

Package ‘HaploDiploidEquilibrium’

May 7, 2026

Type Package

Title Calculate F Statistics Using Mixed Haploid and Diploid Organism
Data

Version 0.1.0

Description Provides functions to estimate population genetics summary statistics from haplo-diploid systems, where one sex is haploid and the other diploid (e.g. Hymenoptera insects). It implements a theoretical model assuming equal sex ratio, random mating, no selection, no mutation, and no gene flow, deriving expected genotype frequencies for both sexes under these equilibrium conditions. The package includes windowed calculations (operating over genomic sliding windows from VCF input) for allele and genotype frequencies, the inbreeding coefficient (Fis), pairwise Fst, Nei's H (gene diversity), Watterson's Theta, and sex-specific reference allele frequencies. Most statistics are agnostic to ploidy, allowing the package to be applied to both strictly haplo-diploid and fully diploid systems.

License GPL-3

Encoding UTF-8

Depends R (>= 4.1.0)

Imports dplyr, vcfR, data.table, matrixStats

RoxygenNote 7.3.3

NeedsCompilation no

Author Paulo Sousa [aut],
Francisco Pina-Martins [cre, aut]

Maintainer Francisco Pina-Martins <f.pinamartins@gmail.com>

Repository CRAN

Date/Publication 2026-02-25 19:30:12 UTC

Contents

allele.freq.WS	2
compute.Female.Male.allele.W	3
compute_allele.freqs_W	5
compute_Hs_W	8

pairwise.fst	9
summarize_fst	11
summarize_geno	12
summarize_NeisH	13
summarize_sex_ref	14
vcf2GT	15

Index	17
--------------	-----------

allele.freq.WS	<i>Compute per-window reference allele frequencies across populations</i>
----------------	---

Description

Iterates over each population defined in `pop.file`, splits the genotype data by contig, and slides a fixed-size window along each contig to compute the reference allele frequency within that window. Both diploid genotypes ("`0/0`", "`0/1`", "`1/1`") and haploid genotypes ("`0`", "`1`") are recognised. Allele frequency is agnostic to ploidy. The resulting per-window frequencies are the direct input expected by [`pairwise.fst()`].

Usage

```
allele.freq.WS(
  geno.data,
  pop.file,
  contigs,
  positions,
  window.size,
  verbose = TRUE
)
```

Arguments

<code>geno.data</code>	A character matrix of genotype strings with dimensions <code>n_sites</code> x <code>n_individuals</code> , as returned by [<code>vcf2GT()</code>].
<code>pop.file</code>	A <code>data.frame</code> or <code>data.table</code> with at least two columns: <code>ID</code> (individual identifiers matching the column names of <code>geno.data</code>) and <code>Pop</code> (population labels).
<code>contigs</code>	A character vector of length <code>n_sites</code> containing the contig (chromosome) name for each variant site, as returned by [<code>vcf2GT()</code>].
<code>positions</code>	A numeric vector of length <code>n_sites</code> containing the physical position (bp) of each variant site, as returned by [<code>vcf2GT()</code>].
<code>window.size</code>	A single positive integer giving the size of each sliding window in base pairs.
<code>verbose</code>	Logical. If 'TRUE' (default), print progress messages.

Value

A [data.table::data.table] with one row per population-contig-window combination and the following columns:

Pop Population label.

Contig Contig (chromosome) name.

Window_starts Genomic coordinate (bp) of the first position in the window.

Window_ends Genomic coordinate (bp) of the last position in the window (`Window_starts + window.size - 1`).

N_sites Total number of called genotype entries (diploid + haploid) within the window.

Freq.A Frequency of the reference allele in the window, computed as $(2*N_{AA} + N_{Aa} + N_A) / (2*N_{dip} + N_{hap})$.

See Also

[pairwise.fst()] for computing Fst from the output of this function.

Examples

```
vcf_path <- system.file("extdata",
                        "example.vcf",
                        package = "HaploDiploidEquilibrium")
result <- vcf2GT(vcf_path)
gt <- result$gt_matrix
contigs <- result$contig_vector
pos <- result$positions

pop.file <- data.frame(ID = colnames(gt),
                      Pop = c("PopA", "PopA", "PopB", "PopB", "PopB"))

af <- allele.freq.WS(geno.data = gt,
                    pop.file = pop.file,
                    contigs = contigs,
                    positions = pos,
                    window.size = 10000)
```

```
compute.Female.Male.allele.W
```

Compute per-window reference allele frequencies by sex

Description

Iterates over each population defined in `pop.file`, splits the genotype data by contig, and slides a fixed-size window along each contig to compute the reference allele frequency separately for diploid individuals (females), haploid individuals (males), and both sexes combined. Sex is inferred from ploidy: diploid genotypes ("0/0", "0/1", "1/1") are assumed to belong to females and haploid genotypes ("0", "1") to males.

