

Package: clusnalyst (via r-universe)

October 14, 2026

Title Point-and-Click Cluster Analysis of Molecular and Trait Data

Version 0.1.0

Description A 'shiny' application and a set of plain R functions for clustering samples from dominant markers, SNP genotypes (CSV, VCF or HapMap), quantitative or mixed traits, and DNA or protein sequences. It computes distance matrices, builds trees by UPGMA, WPGMA, single, complete or Ward linkage, neighbour joining or BIONJ, and checks them with marker bootstrap support (Felsenstein 1985 <[doi:10.1111/j.1558-5646.1985.tb00420.x](https://doi.org/10.1111/j.1558-5646.1985.tb00420.x)>), cophenetic correlation, average silhouette width, the gap statistic and Mantel tests. It also offers principal coordinates and principal components, k-means and partitioning around medoids, and exports a self-contained HTML report and an R script that reproduces the analysis.

License MIT + file LICENSE

URL <https://github.com/ebenogoe/clusnalyst>,
<https://ebenogoe.github.io/clusnalyst/>

BugReports <https://github.com/ebenogoe/clusnalyst/issues>

Depends R (>= 4.1.0)

Imports ade4, ape, Biostrings, bsicons, bslib (>= 0.8.0), cicerone, cluster, config, ggiraph, ggplot2 (>= 3.5.0), ggtree, golem, grDevices, htmltools, jsonlite, methods, msa, reactable, readxl, seqinr, shiny (>= 1.8.1), stats, svglite, tools, utils, vcfR, writexl

Suggests chromote, pkgload, shinytest2, testthat (>= 3.0.0)

Config/testthat/edition 3

Encoding UTF-8

Language en-GB

Roxygen list(markdown = TRUE)

RoxygenNote 7.3.3

Config/pak/sysreqs libcairo2-dev cmake libfontconfig1-dev libfreetype6-dev libfribidi-dev make libharfbuzz-dev libicu-dev libpng-dev libuv1-dev zlib1g-dev

Repository <https://ebenogoe.r-universe.dev>

Date/Publication 2026-10-14 19:33:12 UTC

RemoteUrl <https://github.com/ebenogoe/clusnalyst>

RemoteRef main

RemoteSha 026a65b73d10f5831353ea0330258c363a2299d5

Contents

align_sequences	3
attach_metadata	3
bootstrap_tree	4
build_report	5
build_script	5
build_tree	6
cl_palette	7
compute_distance	7
consensus_tree	8
cophenetic_cor	8
cophenetic_table	9
data_types	9
default_distance	10
default_settings	10
demo_info	11
detect_format	11
distance_choices	12
filter_data	12
genotypes_to_dosage	13
guess_type	14
has_numeric	14
mantel_test	15
metadata_distance	15
optimal_k	16
ordinate	16
partition	17
plot_heatmap	18
plot_k	18
plot_mantel	19
plot_ordination	19
plot_qc	20
plot_scree	21
plot_support	21
plot_tree	22
qc_summary	23
read_fasta	23

read_hapmap	24
read_markers	24
read_metadata	25
read_second_matrix	25
read_vcf	26
run_analysis	26
run_app	27
sample_names	27
save_session	28
silhouette_meaning	28
theme_clusnalyt	29
tree_choices	29
Index	30

align_sequences	<i>Align sequences with the msa package</i>
-----------------	---

Description

Align sequences with the msa package

Usage

```
align_sequences(d, method = c("ClustalOmega", "ClustalW", "Muscle"))
```

Arguments

d	A cl_data object of type "dna" or "protein".
method	"ClustalOmega", "ClustalW" or "Muscle".

Value

The same object with an aligned matrix in x.

attach_metadata	<i>Attach metadata to a dataset</i>
-----------------	-------------------------------------

Description

Attach metadata to a dataset

Usage

```
attach_metadata(d, meta)
```

Arguments

d A cl_data object.
 meta A data frame from `read_metadata()`.

Value

d with meta holding one row per sample (in sample order).

bootstrap_tree	<i>Bootstrap support for a tree</i>
----------------	-------------------------------------

Description

Resamples the markers, traits or alignment sites with replacement, recomputes the distance matrix and the tree for each replicate, and reports how often each group in the original tree is found again (Felsenstein 1985). For rooted trees the groups are clades; for unrooted NJ and BIONJ trees they are bipartitions.

