

# Package ‘rehh’

July 17, 2020

**Maintainer** Alexander Klassmann <rehh@oneoverx.eu>

**Author** Mathieu Gautier, Alexander Klassmann and Renaud Vitalis

**Version** 3.1.2

**License** GPL (>= 2)

**Title** Searching for Footprints of Selection using 'Extended Haplotype Homozygosity' Based Tests

**Description** Population genetic data such as 'Single Nucleotide Polymorphisms' (SNPs) is often used to identify genomic regions that have been under recent natural or artificial selection and might provide clues about the molecular mechanisms of adaptation. One approach, the concept of an 'Extended Haplotype Homozygosity' (EHH), introduced by (Sabeti 2002) <doi:10.1038/nature01140>, has given rise to several statistics designed for whole genome scans. The package provides functions to compute three of these, namely: 'iHS' (Voight 2006) <doi:10.1371/journal.pbio.0040072> for detecting positive or 'Darwinian' selection within a single population as well as 'Rsb' (Tang 2007) <doi:10.1371/journal.pbio.0050171> and 'XP-EHH' (Sabeti 2007) <doi:10.1038/nature06250>, targeted at differential selection between two populations. Various plotting functions are also included to facilitate visualization and interpretation of these statistics. Due to changes in the API, albeit mostly minor, versions 3.X are not compatible with versions 2.0.X. Note: optionally, vcf files can be imported using package vcfR. That package is currently removed from CRAN, but can still be installed from <<https://github.com/knausb/vcfR>> following instructions there.

**Depends** R (>= 2.10)

**Imports** methods, rehh.data

**Suggests** ape, bookdown, data.table, gap, knitr, qqman, rmarkdown, R.utils, testthat, vcfR

**VignetteBuilder** knitr

**NeedsCompilation** yes

**RoxygenNote** 7.1.0

**URL** <https://CRAN.R-project.org/package=rehh>,  
<https://gitlab.com/oneoverx/rehh>

**BugReports** <https://gitlab.com/oneoverx/rehh/issues>

**Repository** CRAN

**Date/Publication** 2020-07-17 10:30:03 UTC

## R topics documented:

rehh-package	3
allelefurcation-class	4
as.newick	4
calc_candidate_regions	5
calc_ehh	6
calc_ehhs	9
calc_furcation	11
calc_haplen	13
calc_pairwise_haplen	14
calc_region_stats	15
data2haplohh	16
distribplot	19
extract_regions	20
freqbinplot	21
ftree-class	22
furcation-class	23
haplen-class	24
haplohh-class	24
haplohh_cgu_bta12	25
ies2xpehh	26
ihh2ihs	27
ines2rsb	29
make.example.files	31
manhattanplot	32
plot.ehh	34
plot.ehhs	35
plot.furcation	37
plot.haplen	38
plot.haplohh	40
remove.example.files	42
scan_hh	42
scan_hh_full	45
subset.haplohh	47
update_haplohh	48

---

rehh-package	<i>rehh: Searching for Footprints of Selection using 'Extended Haplotype Homozygosity' Based Tests</i>
--------------	--

---

## Description

Population genetic data such as 'Single Nucleotide Polymorphisms' (SNPs) is often used to identify genomic regions that have been under recent natural or artificial selection and might provide clues about the molecular mechanisms of adaptation. One approach, the concept of an 'Extended Haplotype Homozygosity' (EHH), introduced by (Sabeti 2002) <doi:10.1038/nature01140>, has given rise to several statistics designed for whole genome scans. The package provides functions to compute three of these, namely: 'iHS' (Voight 2006) <doi:10.1371/journal.pbio.0040072> for detecting positive or 'Darwinian' selection within a single population as well as 'Rsb' (Tang 2007) <doi:10.1371/journal.pbio.0050171> and 'XP-EHH' (Sabeti 2007) <doi:10.1038/nature06250>, targeted at differential selection between two populations. Various plotting functions are also included to facilitate visualization and interpretation of these statistics. Due to changes in the API, albeit mostly minor, versions 3.X are not compatible with versions 2.0.X. Note: optionally, vcf files can be imported using package vcfR. That package is currently removed from CRAN, but can still be installed from <<https://github.com/knausb/vcfR>> following instructions there.

## Details

See vignette("rehh", package = "rehh") for an overview of the package and vignette("examples", package = "rehh") for a more detailed discussion of two small example data sets.

## References

- Gautier, M. and Naves, M. (2011). Footprints of selection in the ancestral admixture of a New World Creole cattle breed. *Molecular Ecology*, **20**, 3128-3143.
- Gautier M. and Vitalis R. (2012). rehh: An R package to detect footprints of selection in genome-wide SNP data from haplotype structure. *Bioinformatics*, **28**(8), 1176-1177.
- Gautier M., Klassmann A., and Vitalis R. (2017). rehh 2.0: a reimplement of the R package rehh to detect positive selection from haplotype structure. *Molecular Ecology Resources*, **17**, 78-90.
- Sabeti, P.C. et al. (2002). Detecting recent positive selection in the human genome from haplotype structure. *Nature*, **419**, 832-837.
- Sabeti, P.C. et al. (2007). Genome-wide detection and characterization of positive selection in human populations. *Nature*, **449**, 913-918.
- Tang, K. and Thornton, K.R. and Stoneking, M. (2007). A New Approach for Using Genome Scans to Detect Recent Positive Selection in the Human Genome. *Plos Biology*, **7**, e171.
- Voight, B.F. and Kudravalli, S. and Wen, X. and Pritchard, J.K. (2006). A map of recent positive selection in the human genome. *Plos Biology*, **4**, e72.

**See Also**

Useful links:

- <https://CRAN.R-project.org/package=rehh>
- <https://gitlab.com/oneoverx/rehh>
- Report bugs at <https://gitlab.com/oneoverx/rehh/issues>

---

allelefurcation-class *An S4 class containing furcation trees for one allele of a focal marker*

---

**Description**

An S4 class containing the furcation trees for both sides of a focal marker for one allele.

**Slots**

allele the allele of the focal marker.  
description "ancestral", "derived", "major", "minor", etc.  
count the number of chromosomes with that allele.  
left furcation tree to the left of the marker.  
right furcation tree to the right of the marker.

**See Also**

[ftree](#), [furcation](#)

---

as.newick *Convert a furcation tree into Newick format*

---

**Description**

Convert a furcation tree into Newick format.

**Usage**

```
as.newick(furcation, allele = 0, side, hap.names = seq_len(furcation@nhap))
```

**Arguments**

furcation	an object of <a href="#">furcation-class</a> .
allele	the allele to be considered (default 0).
side	side (either "left" or "right").
hap.names	names/labels of chromosomes in haplotype data file. Per default haplotypes are numbered by their order in the input file.

**See Also**

[ftree-class](#), [calc\\_furcation](#), [plot.furcation](#)

**Examples**

```
#example haplohh object (280 haplotypes, 1424 SNPs)
#see ?haplohh_cgu_bta12 for details
data(haplohh_cgu_bta12)
#calculate furcation for the marker "F1205400"
#which displays a strong signal of selection
f <- calc_furcation(haplohh_cgu_bta12, mrk = "F1205400")
#get left tree of ancestral allele (coded as '0')
as.newick(f, 0, "left")
```

---

calc\_candidate\_regions

*Determine candidate regions of selection*

---

**Description**

Determine candidate regions of selection.

**Usage**

```
calc_candidate_regions(
  scan,
  threshold = NA,
  pval = FALSE,
  ignore_sign = FALSE,
  window_size = 1e+06,
  overlap = 0,
  right = TRUE,
  min_n_mrk = 1,
  min_n_extr_mrk = 1,
  min_perc_extr_mrk = 0,
  join_neighbors = TRUE
)
```

**Arguments**

scan	a data frame containing scores (output of <a href="#">ihh2ihs</a> , <a href="#">ines2rsb</a> or <a href="#">ies2xpehh</a> ).
threshold	boundary score above which markers are defined as "extreme".
pval	logical. If TRUE use the (negative log-) p-value instead of the score.
ignore_sign	logical. If TRUE (default), take absolute values of score.
window_size	size of sliding windows. If set to 1, no windows are constructed and only the individual extremal markers are reported.

overlap	size of window overlap (default 0, i.e. no overlap).
right	logical, indicating if the windows should be closed on the right (and open on the left) or vice versa.
min_n_mrk	minimum number of markers per window.
min_n_extr_mrk	minimum number of markers with extreme value in a window.
min_perc_extr_mrk	minimum percentage of extremal markers among all markers.
join_neighbors	logical. If TRUE (default), merge neighboring windows with extreme values.

### Details

There is no generally agreed method how to determine genomic regions which might have been under recent selection. Since selection tends to yield clusters of markers with outlier values, a common approach is to search for regions with an elevated number or fraction of outlier or extremal markers. This function allows to set three conditions a window must fulfill in order to classify as candidate region:

- min\_n\_mrk a minimum number of (any) markers.
- min\_n\_extr\_mrk a minimum number of markers with outlier / extreme value.
- min\_perc\_extr\_mrk a minimum percentage of extremal markers among all markers.

"Extreme" markers are defined by having a score above the specified threshold.

### Value

A data frame with chromosomal regions, i.e. windows that fulfill the necessary conditions to qualify as candidate regions under selection. For each region the overall number of markers, their mean and maximum, the number of markers with extremal values, their percentage of all markers and their average are reported.

### See Also

[calc\\_region\\_stats](#)

---

calc\_ehh

*EHH and iHH computation for a given focal marker*

---

### Description

Compute Extended Haplotype Homozygosity (EHH) and integrated EHH (iHH) for a given focal marker.

