Package 'numbat'

January 11, 2023

Title Haplotype-Aware CNV Analysis from scRNA-Seq

URL https://github.com/kharchenkolab/numbat

Version 1.2.1

Description A computational method that infers copy number variations (CNVs) in cancer scRNA-seq data and reconstructs the tumor phylogeny. 'numbat' integrates signals from gene expression, allelic ratio, and population haplotype structures to accurately infer allele-specific CNVs in single cells and reconstruct their lineage relationship. 'numbat' can be used to: 1. detect allele-specific copy number variations from single-cells; 2. differentiate tumor versus normal cells in the tumor microenvironment; 3. infer the clonal architecture and evolutionary history of profiled tumors. 'numbat' does not require tumor/normal-paired DNA or genotype data, but operates solely on the donor scRNA-data data (for example, 10x Cell Ranger output). Additional examples and documentations are available at <https://kharchenkolab.github.io/numbat/>. For details on the method please see Gao et al. Nature Biotechnology (2022) <doi:10.1038/s41587-022-01468-y>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Depends R (>= 4.1.0), Matrix

biocViews

Imports ape, caTools, data.table, dendextend, dplyr, GenomicRanges, ggplot2, ggraph, ggtree, glue, igraph, IRanges, logger, magrittr, methods, optparse, parallel, parallelDist, patchwork, pryr, purrr, Rcpp, RhpcBLASctl, R.utils, scales, scistreer (>= 1.1.0), stats4, stringr, tibble, tidygraph, tidyr, vcfR, zoo

Suggests ggrastr, ggrepel, knitr, matrixStats, testthat (>= 3.0.0),

Config/testthat/edition 3

LinkingTo Rcpp, RcppArmadillo, roptim

NeedsCompilation yes

SystemRequirements C++11, GNU make

Author Teng Gao [cre, aut],
Hirak Sarkar [aut],
Evan Biederstedt [aut],
Peter Kharchenko [aut]

Maintainer Teng Gao <tgao@g.harvard.edu>

RoxygenNote 7.2.1

Repository CRAN

Date/Publication 2023-01-11 00:20:02 UTC

R topics documented:

`acen_hg19` *centromere regions (hg19)*

Description

centromere regions (hg19)

Usage

```
acen_hg19
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 22 rows and 3 columns.

`acen_hg38` *centromere regions (hg38)*

Description

centromere regions (hg38)

Usage

```
acen_hg38
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 22 rows and 3 columns.

aggregate_counts *Utility function to make reference gene expression profiles*

Description

Utility function to make reference gene expression profiles

Usage

```
aggregate_counts(count_mat, annot, normalized = TRUE, verbose = TRUE)
```

Arguments

count_mat	matrix/dgCMatrix Gene expression counts
annot	dataframe Cell annotation with columns "cell" and "group"
normalized	logical Whether to return normalized expression values
verbose	logical Verbosity

Value

matrix Reference gene expression levels

Examples

```
ref_custom = aggregate_counts(count_mat_ref, annot_ref, verbose = FALSE)
```

analyze_bulk *Call CNVs in a pseudobulk profile using the Numbat joint HMM*

Description

Call CNVs in a pseudobulk profile using the Numbat joint HMM

Usage

```
analyze_bulk(
  bulk,
  t = 1e-05,
  gamma = 20,
  theta_min = 0.08,
  logphi_min = 0.25,
  nu = 1,
  min_genes = 10,
  exp_only = FALSE,
  allele_only = FALSE,
```

```
  bal_cnv = TRUE,
  retest = TRUE,
  find_diploid = TRUE,
  diploid_chroms = NULL,
  segs_loh = NULL,
  classify_allele = FALSE,
  run_hmm = TRUE,
  prior = NULL,
  exclude_neu = TRUE,
  phasing = TRUE,
  verbose = TRUE
)
```