Usage

```
bootstrap_tree(
  d,
  tree,
  distance = NULL,
  B = 100,
  seed = 1,
  sqrt_binary = FALSE,
  scale = TRUE,
  progress = NULL
)
```

Arguments

d The cl_data the tree was built from.
 tree A cl_tree from `build_tree()`.
 distance Distance method used for the tree.
 B Number of bootstrap replicates.
 seed Random seed, so the run can be repeated exactly.
 sqrt_binary, scale Passed to `compute_distance()`.
 progress Optional function called with the share done (0 to 1).

Value

A list with support (percent, one value per internal node of `tree$phylo`), trees (the replicate trees), B, failed and seed.

build_report	<i>Write a self-contained HTML report</i>
--------------	---

Description

Includes whichever results are present: data summary, quality check, distances, tree, bootstrap, cluster number, groups and ordination, plus the settings and software versions. Figures are embedded as SVG, so the file can be opened anywhere and needs no other files.

Usage

```
build_report(  
  res,  
  settings = list(),  
  path,  
  title = "Clusnalyst report",  
  tree_layout = "rectangular"  
)
```

Arguments

res	A list with any of raw, data, dist, tree, boot, kres, groups, ord, mantel, as produced by the app or by run_analysis() (add raw for the unfiltered data).
settings	The settings list.
path	Output file.
title	Report title.
tree_layout	Tree layout for the figure.

Value

path, invisibly.

build_script	<i>Write an R script that repeats an analysis</i>
--------------	---

Description

Write an R script that repeats an analysis

Usage

```
build_script(settings, source, meta_file = NULL)
```

Arguments

settings	A settings list (see default_settings()).
source	List describing the data: demo (a demo name) or file, plus type and transposed for tables.
meta_file	Optional metadata file name.

Value

A character string of R code.

build_tree	<i>Build a tree from a distance matrix</i>
------------	--

Description

Hierarchical methods give a rooted, ultrametric tree. NJ and BIONJ give an unrooted tree; give outgroup to root it. Negative branch lengths from NJ are set to zero.

Usage

```
build_tree(dist, method = "upgma", outgroup = NULL)
```

Arguments

dist	A dist object.
method	A method from tree_choices() .
outgroup	Optional sample name(s) to root an NJ or BIONJ tree on.

Value

A cl_tree list with phylo (an ape tree), hclust (for hierarchical methods), method and rooted.

Examples

```
d <- clusnalyt_demo("binary")
tr <- build_tree(compute_distance(d), "upgma")
```

cl_palette	<i>Colours for groups</i>
------------	---------------------------

Description

The Okabe and Ito palette (safe for colour-blind readers), without its yellow and black, and HCL colours beyond eight groups.

Usage

```
cl_palette(n)
```

Arguments

n	Number of colours.
---	--------------------

Value

Character vector of hex colours.

compute_distance	<i>Compute a distance matrix</i>
------------------	----------------------------------

Description

Missing values are handled pairwise: each pair of samples is compared on the markers (or sites) both of them have.

Usage

```
compute_distance(d, method = NULL, sqrt_binary = FALSE, scale = TRUE)
```

Arguments

d	A cl_data object.
method	A method from distance_choices() ; the default for the data type if NULL.
sqrt_binary	Use $\sqrt{1 - S}$ for binary measures.
scale	Scale numeric traits to unit variance before Euclidean or Manhattan distances.

Details

Binary measures are returned as 1 minus the similarity. Set `sqrt_binary = TRUE` for the square root form that `ade4::dist.binary()` returns, which is Euclidean. For biallelic SNPs, IBS distance equals Prevosti's distance.

Value

A dist object with a method attribute.

Examples

```
d <- clusnalyt_demo("binary")
dm <- compute_distance(d, "jaccard")
```

consensus_tree	<i>Majority-rule consensus of bootstrap trees</i>
----------------	---

Description

Majority-rule consensus of bootstrap trees

Usage

```
consensus_tree(boot, p = 0.5)
```

Arguments

- boot Result of `bootstrap_tree()`.
- p Share of trees a group must appear in (0.5 = majority rule).

Value

An ape tree.

cophenetic_cor	<i>Cophenetic correlation of a tree</i>
----------------	---

Description

The correlation between the distances you started with and the distances implied by the tree. Values near 1 mean the tree keeps the original distances well.