**Usage**

```
calc_ehh(
  haplohh,
  mrk,
  limhaplo = 2,
  limhomohaplo = 2,
  limehh = 0.05,
  include_zero_values = FALSE,
  include_nhaplo = FALSE,
  phased = TRUE,
  polarized = TRUE,
  scalegap = NA,
  maxgap = NA,
  discard_integration_at_border = TRUE,
  lower_y_bound = limehh,
  interpolate = TRUE
)
```

**Arguments**

haplohh	an object of class haplohh (see <a href="#">data2haplohh</a> ).
mrk	integer representing the number of the focal marker within the haplohh object or string representing its ID/name.
limhaplo	if there are less than limhaplo chromosomes that can be used for the calculation of EHH, the calculation is stopped. The option is intended for the case of missing data, which leads to the successive exclusion of haplotypes: the further away from the focal marker the less haplotypes contribute to EHH.
limhomohaplo	if there are less than limhomohaplo homozygous chromosomes, the calculation is stopped. This option is intended for unphased data and should be invoked only if relatively low frequency variants are not filtered subsequently (see main vignette and Klassmann et al. 2020).
limehh	limit at which EHH stops to be evaluated
include_zero_values	logical. If FALSE, return values only for those positions where the calculation is actually performed, i.e. until stopped by reaching either limehh or limhaplo. If TRUE, report EHH values for all markers, the additional ones being zero.
include_nhaplo	logical. If TRUE, report the number of evaluated haplotypes at each marker (only informative, if missing data leads to a decrease of evaluated haplotypes).
phased	logical. If TRUE (default) chromosomes are expected to be phased. If FALSE, the haplotype data is assumed to consist of pairwise ordered chromosomes belonging to diploid individuals. EHH is then estimated over individuals which are homozygous at the focal marker.
polarized	logical. TRUE by default. If FALSE, use major and minor allele instead of ancestral and derived. If there are more than two alleles then the minor allele refers to the second-most frequent allele.

scalegap	scale or cap gaps larger than the specified size to the specified size (default=NA, i.e. no scaling).
maxgap	maximum allowed gap in bp between two markers. If exceeded, further calculation of EHH is stopped at the gap (default=NA, i.e no limitation).
discard_integration_at_border	logical. If TRUE (default) and computation reaches first or last marker or a gap larger than maxgap, iHH is set to NA.
lower_y_bound	lower y boundary of the area to be integrated over (default: 1imehh). Can be set to zero for compatibility with the program hapbin.
interpolate	logical. Affects only IHH values. If TRUE (default), integration is performed over a continuous EHH curve (values are interpolated linearly between consecutive markers), otherwise the EHH curve decreases stepwise at markers.

### Details

Values for allele-specific Extended Haplotype Homozygosity (EHH) are computed upstream and downstream of the focal marker for each of its alleles. These values are integrated with respect to their genomic positions to yield an 'integrated EHH' (iHH) value for each allele.

### Value

The returned value is a list containing the following elements:

**mrk.name** The name/identifier of the focal marker.

**freq** A vector with the frequencies of the alleles of the focal marker.

**ehh** A data frame with EHH values for each allele of the focal marker.

**iHH** A vector with iHH (integrated EHH) values for each allele of the focal marker.

### References

- Gautier, M. and Naves, M. (2011). Footprints of selection in the ancestral admixture of a New World Creole cattle breed. *Molecular Ecology*, **20**, 3128-3143.
- Klassmann, A. et al. (2020). Detecting selection using Extended Haplotype Homozygosity (EHH)-based statistics on unphased or unpolarized data. (submitted).
- Sabeti, P.C. et al. (2002). Detecting recent positive selection in the human genome from haplotype structure. *Nature*, **419**, 832-837.
- Sabeti, P.C. et al. (2007). Genome-wide detection and characterization of positive selection in human populations. *Nature*, **449**, 913-918.
- Tang, K. and Thornton, K.R. and Stoneking, M. (2007). A New Approach for Using Genome Scans to Detect Recent Positive Selection in the Human Genome. *Plos Biology*, **7**, e171.
- Voight, B.F. and Kudravalli, S. and Wen, X. and Pritchard, J.K. (2006). A map of recent positive selection in the human genome. *Plos Biology*, **4**, e72.

### See Also

[data2haplohh](#), [plot.ehh](#), [calc\\_ehhs](#), [scan\\_hh](#).



**Examples**

```
#example haplohh object (280 haplotypes, 1424 SNPs)
#see ?haplohh_cgu_bta12 for details
data(haplohh_cgu_bta12)
#computing EHH statistics for the marker "F1205400"
#which displays a strong signal of selection
ehh <- calc_ehh(haplohh_cgu_bta12, mrk = "F1205400")
```

calc\_ehhs

*EHHS and iES computation for a given focal marker***Description**

Compute site-specific Extended Haplotype Homozygosity (EHHS) and integrated EHHS (iES) for a given focal marker.

**Usage**

```
calc_ehhs(
  haplohh,
  mrk,
  limhaplo = 2,
  limhomohaplo = 2,
  limehhs = 0.05,
  include_zero_values = FALSE,
  include_nhaplo = FALSE,
  phased = TRUE,
  scalegap = NA,
  maxgap = NA,
  discard_integration_at_border = TRUE,
  lower_y_bound = limehhs,
  interpolate = TRUE
)
```

**Arguments**

haplohh	an object of class haplohh (see <a href="#">data2haplohh</a> ).
mrk	integer representing the number of the focal marker within the haplohh object or string representing its ID/name.
limhaplo	if there are less than limhaplo chromosomes that can be used for the calculation of EHH, the calculation is stopped. The option is intended for the case of missing data, which leads to the successive exclusion of haplotypes: the further away from the focal marker the less haplotypes contribute to EHH.
limhomohaplo	if there are less than limhomohaplo homozygous chromosomes, the calculation is stopped. This option is intended for unphased data and should be invoked only if relatively low frequency variants are not filtered subsequently (see main vignette and Klassmann et al. 2020).

limehhs	limit at which EHHS stops to be evaluated.
include_zero_values	logical. If FALSE, return values only for those positions where the calculation is actually performed, i.e. until stopped by reaching either limehh or limhaplo. If TRUE, report EHH values for all markers, the additional ones being zero.
include_nhaplo	logical. If TRUE, report the number of evaluated haplotypes at each marker (only informative, if missing data leads to a decrease of evaluated haplotypes).
phased	logical. If TRUE (default) chromosomes are expected to be phased. If FALSE, the haplotype data is assumed to consist of pairwise ordered chromosomes belonging to diploid individuals. EHHS is then estimated over individuals which are homozygous at the focal marker.
scalegap	scale or cap gaps larger than the specified size to the specified size (default=NA, i.e. no scaling).
maxgap	maximum allowed gap in bp between two markers. If exceeded, further calculation of EHHS is stopped at the gap (default=NA, i.e. no limitation).
discard_integration_at_border	logical. If TRUE (default) and computation reaches first or last marker or a gap larger than maxgap, iHH is set to NA.
lower_y_bound	lower y boundary of the area to be integrated over (default: limehhs). Can be set to zero for compatibility with the program hapbin.
interpolate	logical. Affects only IES and INES values. If TRUE (default), integration is performed over a continuous EHHS curve (values are interpolated linearly between consecutive markers), otherwise the EHHS curve decreases stepwise at markers.

### Details

Values for site-specific Extended Haplotype Homozygosity (EHHS) are computed at each position upstream and downstream of the focal marker. These values are integrated with respect to their genomic position to yield an 'integrated EHHS' (IES) value.

### Value

The returned value is a list containing the following elements:

**mrk.name** The name/identifier of the focal marker.

**ehhs** A table containing EHHS values as used by Sabeti et al. (2007), resp. the same values normalized to 1 at the focal marker (nEHHS) as used by Tang et al. (2007).

**IES** Integrated EHHS.

**INES** Integrated normalized EHHS.

### References

- Gautier, M. and Naves, M. (2011). Footprints of selection in the ancestral admixture of a New World Creole cattle breed. *Molecular Ecology*, **20**, 3128-3143.
- Klassmann, A. et al. (2020). Detecting selection using Extended Haplotype Homozygosity (EHH)-based statistics on unphased or unpolarized data. (submitted).

Sabeti, P.C. et al. (2002). Detecting recent positive selection in the human genome from haplotype structure. *Nature*, **419**, 832-837.

Sabeti, P.C. et al. (2007). Genome-wide detection and characterization of positive selection in human populations. *Nature*, **449**, 913-918.

Tang, K. and Thornton, K.R. and Stoneking, M. (2007). A New Approach for Using Genome Scans to Detect Recent Positive Selection in the Human Genome. *Plos Biology*, **7**, e171.

Voight, B.F. and Kudravalli, S. and Wen, X. and Pritchard, J.K. (2006). A map of recent positive selection in the human genome. *Plos Biology*, **4**, e72.

### See Also

[data2haplohh](#), [plot.ehhs](#), [calc\\_ehh](#), [scan\\_hh](#).

### Examples

```
#example haplohh object (280 haplotypes, 1424 SNPs)
#see ?haplohh_cgu_bta12 for details
data(haplohh_cgu_bta12)
#computing EHS statistics for the marker "F1205400"
#which displays a strong signal of selection
ehhs <- calc_ehhs(haplohh_cgu_bta12, mrk = "F1205400")
```

---

calc\_furcation

*calculate furcation trees around a focal marker*

---

### Description

Calculate furcation trees around a focal marker. A furcation tree captures in greater detail than EHH values the decrease of extended haplotype homozygosity at increasing distances from the selected focal marker.

### Usage

```
calc_furcation(
  haplohh,
  mrk,
  allele = NA,
  limhaplo = 2,
  phased = TRUE,
  polarized = TRUE
)
```

**Arguments**

haplohh	an object of class haplohh (see <a href="#">data2haplohh</a> ).
mrk	integer representing the number of the focal marker within the haplohh object or string representing its ID/name.
allele	a vector of alleles as coded internally, i.e. in case of polarized alleles, 0 represents the ancestral, 1 or higher the derived alleles. If NULL, all alleles of the focal marker are considered.
limhaplo	if there are less than limhaplo chromosomes that can be used for the calculation, it is stopped. This is useful in case of missing data, which lead to a successive exclusion of haplotypes: the further away from the focal marker the less haplotypes are evaluated.
phased	logical. If TRUE (default), chromosomes are expected to be phased. If FALSE, consecutive chromosomes are assumed to belong to diploid individuals and furcation trees are limited to within individuals which are homozygous at the focal marker.
polarized	logical. Affects only the order of furcations. If TRUE (default), the ancestral allele becomes the first furcation and derived alleles are sorted by their internal coding. Otherwise all alleles are sorted by their internal coding.

**Details**

A haplotype furcation tree visualizes the breakdown of LD at increasing distances from the focal marker. The root of each tree is an allele of the focal marker, which in turn is identified by a vertical dashed line. Moving either to the "left" or to the "right" of the focal marker, each further marker is an opportunity for a node; the tree either divides or does not, based on whether alleles at that marker distinguish between hitherto identical extended haplotypes. The thickness of the lines corresponds to the number of chromosomes sharing an extended haplotype.

**Value**

An object of class furcation, containing the furcation structure of the specified alleles at the focal marker.

**References**

Sabeti, P.C. and Reich, D.E. and Higgins, J.M. and Levine, H.Z.P and Richter, D.J. and Schaffner, S.F. and Gabriel, S.B. and Platko, J.V. and Patterson, N.J. and McDonald, G.J. and Ackerman, H.C. and Campbell, S.J. and Altshuler, D. and Cooper, R. and Kwiatkowski, D. and Ward, R. and Lander, E.S. (2002). Detecting recent positive selection in the human genome from haplotype structure. *Nature*, 419, 832-837.

**See Also**

[plot.furcation](#), [calc\\_haplen](#).