Usage

```
compute.Female.Male.allele.W(
  geno.data,
  pop.file,
  contigs,
  positions,
  window.size,
  verbose = TRUE
)
```

Arguments

<code>geno.data</code>	A character matrix of genotype strings with dimensions <code>n_sites</code> x <code>n_individuals</code> , as returned by <code>[vcf2GT()]</code> .
<code>pop.file</code>	A <code>data.frame</code> or <code>data.table</code> with at least two columns: ID (individual identifiers matching the column names of <code>geno.data</code>) and Pop (population labels).
<code>contigs</code>	A character vector of length <code>n_sites</code> containing the contig (chromosome) name for each variant site, as returned by <code>[vcf2GT()]</code> .
<code>positions</code>	A numeric vector of length <code>n_sites</code> containing the physical position (bp) of each variant site, as returned by <code>[vcf2GT()]</code> .
<code>window.size</code>	A single positive integer giving the size of each sliding window in base pairs.
<code>verbose</code>	Logical. If 'TRUE' (default), print progress messages.

Value

A [`data.table::data.table`] with one row per population-contig-window combination and the following columns:

Pop Population label.

Contig Contig (chromosome) name.

Window_starts Genomic coordinate (bp) of the first position in the window.

Window_ends Genomic coordinate (bp) of the last position in the window (`Window_starts + window.size - 1`).

N_sites Total number of called genotype entries (diploid + haploid) within the window.

Females.freq Reference allele frequency computed from diploid genotypes only: $(2*N_{AA} + N_{Aa}) / (2*N_{dip})$.

Males.freq Reference allele frequency computed from haploid genotypes only: N_A / N_{hap} .

Total.freq Reference allele frequency computed from both sexes combined: $(2*N_{AA} + N_{Aa} + N_A) / (2*N_{dip} + N_{hap})$.

See Also

`[summarize_sex_ref()]` for computing weighted genome-wide summary statistics from the output.

Examples

```
vcf_path <- system.file("extdata",
                        "example.vcf",
                        package = "HaploDiploidEquilibrium")
result <- vcf2GT(vcf_path)
gt      <- result$gt_matrix
contigs <- result$contig_vector
pos     <- result$positions

pop.file <- data.frame(ID = colnames(gt),
                       Pop = c("PopA", "PopA", "PopB", "PopB", "PopB"))

sex_ref <- compute.Female.Male.allele.W(geno.data = gt,
                                         pop.file  = pop.file,
                                         contigs    = contigs,
                                         positions  = pos,
                                         window.size = 10000)
```

```
compute_allele.freqs_W
```

Compute per-window genotype frequencies, allele frequencies, and Fis

Description

Iterates over each population defined in pop.file, splits the genotype data by contig, and slides a fixed-size window along each contig to compute observed and expected genotype frequencies, allele frequencies, and the inbreeding coefficient (Fis). Expected genotype frequencies are derived from the haplo-diploid equilibrium model, where the proportion of diploid and haploid individuals in the population is controlled by dip_freq. Sex is inferred from ploidy: diploid genotypes ("0/0", "0/1", "1/1") are assumed to belong to females and haploid genotypes ("0", "1") to males. Fis is computed as $1 - (\text{Obs.Het} / \text{Exp.Het})$ and is set to NA when expected heterozygosity is zero.

Usage

```
compute_allele.freqs_W(
  geno.data,
  pop.file,
  contigs,
  positions,
  window.size,
  dip_freq,
  verbose = TRUE
)
```

Arguments

<code>geno.data</code>	A character matrix of genotype strings with dimensions <code>n_sites</code> x <code>n_individuals</code> , as returned by <code>[vcf2GT()]</code> .
<code>pop.file</code>	A <code>data.frame</code> or <code>data.table</code> with at least two columns: <code>ID</code> (individual identifiers matching the column names of <code>geno.data</code>) and <code>Pop</code> (population labels).
<code>contigs</code>	A character vector of length <code>n_sites</code> containing the contig (chromosome) name for each variant site, as returned by <code>[vcf2GT()]</code> .
<code>positions</code>	A numeric vector of length <code>n_sites</code> containing the physical position (bp) of each variant site, as returned by <code>[vcf2GT()]</code> .
<code>window.size</code>	A single positive integer giving the size of each sliding window in base pairs.
<code>dip_freq</code>	A single numeric value in the interval $(0, 1)$ giving the expected proportion of diploid individuals in the population. The haploid proportion is set to $1 - \text{dip_freq}$. A value of 0.5 corresponds to an equal sex ratio and is recommended for standard haplo-diploid systems.
<code>verbose</code>	Logical. If 'TRUE' (default), print progress messages.