Usage

```
cophenetic_cor(dist, tree)
```

Arguments

- dist A dist object.
- tree A `cl_tree`.

Value

A number.

cophenetic_table	<i>Compare tree methods by cophenetic correlation</i>
------------------	---

Description

Compare tree methods by cophenetic correlation

Usage

```
cophenetic_table(dist, methods = tree_choices())
```

Arguments

dist	A dist object.
methods	Methods to compare.

Value

A data frame sorted from the best-fitting method down.

data_types	<i>Data types Clusnalyst works with</i>
------------	---

Description

Data types Clusnalyst works with

Usage

```
data_types()
```

Value

A named character vector, labels as names.

default_distance	<i>Default distance for a dataset</i>
------------------	---------------------------------------

Description

Default distance for a dataset

Usage

```
default_distance(d)
```

Arguments

d A cl_data object.

Value

A method name.

default_settings	<i>Default analysis settings</i>
------------------	----------------------------------

Description

Every step of the app is described by one entry in this list. The app fills it in as you go, [run_analysis\(\)](#) replays it, and [build_script\(\)](#) turns it into R code.

Usage

```
default_settings(d = NULL)
```

Arguments

d Optional cl_data; picks defaults that suit its type.

Value

A named list.

demo_info

*Demo datasets***Description**

Five small datasets for trying the app. binary, snp and traits are simulated (see data-raw/make_demos.R in the source). dna is the woodmouse cytochrome b alignment from the ape package, and protein is the hemoglobin alpha example from the msa package.

Usage

```
demo_info()
```

```
clusnalyst_demo(name = c("snp", "binary", "traits", "dna", "protein"))
```

Arguments

name One of demo_info()\$name.

Value

demo_info() returns a data frame describing each dataset.

clusnalyst_demo() returns a cl_data object, with metadata where the demo has it.

Examples

```
clusnalyst_demo("snp")
```

detect_format

*Guess the format of a data file***Description**

Looks at the file extension first and then at the first lines of the file.

Usage

```
detect_format(path, name = basename(path))
```

Arguments

path Path to the file.

name Original file name, if path is a temporary upload.

Value

One of "csv", "tsv", "excel", "vcf", "hapmap" or "fasta".

distance_choices	<i>Distance measures available for a data type</i>
------------------	--

Description

Distance measures available for a data type

Usage

```
distance_choices(type)
```

Arguments

type A data type from `data_types()`.

Value

A named character vector (labels as names). The first entry is the default.

filter_data	<i>Filter markers, traits or sequences</i>
-------------	--

Description

Markers (or traits) are filtered first, then samples are checked against the markers that are left.

Usage

```
filter_data(
  d,
  max_marker_missing = 0.2,
  min_maf = 0,
  drop_monomorphic = TRUE,
  max_sample_missing = 0.5,
  drop_empty_sites = TRUE
)
```

Arguments

d A `cl_data` object.

max_marker_missing Largest share of missing values allowed per marker or trait (0 to 1).

min_maf Smallest minor allele (or band) frequency kept (marker data only).

drop_monomorphic Drop markers or traits that do not vary.

max_sample_missing

Largest share of missing values allowed per sample.

drop_empty_sites

For aligned sequences, drop sites that are gaps in every sequence.

Value

The filtered `cl_data`, with a `filter` element recording what was removed.

genotypes_to_dosage	<i>Convert letter genotypes to allele dosages</i>
---------------------	---

Description

Accepts two-letter calls (AA, AG), separated calls (A/G, A|G) and single IUPAC codes (A, R, Y, ...). For each marker the most common allele is the reference, so a dosage counts copies of the other allele: 0 = homozygous for the common allele, 1 = heterozygous, 2 = homozygous for the other allele. Markers with more than two alleles count every allele that is not the common one.

Usage

```
genotypes_to_dosage(m)
```

Arguments

`m` Character matrix with samples in rows and markers in columns.

Value

Numeric matrix of 0, 1, 2 and NA, same dimensions as `m`.

Examples

```
genotypes_to_dosage(matrix(c("AA", "AG", "GG", "R", "NN", "A/G"), 3))
```

guess_type	<i>Guess the data type of a table</i>
------------	---------------------------------------

Description

Guess the data type of a table

Usage

```
guess_type(tab)
```

Arguments

tab	A data frame whose first column holds IDs and whose other columns are character.
-----	--

Value

A list with type and encoding.

has_numeric	<i>Can PCA or k-means be used on this dataset?</i>
-------------	--

Description

Can PCA or k-means be used on this dataset?