Arguments

bulk	dataframe Pesudobulk profile
t	numeric Transition probability
gamma	numeric Dispersion parameter for the Beta-Binomial allele model
theta_min	numeric Minimum imbalance threshold
logphi_min	numeric Minimum log expression deviation threshold
nu	numeric Phase switch rate
min_genes	integer Minimum number of genes to call an event
exp_only	logical Whether to run expression-only HMM
allele_only	logical Whether to run allele-only HMM
bal_cnv	logical Whether to call balanced amplifications/deletions
retest	logical Whether to retest CNVs after Viterbi decoding
find_diploid	logical Whether to run diploid region identification routine
diploid_chroms	character vector User-given chromosomes that are known to be in diploid state
segs_loh	dataframe Segments with clonal LOH to be excluded
classify_allele	logical Whether to only classify allele (internal use only)
run_hmm	logical Whether to run HMM (internal use only)
prior	numeric vector Prior probabilities of states (internal use only)
exclude_neu	logical Whether to exclude neutral segments from retesting (internal use only)
phasing	logical Whether to use phasing information (internal use only)
verbose	logical Verbosity

Value

a pseudobulk profile dataframe with called CNV information

Examples

```
bulk_analyzed = analyze_bulk(bulk_example, t = 1e-5, find_diploid = FALSE, retest = FALSE)
```

annotate_genes *Annotate genes on allele dataframe*

Description

Annotate genes on allele dataframe

Usage

```
annotate_genes(df, gtf)
```

Arguments

df	dataframe Allele count dataframe
gtf	dataframe Gene gtf

Value

dataframe Allele dataframe with gene column

annot_ref *example reference cell annotation*

Description

example reference cell annotation

Usage

```
annot_ref
```

Format

An object of class `data.frame` with 50 rows and 2 columns.

bulk_example *example pseudobulk dataframe*

Description

example pseudobulk dataframe

Usage

```
bulk_example
```

Format

An object of class `tbl_df` (inherits from `tbl`, `data.frame`) with 3935 rows and 83 columns.

chrom_sizes_hg19 *chromosome sizes (hg19)*

Description

chromosome sizes (hg19)

Usage

```
chrom_sizes_hg19
```

Format

An object of class `data.table` (inherits from `data.frame`) with 22 rows and 2 columns.

chrom_sizes_hg38 *chromosome sizes (hg38)*

Description

chromosome sizes (hg38)

Usage

```
chrom_sizes_hg38
```

Format

An object of class `data.table` (inherits from `data.frame`) with 22 rows and 2 columns.

cnv_heatmap *Plot CNV heatmap*

Description

Plot CNV heatmap

Usage

```
cnv_heatmap(
  segs,
  var = "group",
  label_group = TRUE,
  legend = TRUE,
  exclude_gap = TRUE,
  genome = "hg38"
)
```

Arguments

segs	dataframe Segments to plot. Need columns "seg_start", "seg_end", "cnv_state"
var	character Column to facet by
label_group	logical Label the groups
legend	logical Display the legend
exclude_gap	logical Whether to mark gap regions
genome	character Genome build, either 'hg38' or 'hg19'

Value

ggplot Heatmap of CNVs along the genome

Examples

```
p = cnv_heatmap(segs_example)
```

count_mat_example *example gene expression count matrix*

Description

example gene expression count matrix

Usage

```
count_mat_example
```

Format

An object of class dgCMatrix with 1024 rows and 173 columns.

`count_mat_ref` *example reference count matrix*

Description

example reference count matrix

Usage

```
count_mat_ref
```

Format

An object of class dgCMatrix with 1000 rows and 50 columns.

`detect_clonal_loh` *Call clonal LOH using SNP density. Rcommended for cell lines or tumor samples with no normal cells.*

Description

Call clonal LOH using SNP density. Rcommended for cell lines or tumor samples with no normal cells.