**Examples**

```
#example haplohh object (280 haplotypes, 1424 SNPs)
#see ?haplohh_cgu_bta12 for details
data(haplohh_cgu_bta12)
#plotting a furcation diagram for both ancestral and derived allele
#from the marker "F1205400"
#which display a strong signal of selection
f <- calc_furcation(haplohh_cgu_bta12, mrk = "F1205400")
plot(f)
```

---

 calc\_haplen

*Calculate length of longest shared haplotypes around a focal marker*


---

**Description**

Calculate for each chromosome the maximum length of its extended haplotype homozygosity.

**Usage**

```
calc_haplen(furcation)
```

**Arguments**

furcation      an object of class furcation calculated by [calc\\_furcation](#).

**Details**

Extended haplotype homozygosity is defined as the region around a focal marker in which a particular chromosome shares a haplotype with (its sequence is identical to) another chromosome. The function calculates for each chromosome the boundaries of its longest shared haplotype. These correspond to the last furcations of a chromosome in a furcation diagram. Note that the calculation is performed independently upstream and downstream of the focal marker and hence upper and lower boundaries do not necessarily arise from the same chromosomal pair.

**Value**

The function returns a list containing four elements:

**mrk.name** name/identifier of the focal marker.

**position** position of the focal marker.

**xlim** positions of left- and rightmost markers covered by extended haplotypes.

**haplen** a data frame with the coordinates of extended haplotypes around the focal marker.

## Examples

```
#example haplohh object (280 haplotypes, 1424 SNPs)
#see ?haplohh_cgu_bta12 for details
data(haplohh_cgu_bta12)
#plotting haplotype lengths for both ancestral and derived allele
#of the marker "F1205400"
#which displays a strong signal of selection
f <- calc_furcation(haplohh_cgu_bta12, mrk = "F1205400")
h <- calc_haplen(f)
plot(h)
```

---

calc\_pairwise\_haplen *Calculate pairwise shared haplotype length between all chromosomes*

---

## Description

Calculate pairwise shared haplotype length between all chromosomes at a focal marker.

## Usage

```
calc_pairwise_haplen(haplohh, mrk, phased = TRUE, maxgap = NA)
```

## Arguments

haplohh	an object of class haplohh (see <a href="#">data2haplohh</a> ).
mrk	integer representing the number of the focal marker within the haplohh object or string representing its ID/name.
phased	logical. If TRUE (default) chromosomes are expected to be phased. If FALSE, the haplotype data is assumed to consist of pairwise ordered chromosomes belonging to diploid individuals and only the two chromosomes of each individual are compared.
maxgap	maximum allowed gap in bp between two markers. If exceeded, further calculation is stopped at the gap (default=NA, i.e no limitation).

## Details

The function computes the length of shared haplotypes (stretches of identical sequence) around the focal marker.

Note that the function [calc\\_haplen](#) calculates for each chromosome the boundaries of its longest shared haplotype; separately upstream and downstream of the focal marker.

## Value

The returned value is a matrix with pairwise shared haplotype lengths.

**See Also**

[data2haplohh](#), [scan\\_hh\\_full](#).

**Examples**

```
#example haplohh object (280 haplotypes, 1424 SNPs)
#see ?haplohh_cgu_bta12 for details
data(haplohh_cgu_bta12)
#computing shared haplotype lengths around the marker "F1205400"
#which displays a strong signal of selection
m <- calc_pairwise_haplen(haplohh_cgu_bta12, mrk = "F1205400")
```

---

calc\_region\_stats      *Calculate score statistics for given regions*

---

**Description**

Calculate score statistics (extremal values) for given regions. This function is intended for the comparison of different scores for the same chromosomal regions.

**Usage**

```
calc_region_stats(
  scan,
  regions,
  threshold = NA,
  pval = FALSE,
  ignore_sign = FALSE,
  right = TRUE
)
```

**Arguments**

scan	a data frame containing scores (output of <a href="#">ihh2ihs</a> , <a href="#">ines2rsb</a> or <a href="#">ies2xpehh</a> ).
regions	a data frame with column names CHR, START and END, specifying chromosomal regions (e.g. as obtained by function <a href="#">calc_candidate_regions</a> ).
threshold	boundary score above which markers are defined as "extreme".
pval	logical. If TRUE use the (negative log-) p-value instead of the score.
ignore_sign	logical. If TRUE (default), take absolute values of score.
right	logical, indicating if the regions should be closed on the right (and open on the left) or vice versa.

**Value**

A data frame with chromosomal regions. For each region the overall number of markers, their mean and maximum, the number of markers with extremal values, their percentage of all markers and their average are reported.

**See Also**

[calc\\_candidate\\_regions](#)

---

data2haplohh

*Convert data from input file to an object of class haplohh*

---

**Description**

Convert input data files to an object of [haplohh-class](#).

**Usage**

```
data2haplohh(
  hap_file,
  map_file = NA,
  min_perc_genotype.hap = NA,
  min_perc_genotype.mrk = 100,
  min_maf = NA,
  chr.name = NA,
  popsel = NA,
  recode.allele = FALSE,
  allele_coding = "12",
  haplotype.in.columns = FALSE,
  remove_multiple_markers = FALSE,
  polarize_vcf = TRUE,
  capitalize_AA = TRUE,
  vcf_reader = "data.table",
  position_scaling_factor = NA,
  verbose = TRUE
)
```

**Arguments**

**hap\_file** file containing haplotype data (see details below).

**map\_file** file containing map information (see details below).

**min\_perc\_genotype.hap** threshold on percentage of missing data for haplotypes (haplotypes with less than `min_perc_genotype.hap` percent of markers genotyped are discarded). Default is NA, hence no constraint.

**min\_perc\_genotype.mrk** threshold on percentage of missing data for markers (markers genotyped on less than `min_perc_genotype.mrk` percent of haplotypes are discarded). By default, `min_perc_genotype.mrk=100`, hence only fully genotyped markers are retained. This value cannot be set to NA or zero.



<code>min_maf</code>	threshold on the Minor Allele Frequency. Markers having a MAF lower than or equal to <code>minmaf</code> are discarded. In case of multi-allelic markers the second-most frequent allele is referred to as minor allele. Setting this value to zero eliminates monomorphic sites. Default is NA, hence no constraint.
<code>chr.name</code>	name of the chromosome considered (relevant if data for several chromosomes is contained in the haplotype or map file).
<code>popsel</code>	code of the population considered (relevant for fastPHASE output which can contain haplotypes from various populations).
<code>recode.allele</code>	<i>*Deprecated*</i> . logical. FALSE by default. TRUE forces parameter <code>allele_coding</code> to "map", FALSE leaves it unchanged.
<code>allele_coding</code>	the allele coding provided by the user. Either "12" (default), "01", "map" or "none". The option is irrelevant for vcf files and ms output.
<code>haplotype.in.columns</code>	logical. If TRUE, phased input haplotypes are assumed to be in columns (as produced by the SHAPEIT2 program (O'Connell et al., 2014)).
<code>remove_multiple_markers</code>	logical. If FALSE (default), conversion stops, if multiple markers with the same chromosomal position are encountered. If TRUE, duplicated markers are removed (all but the first marker with identical positions).
<code>polarize_vcf</code>	logical. Only of relevance for vcf files. If TRUE (default), tries to polarize variants with help of the AA entry in the INFO field. Unpolarized alleles are discarded. If FALSE, allele coding of vcf file is used unchanged as internal coding.
<code>capitalize_AA</code>	logical. Only of relevance for vcf files with ancestral allele information. Low confidence ancestral alleles are usually coded by lower-case letters. If TRUE (default), these are changed to upper case before the alleles of the sample are matched for polarization.
<code>vcf_reader</code>	library used to read vcf. By default, low-level parsing is performed using the generic package <code>data.table</code> . In order to read compressed files, the package <code>R.utils</code> must be installed, too. If the specialized package <code>vcfR</code> is available, set this parameter to "vcfR".
<code>position_scaling_factor</code>	intended primarily for output of ms where positions lie in the interval [0,1]. These can be rescaled to sizes of typical markers in real data.
<code>verbose</code>	logical. If TRUE (default), report verbose progress.

## Details

Five haplotype input formats are supported:

- a "standard format" with haplotypes in rows and markers in columns (with no header, but a haplotype ID/name in the first column).
- a "transposed format" similar to the one produced by the phasing program SHAPEIT2 (O'Connell et al., 2014) in which haplotypes are in columns and markers in rows (with neither header nor marker IDs nor haplotype IDs).

- output files from the fastPHASE program (Sheet and Stephens, 2006). If haplotypes from several different population were phased simultaneously (-u fastPHASE option was used), it is necessary to specify the population of interest by parameter `popse1` (if this parameter is not or wrongly set, the error message will provide a list of the population numbers contained in the file).
- files in variant call format (vcf). No mapfile is needed in this case. If the file contains several chromosomes, it is necessary to choose one by parameter `chr.name`.
- output of the simulation program 'ms'. No mapfile is needed in this case. If the file contains several 'runs', a specific number has to be specified by the parameter `chr.name`.

The "transposed format" has to be explicitly set while the other formats are recognized automatically.

The map file contains marker information in three, or, if it is used for polarization (see below), five columns:

- marker name/id
- chromosome
- position (physical or genetic)
- ancestral allele encoding
- derived allele encoding

The markers must be in the same order as in the haplotype file. If several chromosomes are represented in the map file, it is necessary to choose that which corresponds to the haplotype file by parameter `chr.name`.

Haplotypes can be given either with alleles already coded as numbers (in two possible ways) or with the actual alleles (e.g. nucleotides) which can be translated into numbers either using the fourth and fifth column of the map file or by their alpha-numeric order. Correspondingly, the parameter `allele_coding` has to be set to either "12", "01", "map" or "none":

- "12": 0 represents missing values, 1 the ancestral allele and 2 (or higher integers) derived allele(s).
- "01": NA or '.' (a point) represent missing values, 0 the ancestral and 1 (or higher integers) derived allele(s).
- "map": for each marker, the fourth column of the map file defines the ancestral allele and the fifth column derived alleles. In case of multiple derived alleles, they must be separated by commas without space. Alleles in the haplotype file which do not appear in neither of the two columns of the map file are regarded as missing values (NA).
- "none": NA or '.' (a point) represent missing values, otherwise for each marker the allele that comes first in alpha-numeric order is coded by 0, the next by 1, etc. Evidently, this coding does not convey any information about allele status as ancestral or derived, hence the alleles cannot be regarded as polarized.

The information of allelic ancestry is exploited only in the frequency-bin-wise standardization of iHS (see [ihh2ihs](#)). However, although ancestry status does not figure in the formulas of the cross populations statistics `Rsb` and `XP-EHH`, their values do depend on the assigned status.