Details

Haplo-diploid equilibrium model assumes a equal sex-ratio for three main reasons: 1: A sex-ratio different from 1:1 (e.g. 1:2) would break the assumptions of equal probability of contributing to next generation (selection and even drift) 2: Population sex-ratio is not sample sex-ratio. Sample sex-ratio might be different from population's sex-ratio due to time of the year when sampling took place, flower resources, etc. 3: Sex-ratio might be different from didderent populations, across time and evolving. So its estimation for one population might not hold for another or the same in the next breeding season

For the above mentioned reasons, we highly recomend that the sex-ratio is left has 0.5 (1:1) unless strong evidence existis of otherwise. A true sex-ratio different from 1:1 (assumed) should also be considered to explain the possible differences between observed and expected genotype frequencies. In other words, sex-ratio different from 1:1 should change the genotype frequencies but not the allele frequencies.

Fis might not be reliable when the number of sampled diploids is to low

Value

A `[data.table::data.table]` with one row per population-contig-window combination and the following columns:

Pop Population label.

Contig Contig (chromosome) name.

Window_starts Genomic coordinate (bp) of the first position in the window.

Window_ends Genomic coordinate (bp) of the last position in the window (`Window_starts + window.size - 1`).

N_sites Total number of called genotype entries (diploid + haploid) within the window.

N_F Number of diploid (female) genotype entries in the window.

N_M Number of haploid (male) genotype entries in the window.

- N_AA** Count of homozygous reference (AA) genotypes.
- N_Aa** Count of heterozygous (Aa) genotypes.
- N_aa** Count of homozygous alternative (aa) genotypes.
- N_A** Count of haploid reference (A) genotypes.
- N_a** Count of haploid alternative (a) genotypes.
- Freq.Ref** Overall reference allele frequency in the window.
- Freq.Alt** Overall alternative allele frequency in the window.
- Obs.Hom** Observed proportion of homozygous diploid genotypes (AA + aa) relative to total entries.
- Obs.Het** Observed proportion of heterozygous diploid genotypes (Aa) relative to total entries.
- Obs.M.Ref** Observed proportion of haploid reference genotypes (A) relative to total entries.
- Exp.Hom** Expected frequency of homozygous diploid genotypes under the haplo-diploid equilibrium model.
- Exp.Het** Expected frequency of heterozygous diploid genotypes under the haplo-diploid equilibrium model.
- Exp.M.Ref** Expected frequency of haploid reference genotypes under the haplo-diploid equilibrium model.
- Fis** Inbreeding coefficient for the window, or NA when expected heterozygosity is zero.

See Also

[summarize_geno()] for computing weighted genome-wide summary statistics from the output.

Examples

```
vcf_path <- system.file("extdata",
                        "example.vcf",
                        package = "HaploDiploidEquilibrium")

result <- vcf2GT(vcf_path)
gt      <- result$gt_matrix
contigs <- result$contig_vector
pos     <- result$positions

pop.file <- data.frame(ID = colnames(gt),
                      Pop = c("PopA", "PopA", "PopB", "PopB", "PopB"))

geno <- compute_allele.freqs_W(geno.data = gt,
                              pop.file  = pop.file,
                              contigs   = contigs,
                              positions = pos,
                              window.size = 10000,
                              dip_freq  = 0.5)
```

 compute_Hs_W

 Compute per-window Nei's *H* (gene diversity)

Description

Iterates over each population defined in `pop.file`, splits the genotype data by contig, and slides a fixed-size window along each contig to compute Nei's *H* (probability of sampling two different alleles) within that window. Nei's *H* is calculated as $2pq$, where *p* and *q* are the reference and alternative allele frequencies respectively. Both diploid genotypes ("*0/0*", "*0/1*", "*1/1*") and haploid genotypes ("*0*", "*1*") are recognised when computing allele frequencies. Despite the different ploidy, allele frequencies should be the same between sexes, which means that Nei's *H* is agnostic to ploidy.