Usage

```
detect_clonal_loh(bulk, t = 1e-05, min_depth = 0)
```

Arguments

bulk	dataframe Pseudobulk profile
t	numeric Transition probability
min_depth	integer Minimum coverage to filter SNPs

Value

dataframe LOH segments

Examples

```
segs_loh = detect_clonal_loh(bulk_example)
```

df_allele_example *example allele count dataframe*

Description

example allele count dataframe

Usage

```
df_allele_example
```

Format

An object of class `data.frame` with 41167 rows and 11 columns.

gaps_hg19 *genome gap regions (hg19)*

Description

genome gap regions (hg19)

Usage

```
gaps_hg19
```

Format

An object of class `data.table` (inherits from `data.frame`) with 28 rows and 3 columns.

gaps_hg38 *genome gap regions (hg38)*

Description

genome gap regions (hg38)

Usage

```
gaps_hg38
```

Format

An object of class `data.table` (inherits from `data.frame`) with 30 rows and 3 columns.

get_bulk *Aggregate single-cell data into combined bulk expression and allele profile*

Description

Aggregate single-cell data into combined bulk expression and allele profile

Usage

```
get_bulk(
  count_mat,
  lambdas_ref,
  df_allele,
  gtf,
  subset = NULL,
  min_depth = 0,
  nu = 1,
  verbose = TRUE
)
```

Arguments

count_mat	dgCMatrix Gene expression counts
lambdas_ref	matrix Reference expression profiles
df_allele	dataframe Single-cell allele counts
gtf	dataframe Transcript gtf
subset	vector Subset of cells to aggregate
min_depth	integer Minimum coverage to filter SNPs
nu	numeric Phase switch rate
verbose	logical Verbosity

Value

dataframe Pseudobulk gene expression and allele profile

Examples

```
bulk_example = get_bulk(
    count_mat = count_mat_example,
    lambdas_ref = ref_hca,
    df_allele = df_allele_example,
    gtf = gtf_hg38)
```

gexp_roll_example *example smoothed gene expression dataframe*

Description

example smoothed gene expression dataframe

Usage

```
gexp_roll_example
```

Format

An object of class `data.frame` with 10 rows and 2000 columns.

gtf_hg19 *gene model (hg19)*

Description

gene model (hg19)

Usage

```
gtf_hg19
```

Format

An object of class `data.table` (inherits from `data.frame`) with 26841 rows and 5 columns.

gtf_hg38 *gene model (hg38)*

Description

gene model (hg38)

Usage

```
gtf_hg38
```

Format

An object of class `data.table` (inherits from `data.frame`) with 26807 rows and 5 columns.

gtf_mm10 *gene model (mm10)*

Description

gene model (mm10)

Usage

```
gtf_mm10
```

Format

An object of class `data.table` (inherits from `data.frame`) with 30336 rows and 5 columns.

hc_example *example hclust tree*

Description

example hclust tree

Usage

```
hc_example
```

Format

An object of class `hclust` of length 7.

joint_post_example *example joint single-cell cnv posterior dataframe*

Description

example joint single-cell cnv posterior dataframe

Usage

```
joint_post_example
```

Format

An object of class `data.table` (inherits from `data.frame`) with 3806 rows and 71 columns.

mut_graph_example *example mutation graph*

Description

example mutation graph

Usage

```
mut_graph_example
```

Format

An object of class igraph of length 5.

