gz")
vcf <- readVCF(vcf.file.path=vcf.file.path)
het.table<-hetTgen(vcf,"GT")
frQ<-allele.freq(het.table)
```
Get allele information for duplicate detection

Description

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

Usage

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

Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>allele depth table generated from the function hetTgen (non-normalized)</td>
</tr>
<tr>
<td>x.norm</td>
<td>a data frame of normalized allele coverage, output of cpm.normal. If not provided, calculated using X.</td>
</tr>
<tr>
<td>method</td>
<td>character. method to be used for normalization (see cpm.normal details). Default TMM</td>
</tr>
<tr>
<td>logratioTrim</td>
<td>numeric. percentage value (0 - 1) of variation to be trimmed in log transformation</td>
</tr>
<tr>
<td>sumTrim</td>
<td>numeric. amount of trim to use on the combined absolute levels (“A” values) for method TMM</td>
</tr>
<tr>
<td>Weighting</td>
<td>logical, whether to compute (asymptotic binomial precision) weights</td>
</tr>
<tr>
<td>Acutoff</td>
<td>numeric, cutoff on “A” values to use before trimming</td>
</tr>
<tr>
<td>plot.allele.cov</td>
<td>logical, plot comparative plots of allele depth coverage in homozygotes and heterozygotes</td>
</tr>
<tr>
<td>verbose</td>
<td>logical, whether to print progress</td>
</tr>
<tr>
<td>...</td>
<td>further arguments to be passed to plot</td>
</tr>
</tbody>
</table>
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 $Z = \frac{\tilde{X} - \tilde{N}_A}{\sigma_x}$, 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, Qiujie Zhou

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

Allele info example data

**Description**

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

**Usage**

data(alleleINF)

**Format**

An object of class data.frame with 2857 rows and 25 columns.

**Source**

Chinook Salmon sequence reads McKinney et al. 2017
References


Examples

data(alleleINF)
with(alleleINF, plot(medRatio ~ propHet))

**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**

cpm.normal(
  het.table,  # allele depth table generated from the function hetTgen
  method = c("TMM", "TMMex", "MedR", "QN"),  # character. method to be used (see details). Default TMM
  logratioTrim = 0.3,  # numeric. percentage value (0 - 1) of variation to be trimmed in log transformation
  sumTrim = 0.05,  # numeric. amount of trim to use on the combined absolute levels ("A" values) for method TMM
  Weighting = TRUE,  # logical, whether to compute (asymptotic binomial precision) weights
  Acutoff = -1e+10,  # numeric, cutoff on "A" values to use before trimming (only for TMM(ex))
  verbose = TRUE  # logical. show progress
)

**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
- **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 (only for TMM(ex))
- **verbose**: logical. show progress
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;

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 no. of samples and depth coverage

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.
**dup.plot**

**Usage**

```r
depthVsSample(
  cov.len = 400,
  sam.len = 1000,
  incr = c(1, 1),
  plot = TRUE,
  plot.cols = c("red", "cyan")
)
```

**Arguments**

- `cov.len`: max value of depth of coverage to be simulated
- `sam.len`: maximum no. of samples to be simulated
- `incr`: a vector of two integers indicating increment size for both depth and no. samples
- `plot`: logical. Whether to plot the output (a plot of no. samples vs median depth of coverage colored by median allele ratios)
- `plot.cols`: 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=50,sam.len=100)
```

---

**dup.plot**

**Plot duplicates**

**Description**

The function plots detected duplicates from functions `sig.snps`, and `dupGet`

**Usage**

```r
dup.plot(ds, ...)
```
**dup.validate**

**Validate detected duplicates**

**Description**

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

**Usage**

```r
dup.validate(d.detect, window.size = 100)
```

**Arguments**

- `d.detect`: a data frame of detected SNPs of duplicates and singlets (output of `dupGet`)
- `window.size`: numerical. a single value of the desired moving window size (default 100 bp)

**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.
dupGet

Author(s)

Piyal Karunarathne

---

dupGet  

Detect duplicates from SNPs

Description

Detect duplicated snps using excess of heterozygotes (alleles that do not follow HWE) and snp deviates (alleles that do not follow a normal or chi-square distribution). See details.

Usage

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,  
  ...  
)

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

Duplicates are detected with both excess of heterozygosity according to HWE and deviant SNPs where deviants are detected using the following methods:

1. Z-score test  
   \[ Z = \frac{\sqrt{N} - N_A}{\sigma_x} \]

2. chi-square test (see references for more details on the method)

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)
**Description**

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

**Usage**

```r
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**

```r
## 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**

```r
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, snp.subset = FALSE, verbose = FALSE)
```
h.zygosity

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.
- **snp subset**: logical. Whether to generate a randomly sampled tenfold subset.
- **verbose**: logical. If TRUE shows progress.

Value

Returns a list with formatted genotype data:
- **$hor**: snps in horizontal format (two lines per snp);
- **$ver**: vertical format (two column per snp);
- **$hor.chunk**: a subset snps of **$hor**

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**  
*Determine 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.
hetTgen

Value

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

Author(s)

Piyal Karunarathne, Pascal Milesi

Examples

```r
## Not run: vcf.file.path <- paste0(path.package("rCNV"), "/example.raw.vcf.gz")
vcf <- readVCF(vcf.file.path=vcf.file.path)
pp<-substr(colnames(vcf$vcf)[-c(1:9)],1,8)
hzygots<-h.zygosity(vcf,plot=TRUE,pops=pp)
## End(Not run)
```

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 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, Genotyp of SNPs for all the individuals extracted from a VCF file

Author(s)
Piyal Karunarathne

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

mf<-maf(het.table)
```

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

norm.fact(
  df,
  method = c("TMM", "TMMex"),
  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.

Value

Returns a numerical vector of normalization factors for each sample

Author(s)

Piyal Karunarathne
readVCF

Description

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

Usage

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.

Author(s)

Piyal Karunarathne

Examples

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)
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

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

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 putatively duplicated snps

Usage

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

Arguments

- **a.info**: allele info table generated from filtered vcf's using the function allele.info
- **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

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

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
### 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, ...)
```

### 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).
- **...**: additional arguments passed to `qgraph`.

### Details

Vst is calculated with the following equation:

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

where VT is the variance of normalized read depths among all individuals from the two populations and VS is the average of the variance within each population, weighted 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

**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)
```
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, 3
  alleleINF, 6
ad.correct, 2
ADnorm, 3
ADtable, 3
allele.freq, 4
allele.info, 5
alleleINF, 6
cpm.normal, 7
depthVsSample, 8
dup.plot, 9
dup.validate, 10
dupGet, 11
exportVCF, 12
get.miss, 13
gt.format, 13
h.zygosity, 14
hetTgen, 15
maf, 16
norm.fact, 17
readVCF, 18
relatedness, 19
sig.hets, 20
sim.als, 21
vcf.stat, 22
vst, 23