Usage

```
has_numeric(d)
```

Arguments

d	A cl_data object.
---	-------------------

Value

TRUE for marker data and for data with numeric traits.

mantel_test	<i>Mantel test between two distance matrices</i>
-------------	--

Description

Uses the samples the two matrices share.

Usage

```
mantel_test(d1, d2, permutations = 999, seed = 1)
```

Arguments

d1, d2	dist objects (or square matrices) with sample names.
permutations	Number of permutations.
seed	Random seed.

Value

A list with r, p, n and permutations.

metadata_distance	<i>Distances from numeric metadata columns</i>
-------------------	--

Description

Useful for a Mantel test of genetic distance against geography (for example latitude and longitude) or any measured covariates.

Usage

```
metadata_distance(meta, columns, scale = FALSE)
```

Arguments

meta	A data frame with sample names as row names.
columns	Numeric columns to use.
scale	Scale the columns first.

Value

A Euclidean dist object.

optimal_k	<i>Choose the number of clusters</i>
-----------	--------------------------------------

Description

For each k from 2 to $k_{\max}$, splits the samples into k groups and records the average silhouette width (Rousseeuw 1987), the within-group sum of squares (for an elbow plot) and, optionally, the gap statistic (Tibshirani et al. 2001). Hierarchical trees are cut with `stats::cutree()`; NJ and BIONJ trees cannot be cut that way, so partitioning around medoids (PAM) is used instead.

Usage

```
optimal_k(dist, tree = NULL, k_max = 10, gap = FALSE, gap_B = 50, seed = 1)
```

Arguments

<code>dist</code>	A dist object.
<code>tree</code>	Optional <code>cl_tree</code> . If it is hierarchical, its cuts are used.
<code>k_max</code>	Largest k to try.
<code>gap</code>	Also compute the gap statistic (slower).
<code>gap_B</code>	Reference datasets for the gap statistic.
<code>seed</code>	Random seed for the gap statistic.

Value

A list with a table (one row per k), `best_silhouette`, `best_gap` and `clustering` (how the groups were formed).

ordinate	<i>Principal coordinates or principal components</i>
----------	--

Description

PCoA (classical multidimensional scaling) works from any distance matrix. PCA needs numeric data; missing values are replaced by the column mean, and traits (not markers) are scaled to unit variance.

Usage

```
ordinate(dist = NULL, d = NULL, method = c("pcoa", "pca"), n_axes = 5)
```

Arguments

dist	A dist object (for PCoA).
d	A cl_data object (for PCA).
method	"pcoa" or "pca".
n_axes	Number of axes to keep.

Value

A cl_ord list with scores, variance (percent for every axis) and method.

partition	<i>Split samples into groups</i>
-----------	----------------------------------

Description

Split samples into groups

Usage

```
partition(
  dist,
  k,
  method = c("tree", "pam", "kmeans"),
  tree = NULL,
  d = NULL,
  seed = 1
)
```

Arguments

dist	A dist object.
k	Number of groups.
method	"tree" (cut a hierarchical tree), "pam" (partitioning around medoids, any distance) or "kmeans" (numeric data only).
tree	A hierarchical cl_tree, for method = "tree".
d	A cl_data object, for method = "kmeans".
seed	Random seed for k-means.

Value

A named integer vector of group numbers. Groups are numbered in the order they first appear down the tree when a tree is given.

plot_heatmap	<i>Heatmap of a distance matrix</i>
--------------	-------------------------------------

Description

Heatmap of a distance matrix

Usage

```
plot_heatmap(dist, order = NULL, dark = FALSE, interactive = TRUE)
```

Arguments

dist	A dist object.
order	Optional sample order (for example the tree's tip order).
dark	Colours for a dark background.
interactive	Add hover text to each cell.

Value

A ggplot object.

plot_k	<i>Plot the cluster-number diagnostics</i>
--------	--

Description

Plot the cluster-number diagnostics

Usage

```
plot_k(kres, dark = FALSE)
```

Arguments

kres	Result of optimal_k() .
dark	Colours for a dark background.