Numbat *Numbat R6 class*

Description

Used to allow users to plot results

Value

a new 'Numbat' object

Public fields

`label` character Sample name

`gtf` dataframe Transcript annotation

`joint_post` dataframe Joint posterior

`exp_post` dataframe Expression posterior

`allele_post` dataframe Allele posetrior

`bulk_subtrees` dataframe Bulk profiles of lineage subtrees

`bulk_clones` dataframe Bulk profiles of clones

`segs_consensus` dataframe Consensus segments

`tree_post` list Tree posterior

`mut_graph` igraph Mutation history graph

`gtree` tbl_graph Single-cell phylogeny

`clone_post` dataframe Clone posteriors

`gexp_roll_wide` matrix Smoothed expression of single cells

`hc` hclust Initial hierarchical clustering

Methods

Public methods:

- Numbat$new()
- Numbat$plot_phylo_heatmap()
- Numbat$plot_exp_roll()
- Numbat$plot_mut_history()
- Numbat$plot_sc_tree()
- Numbat$plot_consensus()
- Numbat$plot_clone_profile()
- Numbat$clone()

Method `new()`**:** initialize Numbat class

Usage:

```
Numbat$new(out_dir, i = 2, gtf = gtf_hg38, verbose = TRUE)
```

Arguments:

`out_dir` character string Output directory

`i` integer Get results from which iteration (either 1 or 2) (default=2)

`gtf` dataframe Transcript gtf (default=gtf_hg38)

`verbose` logical Whether to output verbose results (default=TRUE)

Returns: a new 'Numbat' object

Method `plot_phylo_heatmap()`**:** Plot the single-cell CNV calls in a heatmap and the corresponding phylogeny

Usage:

```
Numbat$plot_phylo_heatmap(...)
```

Arguments:

`...` additional parameters passed to plot_phylo_heatmap()

Method `plot_exp_roll()`**:** Plot window-smoothed expression profiles

Usage:

```
Numbat$plot_exp_roll(k = 3, n_sample = 300, ...)
```

Arguments:

`k` integer Number of clusters

`n_sample` integer Number of cells to subsample

`...` additional parameters passed to plot_exp_roll()

Method `plot_mut_history()`**:** Plot the mutation history of the tumor

Usage:

```
Numbat$plot_mut_history(...)
```

Arguments:

`...` additional parameters passed to plot_mut_history()

Method `plot_sc_tree()`**:** Plot the single cell phylogeny

Usage:

```
Numbat$plot_sc_tree(...)
```

Arguments:

`...` additional parameters passed to plot_sc_tree()

Method `plot_consensus()`**:** Plot consensus segments

Usage:

```
Numbat$plot_consensus(...)
```

Arguments:

`...` additional parameters passed to plot_sc_tree()

Method `plot_clone_profile()`**:** Plot clone cnv profiles

Usage:

```
Numbat$plot_clone_profile(...)
```

Arguments:

`...` additional parameters passed to plot_clone_profile()

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
Numbat$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

`phylogeny_example` *example single-cell phylogeny*

Description

example single-cell phylogeny

Usage

```
phylogeny_example
```

Format

An object of class `tbl_graph` (inherits from `igraph`) of length 345.

plot_bulks *Plot a group of pseudobulk HMM profiles*

Description

Plot a group of pseudobulk HMM profiles

Usage

```
plot_bulks(bulks, ..., ncol = 1, title = TRUE, title_size = 8)
```

Arguments

bulks	dataframe Pseudobulk profiles annotated with "sample" column
...	additional parameters passed to plot_psbulk()
ncol	integer Number of columns
title	logical Whether to add titles to individual plots
title_size	numeric Size of titles

Value

a ggplot object

Examples

```
p = plot_bulks(bulk_example)
```

plot_consensus *Plot consensus CNVs*

Description

Plot consensus CNVs

Usage

```
plot_consensus(segs)
```

Arguments

segs	dataframe Consensus segments

Value

ggplot object

Examples

```
p = plot_consensus(segs_example)
```

`plot_exp_roll` *Plot single-cell smoothed expression magnitude heatmap*

Description

Plot single-cell smoothed expression magnitude heatmap

Usage

```
plot_exp_roll(
  gexp_roll_wide,
  hc,
  k,
  gtf,
  lim = 0.8,
  n_sample = 300,
  reverse = TRUE,
  plot_tree = TRUE
)
```