Usage

```
compute_Hs_W(
  geno.data,
  pop.file,
  contigs,
  positions,
  window.size,
  verbose = TRUE
)
```

Arguments

<code>geno.data</code>	A character matrix of genotype strings with dimensions <code>n_sites</code> x <code>n_individuals</code> , as returned by <code>[vcf2GT()]</code> .
<code>pop.file</code>	A <code>data.frame</code> or <code>data.table</code> with at least two columns: ID (individual identifiers matching the column names of <code>geno.data</code>) and Pop (population labels).
<code>contigs</code>	A character vector of length <code>n_sites</code> containing the contig (chromosome) name for each variant site, as returned by <code>[vcf2GT()]</code> .
<code>positions</code>	A numeric vector of length <code>n_sites</code> containing the physical position (bp) of each variant site, as returned by <code>[vcf2GT()]</code> .
<code>window.size</code>	A single positive integer giving the size of each sliding window in base pairs
<code>verbose</code>	Logical. If 'TRUE' (default), print progress messages.

Value

A `[data.table::data.table]` with one row per population-contig-window combination and the following columns:

Pop Population label.

Contig Contig (chromosome) name.

Window_starts Genomic coordinate (bp) of the first position in the window.

Window_ends Genomic coordinate (bp) of the last position in the window (`Window_starts + window.size - 1`).

N_sites Total number of called genotype entries (diploid + haploid) within the window.

Neis_H Nei's H (gene diversity) for the window, computed as $2 * \text{Freq.Ref} * \text{Freq.Alt}$.

See Also

[`summarize_NeisH()`] for computing weighted genome-wide summary statistics from the output.

Examples

```
vcf_path <- system.file("extdata",
                        "example.vcf",
                        package = "HaploDiploidEquilibrium")

result <- vcf2GT(vcf_path)
gt      <- result$gt_matrix
contigs <- result$contig_vector
pos     <- result$positions

pop.file <- data.frame(ID = colnames(gt),
                      Pop = c("PopA", "PopA", "PopB", "PopB", "PopB"))

hs <- compute_Hs_W(geno.data = gt,
                  pop.file   = pop.file,
                  contigs    = contigs,
                  positions  = pos,
                  window.size = 10000)
```

pairwise.fst

Compute pairwise Fst for all population pairs

Description

Takes the per-window allele frequency table produced by [`allele.freq.WS()`] and computes Wright's Fst for every unique pair of populations. Within each contig-window, Fst is estimated as the ratio of the among-population variance in reference allele frequency to its expected maximum under panmixia: $F_{st} = \text{Var}(p) / (\text{mean}(p) * (1 - \text{mean}(p)))$. Windows in which the mean reference allele frequency is 0 or 1 (i.e. monomorphic across the pair) are set to 0.

Usage

```
pairwise.fst(allele.freq.table, verbose = TRUE)
```

Arguments

allele.freq.table A [data.table::data.table] produced by [allele.freq.WS()], containing at minimum the columns Pop, Contig, Window_starts, Window_ends, N_sites, and Freq.A.

verbose Logical. If 'TRUE' (default), print progress messages.

Value

A [data.table::data.table] with one row per population-pair-contig- window combination and the following columns:

Contig Contig (chromosome) name.

window_lims A character string of the form "Window_starts - Window_ends" identifying the window.

Sum.Sites Total number of called genotype entries summed across both populations in the window.

Mean.p Mean reference allele frequency across the two populations in the window.

Var.p Population variance of the reference allele frequency across the two populations in the window.

Fst Estimated Fst for the window.

Pop_pair A character string of the form "Pop1 - Pop2" identifying the population pair.

See Also

[allele.freq.WS()] for computing the input allele frequency table, and [summarize_fst()] for computing weighted genome-wide summary statistics from the output.

Examples

```
vcf_path <- system.file("extdata",
                        "example.vcf",
                        package = "HaploDiploidEquilibrium")
result <- vcf2GT(vcf_path)
gt <- result$gt_matrix
contigs <- result$contig_vector
pos <- result$positions

pop.file <- data.frame(ID = colnames(gt),
                      Pop = c("PopA", "PopA", "PopB", "PopB", "PopB"))

af <- allele.freq.WS(geno.data = gt,
                    pop.file = pop.file,
                    contigs = contigs,
                    positions = pos,
                    window.size = 10000)
fst <- pairwise.fst(af)
```

summarize_fst	<i>Summarize genome-wide Fst per population pair</i>
---------------	--

Description

Computes the site-count-weighted mean and standard deviation of Fst across all windows for each population pair, using the per-window Fst table produced by [pairwise.fst()]. Weighting by Sum.Sites ensures that windows with more called genotypes contribute more to the estimate.