The arguments `min_perc_genos.hap`, `min_perc_genos.mrk` and `min_maf` are evaluated in this order.

**Value**

The returned value is an object of `haplohh-class`.

**References**

Scheet P, Stephens M (2006) A fast and flexible statistical model for large-scale population genotype data: applications to inferring missing genotypes and haplotypic phase. *Am J Hum Genet*, **78**, 629-644.

O'Connell J, Gurdasani D, Delaneau O, et al (2014) A general approach for haplotype phasing across the full spectrum of relatedness. *PLoS Genet*, **10**, e1004234.

**Examples**

```
#copy example files into the current working directory.
make.example.files()
#create object using a haplotype file in "standard format"
hap <- data2haplohh(hap_file = "bta12_cgu.hap",
                   map_file = "map.inp",
                   chr.name = 12,
                   allele_coding = "map")
#create object using fastPHASE output
hap <- data2haplohh(hap_file = "bta12_hapguess_switch.out",
                   map_file = "map.inp",
                   chr.name = 12,
                   popsel = 7,
                   allele_coding = "map")
#clean up demo files
remove.example.files()
```

---

distribplot

*Plot distribution of standardized iHS, Rsb or XP-EHH values*

---

**Description**

Plot the observed distribution of standardized iHS, Rsb or XP-EHH values together with the standard Gaussian distribution.

**Usage**

```
distribplot(
  data,
  lty = 1,
  lwd = 1.5,
  col = c("blue", "red"),
  qqplot = FALSE,
  resolution = 0.01,
  ...
)
```

**Arguments**

data	a vector of iHS, Rsb or XPEHH values.
lty	line type.
lwd	line width.
col	a vector describing the colors of the observed and Gaussian distribution, respectively.
qqplot	logical. If TRUE a qq-plot is drawn instead of the distribution density curve.
resolution	affects only qqplot. Rasterize data points to a quadratic grid with the specified resolution and remove duplicate points. Defaults to 0.01.
...	further arguments passed to <a href="#">plot.default</a> .

**Value**

The function returns a plot.

**See Also**

[ihh2ihs](#), [ines2rsb](#), [ies2xpehh](#), [manhattanplot](#).

**Examples**

```
library(rehh.data)
#results from a genome scan (44,057 SNPs) see ?wgscan.cgu for details
data(wgscan.cgu)
#extract vector with iHS values from data frame
IHS <- ihh2ihs(wgscan.cgu)$ihs[["IHS"]]
distribplot(IHS, main = "iHS (CGU population)")
distribplot(IHS, main = "iHS (CGU population)", qqplot = TRUE)
```

---

extract_regions	<i>Extract regions from a scan</i>
-----------------	------------------------------------

---

**Description**

Extract regions from a scan data frame.

**Usage**

```
extract_regions(scan, regions, right = TRUE)
```

**Arguments**

scan	A data frame with chromosomal positions like obtained by <a href="#">scan_hh</a> , <a href="#">ihh2ihs</a> , <a href="#">ines2rsb</a> or <a href="#">ies2xpehh</a> .
regions	A data frame with genomic regions like the output of <a href="#">calc_candidate_regions</a> .
right	logical, indicating if the intervals should be closed on the right (and open on the left) or vice versa.

**Value**

A subset of data frame scan, retaining only positions belonging to the regions specified in data frame regions.

**Examples**

```
library(rehh.data)
data(wgscan.cgu)
regions <- data.frame(CHR = 12, START = 2.88e+7, END = 2.92e+7)
extract_regions(wgscan.cgu, regions)
```

---

freqbinplot

*Plot of unstandardized iHS within frequency bins*


---

**Description**

Plot of unstandardized iHS within frequency bins.

**Usage**

```
freqbinplot(
  x,
  spectrum = FALSE,
  main = NA,
  xlab = "Derived allele frequency",
  ylab = NA,
  xlim = c(0, 1),
  ylim = NULL,
  pch = 20,
  ...
)
```

**Arguments**

x	data (output of function <code>ihh2ihs</code> )
spectrum	logical. If TRUE, plot frequency spectrum instead of iHS.
main	an overall title for the plot.
xlab	a title for the x axis.
ylab	a title for the y axis.
xlim	the x coordinate range of the plot.
ylim	the y coordinate range of the plot.
pch	plotting 'character' see <code>points</code> .
...	further arguments to be passed to <code>plot</code> resp. <code>points</code> .

**Details**

The plot shows the mean and the quantiles calculated by function `ihh2ihs` for the unstandardized iHS in each frequency bin. Note that the standardization of iHS is performed bin-wise in order to reduce the frequency-dependence of iHS values (expected under neutrality). An implicit assumption of this procedure is that each bin is dominated by neutral markers.

**See Also**

[ihh2ihs](#)

**Examples**

```
library(rehh.data)
data(wgscan.cgu)
#results from a genome scan (44,057 SNPs)
#see ?wgscan.eut and ?wgscan.cgu for details
wgscan.cgu.ihs <- ihh2ihs(wgscan.cgu)
freqbinplot(wgscan.cgu.ihs)
```

---

ftree-class	<i>An S4 class to represent a furcation tree on one side of one allele of a focal marker</i>
-------------	--

---

**Description**

An S4 class to represent a furcation tree on one side of one allele of a focal marker

**Details**

A furcation structure consists of two trees ("left" and "right") for each allele of a focal marker. If there are only bi-allelic markers and no missing values, the trees are bifurcating.

Missing values are treated similarly to an extra allele in so far as they cause a furcation. However, the resulting daughter node is marked accordingly and the chromosomes excluded from further calculations. If all chromosomes of a parent node have missing values, the "furcation" is degenerated and yields a single daughter node.

Note that a tree with  $n$  leaves can have at most  $2n-1$  nodes.

In a furcation tree, the leaves do not necessarily represent single chromosomes, either due to multiple missing data or because the first/last marker was reached before all extended haplotypes were distinct.

**Slots**

`node_parent` a vector, representing the tree structure. Each node (number) is assigned its parent node (number).

`node_pos` a vector, assigning to each node (number) its position in the chromosome, i.e. at which marker position the furcation occurred.

`node_with_missing_data` a vector of type logical. Pseudo-furcations arise due to missing data at a marker. The daughter node (number) is marked accordingly.

`label_parent` a vector, that attaches an "extra leave", representing the haplotype number (defined by the order in the haplotype data file) to leaves of the tree. This is necessary because in general not all leaves of the original tree represent a single haplotype/chromosome.

---

<code>furcation-class</code>	<i>An S4 class representing the complete furcation pattern around a focal marker.</i>
------------------------------	---

---

### Description

An S4 class representing the complete furcation pattern around a focal marker.

### Slots

`.Data` a list containing for each allele an object of `allelefurcation-class`.

`mrk.name` the name/identifier of the focal marker.

`position` the chromosomal position of the focal marker.

`xlim` the range of marker positions.

`nhap` the number of haplotypes in the sample.

### See Also

[calc\\_furcation](#)

### Examples

```
# copy example files into working directory
make.example.files()
# read first example file
hh <- data2haplohh(hap_file = "example1.hap", map_file = "example1.map", allele_coding = "01")
# remove example files
remove.example.files()
# calculate furcation structure around marker "rs6"
f <- calc_furcation(hh, mrk = "rs6")
# extract left side tree of ancestral allele (which is coded by '0')
f[['0']]@left
# the tree consists of seven nodes, '1' being the root node
# nodes 2 and 3 have the root node as parent, etc.
# the first chromosome is attached as a label node to node 7, etc.
# For comparison, a plot of the complete furcation structure:
plot(f)
```

---

haplen-class	<i>class for haplotype length</i>
--------------	-----------------------------------

---

**Description**

class for haplotype length

---

haplohh-class	<i>Class "haplohh"</i>
---------------	------------------------

---

**Description**

An object of this class contains the information needed for computation of EHH based statistics.

**Usage**

```
## S4 method for signature 'haplohh'
chr.name(x)

## S4 method for signature 'haplohh'
positions(x)

## S4 method for signature 'haplohh'
haplo(x)

## S4 method for signature 'haplohh'
nmrk(x)

## S4 method for signature 'haplohh'
mrk.names(x)

## S4 method for signature 'haplohh'
nhap(x)

## S4 method for signature 'haplohh'
hap.names(x)
```

**Arguments**

x                    an object of this class.

**Details**

This class is the basis for all calculations done by this package. Note that the matrix in slot haplo has to be of type integer, not numeric. Objects built by versions of rehh up to 2.0.4 coded this matrix as numeric and used a different coding scheme. They can be converted e.g. by haplohh <-update\_haplohh(old\_haplohh) in order to be used with the present version.



**Slots**

`chr.name` name of the chromosome/scaffold to which the markers belong.

`positions` vector of type `numeric` containing the marker positions within the chromosome.

`haplo` matrix of type `integer` containing haplotypes in rows and markers in columns.

**See Also**

[data2haplohh](#), [update\\_haplohh](#)

**Examples**

```
showClass("haplohh")
```

---

`haplohh_cgu_bta12`      *Example of an haplohh object*

---

**Description**

The object contains haplotype data for 140 cattle individuals (280 haplotypes) belonging to the Creole breed from Guadeloupe (CGU) and 1424 markers (mapping to chromosome BTA12).

**Usage**

```
data(haplohh_cgu_bta12)
```

**Format**

An object of [haplohh-class](#).

**References**

Gautier, M. and Naves, M. (2011). Footprints of selection in the ancestral admixture of a New World Creole cattle breed. *Molecular Ecology*, **20**, 3128-3143.

**See Also**

[data2haplohh](#)

ies2xpehh

*Compute XP-EHH***Description**

Compute XP-EHH (standardized ratio of iES of two populations).

**Usage**

```
ies2xpehh(
  scan_pop1,
  scan_pop2,
  popname1 = NA,
  popname2 = NA,
  min_nhaplo = NA,
  standardize = TRUE,
  include_freq = FALSE,
  p.side = NA,
  p.adjust.method = "none",
  verbose = TRUE
)
```

**Arguments**

scan_pop1	a data frame with markers in rows and columns with chromosome name, position of the marker, frequency of the ancestral allele and iES as obtained by <a href="#">scan_hh</a> on the first population.
scan_pop2	a data frame with markers in rows and columns with chromosome name, position of the marker, frequency of the ancestral allele and iES as obtained by <a href="#">scan_hh</a> on the second population.
popname1	short ID/name of the first population; to be added to an output column name.
popname2	short ID/name of the second population; to be added to an output column name.
min_nhaplo	discard positions where in at least one of the populations fewer than min_nhaplo haplotypes have been evaluated (default NA).
standardize	logical. If TRUE (default), then standardize XP-EHH, else report unstandardized XP-EHH.
include_freq	logical. If TRUE include columns with allele frequencies into result.
p.side	side to which refers the p-value. Default NA, meaning two-sided. Can be set to "left" or "right".
p.adjust.method	method passed to function <a href="#">p.adjust</a> to correct the p-value for multiple testing. Default "none".
verbose	logical. If TRUE (default), report number of markers of the two source data frames and result data frame.