Arguments

`gexp_roll_wide`	matrix Cell x gene smoothed expression magnitudes
`hc`	hclust Hierarchical clustring result
`k`	integer Number of clusters
`gtf`	dataframe Transcript GTF
`lim`	numeric Limit for expression magnitudes
`n_sample`	integer Number of cells to subsample
`reverse`	logical Whether to reverse the cell order
`plot_tree`	logical Whether to plot the dendrogram

Value

ggplot A single-cell heatmap of window-smoothed expression CNV signals

Examples

```
p = plot_exp_roll(gexp_roll_example, gtf = gtf_hg38, hc = hc_example, k = 3)
```

`plot_mut_history` *Plot mutational history*

Description

Plot mutational history

Usage

```
plot_mut_history(
  G,
  clone_post = NULL,
  edge_label_size = 4,
  node_label_size = 6,
  node_size = 10,
  arrow_size = 2,
  show_clone_size = TRUE,
  show_distance = TRUE,
  legend = TRUE,
  edge_label = TRUE,
  node_label = TRUE,
  horizontal = TRUE,
  pal = NULL
)
```

Arguments

`G`	igraph Mutation history graph
`clone_post`	dataframe Clone assignment posteriors
`edge_label_size`	numeric Size of edge label
`node_label_size`	numeric Size of node label
`node_size`	numeric Size of nodes
`arrow_size`	numeric Size of arrows
`show_clone_size`	logical Whether to show clone size
`show_distance`	logical Whether to show evolutionary distance between clones
`legend`	logical Whether to show legend
`edge_label`	logical Whether to label edges
`node_label`	logical Whether to label nodes
`horizontal`	logical Whether to use horizontal layout
`pal`	named vector Node colors

Value

ggplot object

Examples

```
p = plot_mut_history(mut_graph_example)
```

`plot_phylo_heatmap` *Plot single-cell CNV calls along with the clonal phylogeny*

Description

Plot single-cell CNV calls along with the clonal phylogeny

Usage

```
plot_phylo_heatmap(
  gtree,
  joint_post,
  segs_consensus,
  clone_post = NULL,
  p_min = 0.9,
  annot = NULL,
  pal_annot = NULL,
  annot_title = "Annotation",
  annot_scale = NULL,
  clone_dict = NULL,
  clone_bar = TRUE,
  clone_stack = TRUE,
  pal_clone = NULL,
  clone_title = "Genotype",
  clone_legend = TRUE,
  line_width = 0.1,
  tree_height = 1,
  branch_width = 0.2,
  tip_length = 0.2,
  annot_bar_width = 0.25,
  clone_bar_width = 0.25,
  bar_label_size = 7,
  tvn_line = TRUE,
  clone_line = FALSE,
  exclude_gap = FALSE,
  root_edge = TRUE,
  raster = FALSE,
  show_phylo = TRUE
)
```

Arguments

gtree	tbl_graph The single-cell phylogeny
joint_post	dataframe Joint single cell CNV posteriors
segs_consensus	datatframe Consensus segment dataframe
clone_post	dataframe Clone assignment posteriors
p_min	numeric Probability threshold to display CNV calls
annot	dataframe Cell annotations, dataframe with 'cell' and additional annotation columns
pal_annot	named vector Colors for cell annotations
annot_title	character Legend title for the annotation bar
annot_scale	ggplot scale Color scale for the annotation bar
clone_dict	named vector Clone annotations, mapping from cell name to clones
clone_bar	logical Whether to display clone bar plot
clone_stack	character Whether to plot clone assignment probabilities as stacked bar
pal_clone	named vector Clone colors
clone_title	character Legend title for the clone bar
clone_legend	logical Whether to display the clone legend
line_width	numeric Line width for CNV heatmap
tree_height	numeric Relative height of the phylogeny plot
branch_width	numeric Line width in the phylogeny
tip_length	numeric Length of tips in the phylogeny
annot_bar_width	numeric Width of annotation bar
clone_bar_width	numeric Width of clone genotype bar
bar_label_size	numeric Size of sidebar text labels
tvn_line	logical Whether to draw line separating tumor and normal cells
clone_line	logical Whether to display borders for clones in the heatmap
exclude_gap	logical Whether to mark gap regions
root_edge	logical Whether to plot root edge
raster	logical Whether to raster images
show_phylo	logical Whether to display phylogeny on y axis