Usage

```
summarize_fst(fst_table)
```

Arguments

fst_table A [data.table::data.table] produced by [pairwise.fst()], containing at minimum the columns Pop_pair, Fst, and Sum.Sites.

Value

A [tibble::tibble] with one row per population pair and the following columns:

Pop_pair A character string identifying the population pair.

wMean.Fst Weighted mean of Fst across all windows.

wSD.Fst Weighted standard deviation of Fst across all windows.

See Also

[pairwise.fst()] for computing the input per-window Fst table.

Examples

```
vcf_path <- system.file("extdata",
                        "example.vcf",
                        package = "HaploDiploidEquilibrium")
result <- vcf2GT(vcf_path)
gt <- result$gt_matrix
contigs <- result$contig_vector
pos <- result$positions

pop.file <- data.frame(ID = colnames(gt),
                      Pop = c("PopA", "PopA", "PopB", "PopB", "PopB"))

af <- allele.freq.WS(geno.data = gt,
                    pop.file = pop.file,
                    contigs = contigs,
                    positions = pos,
                    window.size = 10000)
fst <- pairwise.fst(af)
```

```
summary <- summarize_fst(fst)
```

summarize_genotype	<i>Summarize per-window genotype frequencies per population</i>
--------------------	---

Description

Computes the site-count-weighted mean and standard deviation of observed and expected heterozygosity, and of observed and expected haploid reference allele frequency, across all windows for each population. Uses the per-window table produced by [compute_allele.freqs_W()]. Weighting by N_sites ensures that windows with more called genotypes contribute more to each estimate.

Usage

```
summarize_genotype(geno_table)
```

Arguments

geno_table A [data.table::data.table] produced by [compute_allele.freqs_W()], containing at minimum the columns Pop, N_sites, Exp.Het, Obs.Het, Exp.M.Ref, and Obs.M.Ref.

Value

A [tibble::tibble] with one row per population and the following columns:

Pop Population label.

wMean.Exp.Het Weighted mean of expected heterozygosity across all windows.

wSD.Exp.Het Weighted standard deviation of expected heterozygosity across all windows.

wMean.Obs.Het Weighted mean of observed heterozygosity across all windows.

wSD.Obs.Het Weighted standard deviation of observed heterozygosity across all windows.

wMean.Exp.M.Ref Weighted mean of expected haploid reference allele frequency across all windows.

wSD.Exp.M.Ref Weighted standard deviation of expected haploid reference allele frequency across all windows.

wMean.Obs.M.Ref Weighted mean of observed haploid reference allele frequency across all windows.

wSD.Obs.M.Ref Weighted standard deviation of observed haploid reference allele frequency across all windows.

See Also

[compute_allele.freqs_W()] for computing the input per-window table.

Examples

```
vcf_path <- system.file("extdata",
                        "example.vcf",
                        package = "HaploDiploidEquilibrium")

result <- vcf2GT(vcf_path)
gt      <- result$gt_matrix
contigs <- result$contig_vector
pos     <- result$positions

pop.file <- data.frame(ID = colnames(gt),
                       Pop = c("PopA", "PopA", "PopB", "PopB", "PopB"))

geno <- compute_allele_freqs_W(geno.data = gt,
                               pop.file  = pop.file,
                               contigs    = contigs,
                               positions  = pos,
                               window.size = 10000,
                               dip_freq  = 0.5)

summary <- summarize_genos(geno)
```

summarize_NeisH

Summarize per-window Nei's H per population

Description

Computes the site-count-weighted mean and standard deviation of Nei's H across all windows for each population, using the per-window table produced by [compute_Hs_W()]. Weighting by N_sites ensures that windows with more called genotypes contribute more to the estimate.

Usage

```
summarize_NeisH(neis_table)
```

Arguments

neis_table A [data.table::data.table] produced by [compute_Hs_W()], containing at minimum the columns Pop, N_sites, and Neis_H.