## Details

Log ratio of iES (population 1 over population 2) computed as described in Sabeti et al. (2007). Note that the two data frames are merged on the basis of chromosome and position. Marker names are kept, if they are identical and unique in both data frames.

Since the standardized XP-EHH values follow, if markers evolve predominantly neutrally, approximately a standard Gaussian distribution, it is practical to assign to the values a p-value relative to the null-hypothesis of neutral evolution. The parameter `p.side` determines if the p-value is assigned to both sides of the distribution or to one side of interest.

## Value

The returned value is a data frame with markers in rows and columns for chromosome name, marker position, XP-EHH and, if standardized, p-value in a negative log<sub>10</sub> scale. Optionally, allele frequencies are included.

## References

Gautier, M. and Naves, M. (2011). Footprints of selection in the ancestral admixture of a New World Creole cattle breed. *Molecular Ecology*, **20**, 3128-3143.

Sabeti, P.C. et al. (2007). Genome-wide detection and characterization of positive selection in human populations. *Nature*, **449**, 913-918.

## See Also

[scan\\_hh](#), [distribplot](#), [manhattanplot](#)

## Examples

```
library(rehh.data)
data(wgscan.cgu) ; data(wgscan.eut)
## results from a genome scan (44,057 SNPs)
##see ?wgscan.eut and ?wgscan.cgu for details
wgscan.xpehh <- ies2xpehh(wgscan.cgu, wgscan.eut, "CGU", "EUT")
```

---

ihh2ihs

*Compute iHS*

---

## Description

Compute iHS (standardized ratio of iHH values of two alleles).

**Usage**

```
ihh2ihs(
  scan,
  freqbin = 0.025,
  min_maf = 0.05,
  min_nhaplo = NA,
  standardize = TRUE,
  include_freq = FALSE,
  right = FALSE,
  alpha = 0.05,
  p.side = NA,
  p.adjust.method = "none",
  verbose = TRUE
)
```

**Arguments**

scan	a data frame with chromosome name, marker position, frequency of ancestral (resp. major) allele, frequency of derived (resp. minor) allele, and iHH for both alleles, as obtained from function <a href="#">scan_hh</a> .
freqbin	size of the bins to standardize $\log(iHH\_A/iHH\_D)$ . Markers are binned with respect to the derived allele frequency at the focal marker. The bins are built from <code>min_maf</code> to $1 - \text{min\_maf}$ in steps of size <code>freqbin</code> . If set to 0, standardization is performed considering each observed frequency as a discrete frequency class (useful in case of a large number of markers and few different haplotypes). If set to an integer of 1 or greater, a corresponding number of equally sized bins are created.
min_maf	focal markers with a MAF (Minor Allele Frequency) lower than or equal to <code>min_maf</code> are discarded from the analysis (default 0.05).
min_nhaplo	focal markers with least one of the two compared alleles carried by fewer than <code>min_nhaplo</code> haplotypes, are discarded (default NA).
standardize	logical. If TRUE (default), then standardize iHS, else report unstandardized iHS.
include_freq	logical. If TRUE include columns with allele frequencies into result.
right	logical. If TRUE the bin intervals are closed on the right (and open on the left).
alpha	calculate quantiles $\alpha/2$ and $(1-\alpha/2)$ for unstandardized binned iHS.
p.side	side to which refers the p-value. Default NA, meaning two-sided. Can be set to "left" or "right".
p.adjust.method	method passed to function <a href="#">p.adjust</a> to correct the p-value for multiple testing. Default "none".
verbose	logical. If TRUE (default), report number of markers of the source data frame and result data frame.

## Details

Computes log ratio of iHH of two focal alleles as described in Voight et al. (2006). The standardization is performed within each bins separately because of the frequency-dependence of expected iHS values under neutrality. An implicit assumption of this approach is that each bin is dominated by neutral markers.

Since the standardized iHS values follow, if markers evolve predominantly neutrally, approximately a standard Gaussian distribution, it is practical to assign to the values a p-value relative to the null-hypothesis of neutral evolution. The parameter `p.side` determines if the p-value is assigned to both sides of the distribution or to one side of interest.

## Value

The returned value is a list containing two elements

**ihs** a data frame with markers in rows and the columns for chromosome name, marker position, iHS and, if standardized, p-value in a negative log10 scale. Optionally, allele frequencies are included.

**frequency.class** a data frame with bins in rows and columns for the number of markers, mean uniHS, standard deviation uniHS, lower quantile uniHS, upper quantile uniHS.

## References

Gautier, M. and Naves, M. (2011). Footprints of selection in the ancestral admixture of a New World Creole cattle breed. *Molecular Ecology*, **20**, 3128-3143.

Voight, B.F. and Kudravalli, S. and Wen, X. and Pritchard, J.K. (2006). A map of recent positive selection in the human genome. *Plos Biology*, **4**, e72.

## See Also

[scan\\_hh](#), [distribplot](#), [freqbinplot](#), [manhattanplot](#)

## Examples

```
library(rehh.data)
data(wgscan.cgu)
#results from a genome scan (44,057 SNPs)
#see ?wgscan.eut and ?wgscan.cgu for details
wgscan.cgu.ihs <- ihs2ihs(wgscan.cgu)
```

---

ines2rsb

*Compute Rsb*

---

## Description

Compute Rsb (standardized ratio of inES of two populations).

**Usage**

```

ines2rsb(
  scan_pop1,
  scan_pop2,
  popname1 = NA,
  popname2 = NA,
  min_nhaplo = NA,
  standardize = TRUE,
  include_freq = FALSE,
  p.side = NA,
  p.adjust.method = "none",
  verbose = TRUE
)

```

**Arguments**

scan_pop1	a data frame with markers in rows and columns with chromosome name, position of the marker, frequency of the ancestral allele and inES as obtained by <a href="#">scan_hh</a> on the first population.
scan_pop2	a data frame with markers in rows and columns with chromosome name, position of the marker, frequency of the ancestral allele and inES as obtained by <a href="#">scan_hh</a> on the second population.
popname1	short ID/name of the first population; to be added to an output column name.
popname2	short ID/name of the second population; to be added to an output column name.
min_nhaplo	discard positions where in at least one of the populations fewer than <code>min_nhaplo</code> haplotypes have been evaluated (default NA).
standardize	logical. If TRUE (default), then standardize Rsb, else report unstandardized Rsb.
include_freq	logical. If TRUE include columns with allele frequencies into result.
p.side	side to which refers the p-value. Default NA, meaning two-sided. Can be set to "left" or "right".
p.adjust.method	method passed to function <a href="#">p.adjust</a> to correct the p-value for multiple testing. Default "none".
verbose	logical. If TRUE (default), report number of markers of the two source data frames and result data frame.

**Details**

Log ratio of inES (population 1 over population 2) computed as described in Tang et al. (2007). Note that the two data frames are merged on the basis of chromosome and position. Marker names are kept, if they are identical and unique in both data frames.

Since the standardized Rsb values follow, if markers evolve predominantly neutrally, approximately a standard Gaussian distribution, it is practical to assign to the values a p-value relative to the null-hypothesis of neutral evolution. The parameter `p.side` determines if the p-value is assigned to both sides of the distribution or to one side of interest.

**Value**

The returned value is a data frame with markers in rows and columns for chromosome name, marker position, Rsb and, if standardized, p-value in a negative log10 scale. Optionally, allele frequencies are included.

**References**

Gautier, M. and Naves, M. (2011). Footprints of selection in the ancestral admixture of a New World Creole cattle breed. *Molecular Ecology*, **20**, 3128-3143.

Tang, K. and Thornton, K.R. and Stoneking, M. (2007). A New Approach for Using Genome Scans to Detect Recent Positive Selection in the Human Genome. *Plos Biology*, **7**, e171.

**See Also**

[scan\\_hh](#), [distribplot](#), [manhattanplot](#)

**Examples**

```
library(rehh.data)
data(wgscan.cgu) ; data(wgscan.eut)
## results from a genome scan (44,057 SNPs)
##see ?wgscan.eut and ?wgscan.cgu for details
wgscan.rsb <- ines2rsb(wgscan.cgu, wgscan.eut, "CGU", "EUT")
```

---

make.example.files      *Copy example input files into current working directory*

---

**Description**

This function copies the following example files to the current working directory:

- example1.hap "example 1" haplotype file in "standard format"
- example1.map "example 1" marker information file
- example1.vcf "example 1" as vcf file
- example2.hap "example 2" haplotype file in "standard format"
- example2.map "example 2" marker information file
- example2.vcf "example 2" as vcf file
- ms.out output from a small simulation by the program 'ms'
- bta12\_cgu.hap an haplotype file in "standard format"
- bta12\_cgu.thap an haplotype file in "transposed format"
- bta12\_hapguess\_switch.out an haplotype file in fastphase output format
- map.inp a marker information file for all bta\_cgu markers

Example 1 was used in (Gautier 2017) to explain the various EHH derived statistics calculated by this package. Example 2 is an extension containing multi-allelic markers and missing values.

The bta12 files contain data for 280 haplotypes, originating from 140 individuals belonging to the Creole cattle breed from Guadeloupe, at 1.424 markers mapping to bovine chromosome 12 (BTA12) (Gautier 2011).

### Usage

```
make.example.files()
```

### References

Gautier, M. and Naves, M. (2011). Footprints of selection in the ancestral admixture of a New World Creole cattle breed. *Molecular Ecology*, **20**, 3128-3143.

Gautier, M., Klassmann, A. and Vitalis, R. (2017). rehh 2.0: a reimplement of the R package rehh to detect positive selection from haplotype structure. *Molecular Ecology Resources*, **17**, 78-90.

### See Also

[data2haplohh](#), [remove.example.files](#)

---

manhattanplot

*Manhattan plot of iHS, XP-EHH or Rsb over a genome.*

---

### Description

Manhattanplot of iHS, XP-EHH or Rsb over a genome.