Value

ggplot panel

Examples

```
p = plot_phylo_heatmap(
    gtree = phylogeny_example,
    joint_post = joint_post_example,
    segs_consensus = segs_example)
```

plot_psbulk *Plot a pseudobulk HMM profile*

Description

Plot a pseudobulk HMM profile

Usage

```
plot_psbulk(
  bulk,
  use_pos = TRUE,
  allele_only = FALSE,
  min_LLR = 5,
  min_depth = 8,
  exp_limit = 2,
  phi_mle = TRUE,
  theta_roll = FALSE,
  dot_size = 0.8,
  dot_alpha = 0.5,
  legend = TRUE,
  exclude_gap = TRUE,
  genome = "hg38",
  text_size = 10,
  raster = FALSE
)
```

Arguments

bulk	dataframe Pseudobulk profile
use_pos	logical Use marker position instead of index as x coordinate
allele_only	logical Only plot alleles
min_LLR	numeric LLR threshold for event filtering
min_depth	numeric Minimum coverage depth for a SNP to be plotted
exp_limit	numeric Expression logFC axis limit
phi_mle	logical Whether to plot estimates of segmental expression fold change
theta_roll	logical Whether to plot rolling estimates of allele imbalance
dot_size	numeric Size of marker dots
dot_alpha	numeric Transparency of the marker dots
legend	logical Whether to show legend
exclude_gap	logical Whether to mark gap regions and centromeres
genome	character Genome build, either 'hg38' or 'hg19'
text_size	numeric Size of text in the plot
raster	logical Whether to raster images

Value

ggplot Plot of pseudobulk HMM profile

Examples

```
p = plot_psbulk(bulk_example)
```

`plot_sc_tree` *Plot single-cell smoothed expression magnitude heatmap*

Description

Plot single-cell smoothed expression magnitude heatmap

Usage

```
plot_sc_tree(
  gtree,
  label_mut = TRUE,
  label_size = 3,
  dot_size = 2,
  branch_width = 0.5,
  tip = TRUE,
  tip_length = 0.5,
  pal_clone = NULL
)
```

Arguments

`gtree`	tbl_graph The single-cell phylogeny
`label_mut`	logical Whether to label mutations
`label_size`	numeric Size of mutation labels
`dot_size`	numeric Size of mutation nodes
`branch_width`	numeric Width of branches in tree
`tip`	logical Whether to plot tip point
`tip_length`	numeric Length of the tips
`pal_clone`	named vector Clone colors

Value

ggplot A single-cell phylogeny with mutation history labeled

Examples

```
p = plot_sc_tree(phylogeny_example)
```

pre_likelihood_hmm *HMM object for unit tests*

Description

HMM object for unit tests

Usage

```
pre_likelihood_hmm
```

Format

An object of class `list` of length 10.

ref_hca *reference expression magnitudes from HCA*

Description

reference expression magnitudes from HCA

Usage

```
ref_hca
```

Format

An object of class `matrix` (inherits from `array`) with 24756 rows and 12 columns.

ref_hca_counts *reference expression counts from HCA*

Description

reference expression counts from HCA

Usage

```
ref_hca_counts
```

Format

An object of class `matrix` (inherits from `array`) with 24857 rows and 12 columns.