Value

A [tibble::tibble] with one row per population and the following columns:

Pop Population label.

wMean.Neis_H Weighted mean of Nei's H across all windows.

wSD.Neis_H Weighted standard deviation of Nei's H across all windows.

See Also

[compute_Hs_W()] for computing the input per-window table.

Examples

```
vcf_path <- system.file("extdata",
                        "example.vcf",
                        package = "HaploDiploidEquilibrium")

result <- vcf2GT(vcf_path)
gt      <- result$gt_matrix
contigs <- result$contig_vector
pos     <- result$positions

pop.file <- data.frame(ID = colnames(gt),
                      Pop = c("PopA", "PopA", "PopB", "PopB", "PopB"))

hs <- compute_Hs_W(geno.data = gt,
                  pop.file   = pop.file,
                  contigs    = contigs,
                  positions  = pos,
                  window.size = 10000)

summary <- summarize_NeishH(hs)
```

summarize_sex_ref *Summarize per-sex reference allele frequencies per population*

Description

Computes the site-count-weighted mean and standard deviation of the reference allele frequency for females, males, and both sexes combined, across all windows for each population. Uses the per-window table produced by [compute.Female.Male.allele.W()]. Weighting by `N_sites` ensures that windows with more called genotypes contribute more to each estimate.

Usage

```
summarize_sex_ref(allele_table)
```

Arguments

`allele_table` A [data.table::data.table] produced by [compute.Female.Male.allele.W()], containing at minimum the columns `Pop`, `N_sites`, `Females.freq`, `Males.freq`, and `Total.freq`.

Value

A [tibble::tibble] with one row per population and the following columns:

Pop Population label.

wMean.F.Ref Weighted mean of the female reference allele frequency across all windows.

wSD.F.Ref Weighted standard deviation of the female reference allele frequency across all windows.

wMean.M.Ref Weighted mean of the male reference allele frequency across all windows.

wSD.M.Ref Weighted standard deviation of the male reference allele frequency across all windows.

wMean.T.Ref Weighted mean of the combined reference allele frequency across all windows.

wSD.T.Ref Weighted standard deviation of the combined reference allele frequency across all windows.

See Also

[compute.Female.Male.allele.W()] for computing the input per-window table.

Examples

```
vcf_path <- system.file("extdata",
                        "example.vcf",
                        package = "HaploDiploidEquilibrium")
result <- vcf2GT(vcf_path)
gt      <- result$gt_matrix
contigs <- result$contig_vector
pos     <- result$positions

pop.file <- data.frame(ID = colnames(gt),
                       Pop = c("PopA", "PopA", "PopB", "PopB", "PopB"))

sex_ref <- compute.Female.Male.allele.W(geno.data = gt,
                                         pop.file  = pop.file,
                                         contigs   = contigs,
                                         positions  = pos,
                                         window.size = 10000)

summary <- summarize_sex_ref(sex_ref)
```

vcf2GT

Import a VCF file and extract genotype and positional data

Description

Reads a VCF file from disk and extracts three pieces of information: the genotype calls (the GT field), the contig (chromosome) name for each variant site, and its physical position. These are the three inputs required by the windowed summary-statistic functions in this package.

Usage

```
vcf2GT(path_to_vcf)
```

Arguments

`path_to_vcf` A length-one character string giving the path to the input VCF file.

Value

A named list with three elements:

contig_vector A character vector of length `n_sites` containing the contig (chromosome) name for each variant.

positions A numeric vector of length `n_sites` containing the physical position (bp) of each variant.

gt_matrix A character matrix of genotype strings with dimensions `n_sites` x `n_individuals` (e.g. "0/0", "0/1", "1"). Row names are inherited from the VCF variant records and column names correspond to sample identifiers in the VCF header.

See Also

[`vcfR::read.vcfR()`] for full control over VCF parsing options.

Examples

```
vcf_path <- system.file("extdata",  
                        "example.vcf",  
                        package = "HaploDiploidEquilibrium")
```

```
result <- vcf2GT(vcf_path)
```

```
head(result$contig_vector)
```

```
head(result$positions)
```

```
head(result$gt_matrix)
```

Index

allele.freq.WS, [2](#)

compute.Female.Male.allele.W, [3](#)

compute_allele.freqs_W, [5](#)

compute_Hs_W, [8](#)

pairwise.fst, [9](#)

summarize_fst, [11](#)

summarize_geno, [12](#)

summarize_Neish, [13](#)

summarize_sex_ref, [14](#)

vcf2GT, [15](#)