### Usage

```
manhattanplot(  
  data,  
  pval = FALSE,  
  threshold = c(-2, 2),  
  chr.name = NA,  
  cr = NULL,  
  cr.col = "gray",  
  cr.opacity = 0.5,  
  cr.lab.cex = 0.6,  
  cr.lab.offset = 0,  
  cr.lab.pos = "top",  
  mrk = NULL,  
  mrk.cex = 1,  
  mrk.col = "gray",  
  mrk.pch = 1,  
  mrk.lab.cex = 0.4,  
  mrk.lab.pos = 4,  
)
```



```

    ignore_sign = FALSE,
    cex = 0.5,
    las = 1,
    pch = 20,
    resolution = NULL,
    ...
)

```

## Arguments

<code>data</code>	output of either <a href="#">ihh2ihs</a> , <a href="#">ies2xpehh</a> or <a href="#">ines2rsb</a> .
<code>pval</code>	logical. If TRUE, the p-value is plotted, otherwise the score itself.
<code>threshold</code>	a horizontal line is added at the corresponding value(s), for instance to represent a significance threshold. A single value (upper or lower threshold) or two values (upper and lower) can be specified.
<code>chr.name</code>	if NA (default), all chromosomes are plotted, otherwise only those specified.
<code>cr</code>	highlight "candidate regions" specified by a data.frame with columns CHR, START and END as obtained by the function <a href="#">calc_candidate_regions</a> .
<code>cr.col</code>	the color for highlighting
<code>cr.opacity</code>	a value between 0 (invisible) and 1 (opaque).
<code>cr.lab.cex</code>	text size of candidate region labels.
<code>cr.lab.offset</code>	offset of candidate region labels.
<code>cr.lab.pos</code>	if "top" (default) or "bottom", candidate regions are labeled by numbers; to turn off, use "none"
<code>mrk</code>	highlight marker specified by a data.frame containing the columns CHR and POSITION. The row names of that data frame are taken as labels. Alternatively a vector with marker IDs can be specified. In the latter case the ID is used as label.
<code>mrk.cex</code>	size of marker label.
<code>mrk.col</code>	color of the highlighted points.
<code>mrk.pch</code>	type of the highlighted points.
<code>mrk.lab.cex</code>	text size of marker label. If zero, no labels are printed.
<code>mrk.lab.pos</code>	a position specifier for the text. Values of 1, 2, 3 and 4, respectively indicate positions below, to the left of, above and to the right of the highlighted marker.
<code>ignore_sign</code>	logical. If TRUE, absolute values are plotted.
<code>cex</code>	size of the points representing markers in the plot(s) (see <a href="#">par</a> ).
<code>las</code>	orientation of axis labels (see <a href="#">par</a> ).
<code>pch</code>	type of the points representing markers in the plot(s) (see <a href="#">points</a> ).
<code>resolution</code>	Rasterize data points to the specified resolution and remove duplicate points. Defaults to NULL, i.e. no rasterization. A typical value might be <code>c(1E5, 0.01)</code> , meaning that resolution on the x-axis (chromosomal position) is 100000 and on the y-axis (score or p-value) is 0.01.
<code>...</code>	further arguments to be passed to <a href="#">plot.default</a> .

**Details**

The color of chromosomes is taken from the "Graphics Palette", see [palette](#).

If a single chromosome is plotted, a genomic region can be specified by argument `xlim`.

**Value**

The function returns a plot.

**See Also**

[ihh2ihs](#), [ies2xpehh](#), [ines2rsb](#), [calc\\_candidate\\_regions](#).

**Examples**

```
library(rehh.data)
data(wgscan.cgu)
## results from a genome scan (44,057 SNPs)
## see ?wgscan.eut and ?wgscan.cgu for details
wgscan.ihs <- ihh2ihs(wgscan.cgu)
manhattanplot(wgscan.ihs)
```

---

plot.ehh

*Plot EHH around a focal marker*


---

**Description**

Plot curve of EHH values around a focal marker.

**Usage**

```
## S3 method for class 'ehh'
plot(
  x,
  ylim = c(0, 1),
  type = "l",
  main = paste0("EHH around '", x$mrk.name, "'"),
  xlab = "Position",
  ylab = "Extended Haplotype Homozygosity",
  col = c("blue", "red", "violet", "orange"),
  mrk.col = "gray",
  bty = "n",
  legend = NA,
  legend.xy.coords = "automatic",
  ...
)
```

**Arguments**

x	data (output of <code>calc_ehh</code> ).
ylim	the y limits of the plot
type	plot type (see <code>codeplot.default</code> and <code>matplot</code> ). Type "s" or "S" yield both (the same) piecewise constant curve.
main	title for the plot (default NA, i.e. none).
xlab	title for the x-axis.
ylab	title for the y-axis.
col	color for the ancestral and derived alleles (respectively) curves.
mrk.col	color of the vertical line at the focal marker position.
bty	box type around plot (see <code>par</code> ).
legend	legend text.
legend.xy.coords	if "automatic" (default) places legend either top left or top right; if "none", no legend is drawn; otherwise the argument is passed to <code>legend</code> .
...	further arguments to be passed to function <code>matplot</code> .

**See Also**

[data2haplohh](#), [calc\\_ehh](#), [plot.ehhs](#), [scan\\_hh](#).

**Examples**

```
#example haplohh object (280 haplotypes, 1424 SNPs)
#see ?haplohh_cgu_bta12 for details
data(haplohh_cgu_bta12)
#computing EHH statistics for the marker "F1205400"
#which displays a strong signal of selection
ehh <- calc_ehh(haplohh_cgu_bta12, mrk = "F1205400")
plot(ehh)
```

---

plot.ehhs

*Plot EHHS around a focal marker*

---

**Description**

Plot curve of EHHS values around a focal marker.

**Usage**

```
## S3 method for class 'ehhs'
plot(
  x,
  nehhs = FALSE,
  ylim = c(0, 1),
  type = "l",
  main = paste0("EHHS around '", x$mrk.name, "'"),
  xlab = "Position",
  ylab = "Extended Haplotype Homozygosity per Site",
  bty = "n",
  mrk.col = "gray",
  ...
)
```

**Arguments**

x	data (output of <a href="#">calc_ehhs</a> ).
nehhs	logical. If TRUE, plot normalized EHHS.
ylim	the y limits of the plot
type	plot type (see <a href="#">codeplot.default</a> . Type "s" or "S" yield both (the same) piecewise constant curve.
main	title for the plot (default NA, i.e. none).
xlab	title for the x-axis.
ylab	title for the y-axis.
bty	box type around plot (see <a href="#">par</a> ).
mrk.col	color of the vertical line at the focal marker position.
...	further arguments to be passed to function <a href="#">plot.default</a> .

**See Also**

[data2haplohh](#), [plot.ehh](#), [calc\\_ehhs](#), [scan\\_hh](#).

**Examples**

```
#example haplohh object (280 haplotypes, 1424 SNPs)
#see ?haplohh_cgu_bta12 for details
data(haplohh_cgu_bta12)
#computing EHHS statistics for the marker "F1205400"
#which displays a strong signal of selection
ehhs <- calc_ehhs(haplohh_cgu_bta12, mrk = "F1205400")
plot(ehhs)
```

---

plot.furcation	<i>Plots furcation trees around a focal marker</i>
----------------	--

---

## Description

Plots furcation trees around a focal marker

## Usage

```
## S3 method for class 'furcation'
plot(
  x,
  allele = NA,
  col = c("blue", "red", "violet", "orange"),
  mrk.col = "gray",
  lwd = 0.1,
  hap.names = NULL,
  cex.lab = 1,
  family.lab = "",
  offset.lab = 0.5,
  legend = NA,
  legend.xy.coords = "automatic",
  ...
)
```

## Arguments

x	an object of class furcation (see <a href="#">calc_furcation</a> ).
allele	If NA (default), furcation trees for all alleles of the focal marker are plotted, otherwise for the specified alleles. Alleles must be specified by their internal coding, i.e. '0' for ancestral resp. major allele, etc.
col	color for each allele (as coded internally).
mrk.col	color of the vertical line at the focal marker position.
lwd	controls the relative width of the diagram lines on the plot (default 0.1).
hap.names	a vector containing names of chromosomes.
cex.lab	relative size of labels. See <a href="#">par</a> .
family.lab	font family for labels. See <a href="#">par</a> .
offset.lab	offset of labels. See <a href="#">par</a> .
legend	legend text.
legend.xy.coords	if "automatic" (default) places legend either top left or top right; if "none", no legend is drawn; otherwise argument is passed to <a href="#">legend</a> .
...	other arguments to be passed to <a href="#">plot.default</a> .

**See Also**

[plot.haplen](#).

**Examples**

```
#example haplohh object (280 haplotypes, 1424 SNPs)
#see ?haplohh_cgu_bta12 for details
data(haplohh_cgu_bta12)
#plotting furcation diagram for both ancestral and derived allele
#from the marker "F1205400"
#which display a strong signal of selection
f <- calc_furcation(haplohh_cgu_bta12, mrk = "F1205400")
plot(f)
plot(f, xlim = c(2e+07,3.5e+07))
plot(f, xlim = c(2.7e+07,3.1e+07))
plot(f, xlim = c(2.7e+07,3.1e+07), hap.names = hap.names(haplohh_cgu_bta12), cex.lab=0.3)
```

---

plot.haplen

*Plot the length of extended haplotypes around a focal marker*

---

**Description**

Plot the length of extended haplotype around a focal marker.

**Usage**

```
## S3 method for class 'haplen'
plot(
  x,
  allele = NA,
  group_by_allele = TRUE,
  order_by_length = FALSE,
  col = c("blue", "red", "violet", "orange"),
  mrk.col = "gray",
  lwd = 1,
  hap.names = NULL,
  cex.lab = 1,
  family.lab = "",
  offset.lab = 0.5,
  pos.lab = "left",
  legend = NA,
  legend.xy.coords = "automatic",
  ...
)
```

**Arguments**

x	an object of class haplen generated by <a href="#">calc_haplen</a> .
allele	if NA (default), haplotypes of all alleles are plotted, otherwise for the specified alleles. Alleles must be specified by their internal coding, i.e. '0' for ancestral resp. major allele, etc.
group_by_allele	logical. If TRUE (default), group chromosomes by their allele at the focal marker; alleles are ordered by their internal coding unless parameter alleles is specified. If FALSE, haplotypes are drawn by their order in the input file.
order_by_length	if TRUE, chromosomes are ordered by their shared haplotype length.
col	color for each allele (as coded internally).
mrk.col	color of the vertical line at the focal marker position.
lwd	line width.
hap.names	a vector containing the names of chromosomes.
cex.lab	relative letter size of labels. See <a href="#">par</a> .
family.lab	font family for labels. See <a href="#">par</a> .
offset.lab	offset of labels. See <a href="#">par</a> .
pos.lab	position of haplotype labels. Either "left", "right" or "both".
legend	legend text.
legend.xy.coords	if "automatic" (default) places legend either top left or top right; if "none", no legend is drawn; otherwise argument is passed to <a href="#">legend</a> .
...	other parameters to be passed to <a href="#">plot.default</a> .

**See Also**

[calc\\_haplen](#), [plot.furcation](#).