run_numbat *Run workflow to decompose tumor subclones*

Description

Run workflow to decompose tumor subclones

Usage

```
run_numbat(
  count_mat,
  lambdas_ref,
  df_allele,
  genome = "hg38",
  out_dir = tempdir(),
  max_iter = 2,
  max_nni = 100,
  t = 1e-05,
  gamma = 20,
  min_LLR = 5,
  alpha = 1e-04,
  eps = 1e-05,
  max_entropy = 0.5,
  init_k = 3,
  min_cells = 50,
  tau = 0.3,
  nu = 1,
  max_cost = ncol(count_mat) * tau,
  min_depth = 0,
  common_diploid = TRUE,
  min_overlap = 0.45,
  ncores = 1,
  ncores_nni = ncores,
  random_init = FALSE,
  segs_loh = NULL,
  verbose = TRUE,
  diploid_chroms = NULL,
  use_loh = NULL,
  min_genes = 10,
  skip_nj = FALSE,
  multi_allelic = TRUE,
  p_multi = 1 - alpha,
  plot = TRUE,
  check_convergence = FALSE,
  exclude_neu = TRUE
)
```

Arguments

count_mat	dgCMatrix Raw count matrices where rownames are genes and column names are cells
lambdas_ref	matrix Either a named vector with gene names as names and normalized expression as values, or a matrix where rownames are genes and columns are pseudobulk names
df_allele	dataframe Allele counts per cell, produced by preprocess_allele
genome	character Genome version (hg38, hg19, or mm10)
out_dir	string Output directory
max_iter	integer Maximum number of iterations to run the phyologeny optimization
max_nni	integer Maximum number of iterations to run NNI in the ML phylogeny inference
t	numeric Transition probability
gamma	numeric Dispersion parameter for the Beta-Binomial allele model
min_LLR	numeric Minimum LLR to filter CNVs
alpha	numeric P value cutoff for diploid finding
eps	numeric Convergence threshold for ML tree search
max_entropy	numeric Entropy threshold to filter CNVs
init_k	integer Number of clusters in the initial clustering
min_cells	integer Minimum number of cells to run HMM on
tau	numeric Factor to determine max_cost as a function of the number of cells (0-1)
nu	numeric Phase switch rate
max_cost	numeric Likelihood threshold to collapse internal branches
min_depth	integer Minimum allele depth
common_diploid	logical Whether to find common diploid regions in a group of peusdobulks
min_overlap	numeric Minimum CNV overlap threshold
ncores	integer Number of threads to use
ncores_nni	integer Number of threads to use for NNI
random_init	logical Whether to initiate phylogney using a random tree (internal use only)
segs_loh	dataframe Segments of clonal LOH to be excluded
verbose	logical Verbosity
diploid_chroms	vector Known diploid chromosomes
use_loh	logical Whether to include LOH regions in the expression baseline
min_genes	integer Minimum number of genes to call a segment
skip_nj	logical Whether to skip NJ tree construction and only use UPGMA
multi_allelic	logical Whether to call multi-allelic CNVs
p_multi	numeric P value cutoff for calling multi-allelic CNVs
plot	logical Whether to plot results
check_convergence	logical Whether to terminate iterations based on consensus CNV convergence
exclude_neu	logical Whether to exclude neutral segments from CNV retesting (internal use only)

Value

a status code

`segs_example` *example CNV segments dataframe*

Description

example CNV segments dataframe

Usage

```
segs_example
```

Format

An object of class `data.table` (inherits from `data.frame`) with 27 rows and 30 columns.

`upgma` *UPGMA and WPGMA clustering*

Description

UPGMA and WPGMA clustering

Usage

```
upgma(D, method = "average", ...)
```

Arguments

`D`	A distance matrix.
`method`	The agglomeration method to be used. This should be (an unambiguous abbreviation of) one of "ward", "single", "complete", "average", "mcquitty", "median" or "centroid". The default is "average".
`...`	Further arguments passed to or from other methods.

vcf_meta *example VCF header*

Description

example VCF header

Usage

```
vcf_meta
```

Format

An object of class `character` of length 65.

Index