**Examples**

```
#example haplohh object (280 haplotypes, 1424 SNPs)
#see ?haplohh_cgu_bta12 for details
data(haplohh_cgu_bta12)
#plotting length of extended haplotypes for both ancestral and derived allele
#of the marker "F1205400"
#which displays a strong signal of selection
f <- calc_furcation(haplohh_cgu_bta12, mrk = "F1205400")
h <- calc_haplen(f)
plot(h)
plot(h, hap.names = hap.names(haplohh_cgu_bta12), cex.lab = 0.3)
```

---

`plot.haplohh`*Plot the variants of a haplohh object*

---

### Description

Plot the variants of a haplohh object. This method is intended for visualization of very small data sets such as the examples provided by the package.

### Usage

```
## S3 method for class 'haplohh'
plot(
  x,
  mrk = NA,
  allele = NA,
  group_by_allele = FALSE,
  ignore.distances = FALSE,
  col = c("blue", "red", "violet", "orange"),
  linecol = "gray",
  mrk.col = "gray",
  pch = 19,
  cex = 1,
  lwd = 1,
  hap.names = NULL,
  mrk.names = NULL,
  cex.lab.hap = 0.8,
  cex.lab.mrk = 0.8,
  family.lab = "",
  offset.lab.hap = 0.5,
  offset.lab.mrk = 0.25,
  pos.lab.hap = "left",
  pos.lab.mrk = "top",
  srt.hap = 0,
  srt.mrk = 0,
  ...
)
```

### Arguments

<code>x</code>	an object of class haplo-hh generated by <a href="#">data2haplohh</a> .
<code>mrk</code>	the focal marker. Used only, if alleles are grouped or (de-)selected.
<code>allele</code>	if NA (default), haplotypes of all alleles are plotted, otherwise for the specified alleles. Alleles must be specified by their internal coding, i.e. '0' for ancestral resp. major allele, etc. Haplotypes with missing values at the focal marker can only be selected in combination with genotyped alleles, e.g. as <code>c(1, NA)</code> .



group_by_allele	logical. If TRUE, group chromosomes by their allele at the focal marker; alleles are ordered by their internal coding unless parameter alleles is specified. If FALSE (default), haplotypes are drawn by their order in the input file.
ignore.distances	logical. If TRUE, markers are drawn equally-spaced.
col	color for each allele (as coded internally).
linecol	the color of the background lines. If more than one color is specified and sequences sorted by the marker allele, the specified colors are used to distinguish the alleles; otherwise consecutive sequences are set into the specified colors.
mrk.col	color of the vertical line at the focal marker position.
pch	symbol used for markers. See <a href="#">points</a> .
cex	relative size of marker symbol. See <a href="#">points</a> .
lwd	line width.
hap.names	a vector containing the names of chromosomes.
mrk.names	a vector containing the names of markers.
cex.lab.hap	relative letter size of haplotype labels. See <a href="#">par</a> .
cex.lab.mrk	relative letter size of marker labels. See <a href="#">par</a> .
family.lab	font family for labels. See <a href="#">par</a> .
offset.lab.hap	offset of haplotype labels. See <a href="#">par</a> .
offset.lab.mrk	offset of marker labels. See <a href="#">par</a> .
pos.lab.hap	position of haplotype labels. Either "left" (default), "right", "none" or "both".
pos.lab.mrk	position of marker labels. Either "top" (default) or "none".
srt.hap	rotation of haplotype labels (see <a href="#">par</a> ).
srt.mrk	rotation of marker labels (see <a href="#">par</a> ).
...	other parameters to be passed to <a href="#">plot.default</a> .

### Details

Specifying a haplohh-object with more than 4096 haplotypes or markers produces an error.

### See Also

[calc\\_haplen](#), [plot.furcation](#).

### Examples

```
#example haplohh object
make.example.files()
hh <- data2haplohh(hap_file = "example1.hap",
                  map_file = "example1.map",
                  allele_coding = "01")

plot(hh)
hh <- data2haplohh(hap_file = "example2.hap",
```

```

        map_file = "example2.map",
        allele_coding = "01",
        min_perc_genomark = 50)
plot(hh)
remove.example.files()

```

---

`remove.example.files` *Remove example files from current working directory.*

---

### Description

Remove example files from current working directory.

### Usage

```
remove.example.files()
```

### Details

Removes the files created by `make.example.files()`. No error is thrown, if files do not exist.

### See Also

[make.example.files](#)

---

`scan_hh` *Compute iHH, iES and inES over a whole chromosome*

---

### Description

Compute integrated EHH (iHH), integrated EHHS (iES) and integrated normalized EHHS (inES) for all markers of a chromosome (or linkage group).

### Usage

```

scan_hh(
  haplohh,
  limhaplo = 2,
  limhomohaplo = 2,
  limehh = 0.05,
  limehhs = 0.05,
  phased = TRUE,
  polarized = TRUE,
  scalegap = NA,
  maxgap = NA,
  discard_integration_at_border = TRUE,

```

```

    lower_ehh_y_bound = limehh,
    lower_ehhs_y_bound = limehhs,
    interpolate = TRUE,
    threads = 1
)

```

## Arguments

haplohh	an object of class haplohh (see <a href="#">data2haplohh</a> )
limhaplo	if there are less than limhaplo chromosomes that can be used for the calculation of EHH(S), the calculation is stopped. The option is intended for the case of missing data, which leads to the successive exclusion of haplotypes: the further away from the focal marker the less haplotypes contribute to EHH(S).
limhomohaplo	if there are less than limhomohaplo homozygous chromosomes, the calculation is stopped. This option is intended for unphased data and should be invoked only if relatively low frequency variants are not filtered subsequently (see main vignette and Klassmann et al. 2020).
limehh	limit at which EHH stops to be evaluated.
limehhs	limit at which EHHS stops to be evaluated.
phased	logical. If TRUE (default) chromosomes are expected to be phased. If FALSE, the haplotype data is assumed to consist of pairwise ordered chromosomes belonging to diploid individuals. EHH(S) is then estimated over individuals which are homozygous at the focal marker.
polarized	logical. TRUE by default. If FALSE, use major and minor allele instead of ancestral and derived. If there are more than two alleles then the minor allele refers to the second-most frequent allele.
scalegap	scale or cap gaps larger than the specified size to the specified size (default=NA, i.e. no scaling).
maxgap	maximum allowed gap in bp between two markers. If exceeded, further calculation of EHH(S) is stopped at the gap (default=NA, i.e. no limitation).
discard_integration_at_border	logical. If TRUE (default) and computation reaches first or last marker or a gap larger than maxgap, iHH, iES and inES are set to NA.
lower_ehh_y_bound	lower y boundary of the area to be integrated over (default: limehh). Can be set to zero for compatibility with the program hapbin.
lower_ehhs_y_bound	lower y boundary of the area to be integrated (default: limehhs). Can be set to zero for compatibility with the program hapbin.
interpolate	logical. If TRUE (default), integration is performed over a continuous EHH(S) curve (values are interpolated linearly between consecutive markers), otherwise the EHH(S) curve decreases stepwise at markers.
threads	number of threads to parallelize computation

## Details

Integrated EHH (iHH), integrated EHHS (iES) and integrated normalized EHHS (inES) are computed for all markers of the chromosome (or linkage group). This function is several times faster as a procedure calling in turn `calc_ehh` and `calc_ehhs` for all markers. To perform a whole genome-scan this function needs to be called for each chromosome and results concatenated.

Note that setting `limehh` or `limehhs` to zero is likely to reduce power, since even under neutrality a tiny fraction ( $\ll 0.05$ ) of extremely long shared haplotypes is expected which, if fully accounted for, would obfuscate the signal at selected sites.

## Value

The returned value is a dataframe with markers in rows and the following columns

1. chromosome name
2. position in the chromosome
3. sample frequency of the ancestral / major allele
4. sample frequency of the second-most frequent remaining allele
5. number of evaluated haplotypes at the focal marker for the ancestral / major allele
6. number of evaluated haplotypes at the focal marker for the second-most frequent remaining allele
7. iHH of the ancestral / major allele
8. iHH of the second-most frequent remaining allele
9. iES (used by Sabeti et al 2007)
10. inES (used by Tang et al 2007)

Note that in case of unphased data the evaluation is restricted to haplotypes of homozygous individuals which reduces the power to detect selection, particularly for iHS (for appropriate parameter setting see the main vignette and Klassmann et al (2020)).

## References

- Gautier, M. and Naves, M. (2011). Footprints of selection in the ancestral admixture of a New World Creole cattle breed. *Molecular Ecology*, **20**, 3128-3143.
- Klassmann, A. et al. (2020). Detecting selection using Extended Haplotype Homozygosity (EHH)-based statistics on unphased or unpolarized data. (submitted).
- Sabeti, P.C. et al. (2002). Detecting recent positive selection in the human genome from haplotype structure. *Nature*, **419**, 832-837.
- Sabeti, P.C. et al. (2007). Genome-wide detection and characterization of positive selection in human populations. *Nature*, **449**, 913-918.
- Tang, K. and Thornton, K.R. and Stoneking, M. (2007). A New Approach for Using Genome Scans to Detect Recent Positive Selection in the Human Genome. *Plos Biology*, **7**, e171.
- Voight, B.F. and Kudravalli, S. and Wen, X. and Pritchard, J.K. (2006). A map of recent positive selection in the human genome. *Plos Biology*, **4**, e72.

**See Also**

[data2haplohh](#), [calc\\_ehh](#), [calc\\_ehhs](#) [ihh2ihs](#), [ines2rsb](#), [ies2xpehh](#)

**Examples**

```
#example haplohh object (280 haplotypes, 1424 SNPs)
#see ?haplohh_cgu_bta12 for details
data(haplohh_cgu_bta12)
scan <- scan_hh(haplohh_cgu_bta12)
```

---

scan_hh_full	<i>Compute iHH, iES and inES over a whole chromosome without cut-offs</i>
--------------	---

---

**Description**

Compute integrated EHH (iHH), integrated EHHS (iES) and integrated normalized EHHS (inES) for all markers of a chromosome (or linkage group). This function computes the statistics by a slightly different algorithm than [scan\\_hh](#): it sidesteps the calculation of EHH and EHHS values and their subsequent integration and consequently no cut-offs relying on these values can be specified. Instead it computes the full lengths of pairwise shared haplotypes and averages them afterwards.

This function is (as yet) exclusively intended for the study of general properties of these statistics using simulated data. The omission of all cut-offs is not recommended for a scan on experimental data.

**Usage**

```
scan_hh_full(
  haplohh,
  phased = TRUE,
  polarized = TRUE,
  maxgap = NA,
  discard_integration_at_border = TRUE,
  geometric.mean = FALSE,
  threads = 1
)
```

**Arguments**

haplohh	an object of class haplohh (see <a href="#">data2haplohh</a> )
phased	logical. If TRUE (default) chromosomes are expected to be phased. If FALSE, the haplotype data is assumed to consist of pairwise ordered chromosomes belonging to diploid individuals. EHH(S) is then estimated over individuals which are homozygous at the focal marker.
polarized	logical. TRUE by default. If FALSE, use major and minor allele instead of ancestral and derived. If there are more than two alleles then the minor allele refers to the second-most frequent allele.

maxgap	maximum allowed gap in bp between two markers. If exceeded, further calculation of EHH(S) is stopped at the gap (default=NA, i.e no limitation).
discard_integration_at_border	logical. If TRUE (default) and computation of any of the statistics reaches first or last marker or a gap larger than maxgap, iHH, iES and inES are set to NA.
geometric.mean	logical. If FALSE (default), the standard arithmetic mean is used to average shared haplotype lengths. If TRUE the geometric mean is used instead (IES values are undefined in this case). Note that usage of the geometric mean has not yet been studied formally and should be considered experimental!
threads	number of threads to parallelize computation

### Details

Integrated EHH (iHH), integrated EHHS (iES) and integrated normalized EHHS (inES) are computed for all markers of the chromosome (or linkage group). This function sidesteps the computation of EHH and EHHS values and their stepwise integration. Instead, the length of all shared haplotypes is computed and afterwards averaged. In the absence of missing values the statistics are identical to those calculated by `scan_hh` with settings `limehh = 0`, `limehhs = 0` and `interpolate = FALSE`, yet this function is faster. The former two settings are however not recommended for the application on experimental data (see vignette).

If `discard_integration_at_border` is set to TRUE and the extension of shared haplotypes reaches a border (i.e. chromosomal boundaries or a gap larger than maxgap), this function discards all statistics, while `scan_hh` handles each statistic independently.

`scan_hh` "removes" chromosomes with missing values from further calculations, while this function treats each missing value as a different allele. This yields a somewhat faster decay of all statistics with respect to the distance to the focal marker.

### Value

The returned value is a dataframe with markers in rows and the following columns

1. chromosome name
2. position in the chromosome
3. sample frequency of the ancestral / major allele
4. sample frequency of the second-most frequent remaining allele
5. number of evaluated haplotypes at the focal marker for the ancestral / major allele
6. number of evaluated haplotypes at the focal marker for the second-most frequent remaining allele
7. iHH of the ancestral / major allele
8. iHH of the second-most frequent remaining allele
9. iES (used by Sabeti et al 2007)
10. inES (used by Tang et al 2007)

Note that in case of unphased data the evaluation is restricted to haplotypes of homozygous individuals which reduces the power to detect selection, particularly for iHS (for appropriate parameter setting see the main vignette and Klassmann et al (2020)).

## References

- Gautier, M. and Naves, M. (2011). Footprints of selection in the ancestral admixture of a New World Creole cattle breed. *Molecular Ecology*, **20**, 3128-3143.
- Klassmann A., Vitalis R., and Gautier M. Detecting selection using Extended Haplotype Homozygosity (EHH)-based statistics on unphased or unpolarized data. Preprint. <https://doi.org/10.22541/au.158584282.24875401>.
- Sabeti, P.C. et al. (2002). Detecting recent positive selection in the human genome from haplotype structure. *Nature*, **419**, 832-837.
- Sabeti, P.C. et al. (2007). Genome-wide detection and characterization of positive selection in human populations. *Nature*, **449**, 913-918.
- Tang, K. and Thornton, K.R. and Stoneking, M. (2007). A New Approach for Using Genome Scans to Detect Recent Positive Selection in the Human Genome. *Plos Biology*, **7**, e171.
- Voight, B.F. and Kudaravalli, S. and Wen, X. and Pritchard, J.K. (2006). A map of recent positive selection in the human genome. *Plos Biology*, **4**, e72.

## See Also

[data2haplohh](#), [codescan\\_hh](#), [ihh2ihs](#), [ines2rsb](#), [ies2xpehh](#)

## Examples

```
#example haplohh object (280 haplotypes, 1424 SNPs)
#see ?haplohh_cgu_bta12 for details
data(haplohh_cgu_bta12)
scan <- scan_hh_full(haplohh_cgu_bta12)
```

---

subset.haplohh	<i>Subsets object of haplohh-class</i>
----------------	--

---

## Description

Subsets the data of an object of class [haplohh-class](#), meeting certain conditions.

## Usage

```
## S3 method for class 'haplohh'
subset(
  x,
  select.hap = NULL,
  select.mrk = NULL,
  min_perc_genom.hap = NA,
  min_perc_genom.mrk = 100,
  min_maf = NA,
  verbose = TRUE,
  ...
)
```

**Arguments**

<code>x</code>	object of class <a href="#">haplohh-class</a> to be subset.
<code>select.hap</code>	expression, indicating haplotypes to select.
<code>select.mrk</code>	expression, indicating markers to select.
<code>min_percgeno.hap</code>	threshold on percentage of missing data for haplotypes (haplotypes with less than <code>min_percgeno.hap</code> percent of markers genotyped are discarded). Default is NA, hence no constraint.
<code>min_percgeno.mrk</code>	threshold on percentage of missing data for markers (markers genotyped on less than <code>min_percgeno.mrk</code> percent of haplotypes are discarded). By default, <code>min_percgeno.mrk=100</code> , hence only fully genotyped markers are retained. This value cannot be set to NA or zero.
<code>min_maf</code>	threshold on the Minor Allele Frequency. Markers having a MAF lower than or equal to <code>minmaf</code> are discarded. In case of multi-allelic markers the second-most frequent allele is referred to as minor allele. Setting this value to zero eliminates monomorphic sites. Default is NA, hence no constraint.
<code>verbose</code>	logical. If TRUE (default), report verbose progress.
<code>...</code>	further arguments are ignored.

**See Also**

[haplohh-class](#), [data2haplohh](#)

**Examples**

```
#example haplohh object (280 haplotypes, 1424 SNPs)
#see ?haplohh_cgu_bta12 for details
data(haplohh_cgu_bta12)
#select subset of first 10 haplotypes and first 5 markers
subset(haplohh_cgu_bta12, select.hap = 1:10, select.mrk = 1:5)
```

---

update_haplohh	<i>Update object of class haplohh</i>
----------------	---------------------------------------

---

**Description**

Update object of class [haplohh-class](#) constructed by rehh versions up to version 2.0.4.

**Usage**

```
update_haplohh(haplohh)
```

**Arguments**

haplohh	an object of an old version of <a href="#">haplohh-class</a> .
---------	--



**Details**

This function is intended to update haplohh objects that have been built by rehh versions up to 2.0.4. These objects cannot be used in functions of the current version. The following changes have been made to the class definition: The internal representation of the haplotype matrix followed the encoding

- 0 missing value
- 1 ancestral allele
- 2 derived allele

and has been replaced by a vcf-like encoding:

- NA missing value
- 0 ancestral allele
- 1 derived allele.

Furthermore the slots nsnp, snp.name and nhap have been removed and slot position renamed to positions. An update of an old haplohh object is done as follows:

```
new_haplohh = update_haplohh(old_haplohh).
```

**See Also**

[haplohh-class](#), [data2haplohh](#).

# Index

- \* **datasets**
  - haplohh\_cgu\_bta12, 25
- allelefurcation
  - (allelefurcation-class), 4
- allelefurcation-class, 4
- as.newick, 4
  
- calc\_candidate\_regions, 5, 15, 16, 20, 33, 34
- calc\_ehh, 6, 11, 35, 45
- calc\_ehhs, 8, 9, 36, 45
- calc\_furcation, 5, 11, 13, 23, 37
- calc\_haplen, 12, 13, 14, 39, 41
- calc\_pairwise\_haplen, 14
- calc\_region\_stats, 6, 15
- chr.name (haplohh-class), 24
- chr.name, (haplohh-class), 24
- chr.name, haplohh-method (haplohh-class), 24
  
- data2haplohh, 7–9, 11, 12, 14, 15, 16, 25, 32, 35, 36, 40, 43, 45, 47–49
- distribplot, 19, 27, 29, 31
  
- ehh (calc\_ehh), 6
- ehh-class (calc\_ehh), 6
- ehhs (calc\_ehhs), 9
- ehhs-class (calc\_ehhs), 9
- extract\_regions, 20
  
- freqbinplot, 21, 29
- ftree, 4
- ftree (ftree-class), 22
- ftree-class, 22
- furcation, 4
- furcation (furcation-class), 23
- furcation-class, 23
  
- hap.names (haplohh-class), 24
- hap.names, (haplohh-class), 24
  
- hap.names, haplohh-method (haplohh-class), 24
- haplen (haplen-class), 24
- haplen-class, 24
- haplo (haplohh-class), 24
- haplo, (haplohh-class), 24
- haplo, haplohh-method (haplohh-class), 24
- haplohh-class, 24, 47
- haplohh\_cgu\_bta12, 25
  
- ies2xpehh, 5, 15, 20, 26, 33, 34, 45, 47
- ihh2ihs, 5, 15, 18, 20, 22, 27, 33, 34, 45, 47
- ines2rsb, 5, 15, 20, 29, 33, 34, 45, 47
  
- legend, 35, 37, 39
  
- make.example.files, 31, 42
- manhattanplot, 20, 27, 29, 31, 32
- matplot, 35
- mrk.names (haplohh-class), 24
- mrk.names, (haplohh-class), 24
- mrk.names, haplohh-method (haplohh-class), 24
  
- nhap (haplohh-class), 24
- nhap, (haplohh-class), 24
- nhap, haplohh-method (haplohh-class), 24
- nmrk (haplohh-class), 24
- nmrk, (haplohh-class), 24
- nmrk, haplohh-method (haplohh-class), 24
  
- p.adjust, 26, 28, 30
- palette, 34
- par, 33, 35–37, 39, 41
- plot.default, 20, 33, 35–37, 39, 41
- plot.ehh, 8, 34, 36
- plot.ehhs, 11, 35, 35
- plot.furcation, 5, 12, 37, 39, 41
- plot.haplen, 38, 38
- plot.haplohh, 40
- points, 33, 41

positions (haplohh-class), [24](#)  
positions, (haplohh-class), [24](#)  
positions, haplohh-method  
    (haplohh-class), [24](#)

rehh (rehh-package), [3](#)  
rehh-package, [3](#)  
remove.example.files, [32](#), [42](#)

scan\_hh, [8](#), [11](#), [20](#), [26–31](#), [35](#), [36](#), [42](#), [45–47](#)  
scan\_hh\_full, [15](#), [45](#)  
subset.haplohh, [47](#)

update\_haplohh, [25](#), [48](#)