Value

A ggplot object.

plot_mantel	<i>Plot two distance matrices against each other</i>
-------------	--

Description

Plot two distance matrices against each other

Usage

```
plot_mantel(mantel, labels = c("Distance 1", "Distance 2"), dark = FALSE)
```

Arguments

mantel	Result of <code>mantel_test()</code> .
labels	Axis labels.
dark	Colours for a dark background.

Value

A ggplot object.

plot_ordination	<i>Plot an ordination</i>
-----------------	---------------------------

Description

Plot an ordination

Usage

```
plot_ordination(  
  ord,  
  groups = NULL,  
  group_title = "Group",  
  axes = c(1, 2),  
  labels = FALSE,  
  ellipses = TRUE,  
  tooltip = NULL,  
  dark = FALSE  
)
```

Arguments

ord	A cl_ord from <code>ordinate()</code> .
groups	Optional named vector of groups for colour.
group_title	Legend title.
axes	Which two axes to plot.
labels	Show sample names.
ellipses	Draw 95% ellipses for groups with at least four samples.
tooltip	Optional named character vector of hover text.
dark	Colours for a dark background.

Value

A ggplot object.

plot_qc	<i>Plot quality-check summaries</i>
---------	-------------------------------------

Description

Plot quality-check summaries

Usage

```
plot_qc(qc, dark = FALSE, thresholds = NULL)
```

Arguments

qc	Result of <code>qc_summary()</code> .
dark	Colours for a dark background.
thresholds	Optional named list with marker_missing and maf cut-offs to draw.

Value

A ggplot object.

plot_scree	<i>Scree plot of an ordination</i>
------------	------------------------------------

Description

Scree plot of an ordination

Usage

```
plot_scree(ord, n = 10, dark = FALSE)
```

Arguments

ord	A cl_ord from <code>ordinate()</code> .
n	Number of axes to show.
dark	Colours for a dark background.

Value

A ggplot object.

plot_support	<i>Plot bootstrap support values</i>
--------------	--------------------------------------

Description

Plot bootstrap support values

Usage

```
plot_support(boot, dark = FALSE)
```

Arguments

boot	Result of <code>bootstrap_tree()</code> .
dark	Colours for a dark background.

Value

A ggplot object.

plot_tree*Plot a tree*

Description

Plot a tree

Usage

```
plot_tree(  
  tree,  
  layout = "rectangular",  
  groups = NULL,  
  group_title = "Group",  
  support = NULL,  
  support_min = 0,  
  tooltip = NULL,  
  label_size = 3.5,  
  title = NULL,  
  dark = FALSE,  
  show_scale = TRUE  
)
```

Arguments

tree	A cl_tree or an ape phylo.
layout	One of "rectangular", "circular", "fan", "unrooted" or "slanted".
groups	Optional named vector (names = sample names) used to colour the tips.
group_title	Legend title for groups.
support	Optional bootstrap support, one value per internal node.
support_min	Only show support at or above this value.
tooltip	Optional named character vector of hover text per tip.
label_size	Size of the tip labels.
title	Plot title.
dark	Colours for a dark background.
show_scale	Draw a branch-length scale bar.

Value

A ggplot object (interactive with [ggiraph::girafe\(\)](#)).

`qc_summary`*Quality summary of a dataset*

Description

For marker data: missing data per marker and per sample, minor allele (or band) frequency, heterozygosity and monomorphic markers. For traits: missing data and spread per trait. For sequences: lengths, gaps and variable sites.

Usage

```
qc_summary(d)
```

Arguments

`d` A `cl_data` object.

Value

A list with markers and samples data frames and totals.

`read_fasta`*Read a FASTA file*

Description

Read a FASTA file

Usage

```
read_fasta(path, type = NULL)
```

Arguments

`path` Path to the file.
`type` "dna" or "protein"; guessed from the letters if NULL.

Value

A `cl_data` object of type "dna" or "protein".

read_hapmap	<i>Read a HapMap genotype file</i>
-------------	------------------------------------

Description

Read a HapMap genotype file

Usage

```
read_hapmap(path)
```

Arguments

path	Path to a HapMap (.hmp.txt) file.
------	-----------------------------------

Value

A cl_data object of type "snp".

read_markers	<i>Read a data file into a Clusnalyt dataset</i>
--------------	--

Description

Reads CSV, TSV or Excel tables (first column = sample IDs), VCF, HapMap and FASTA files. The data type and the table orientation are guessed when not given.

Usage

```
read_markers(
  path,
  name = basename(path),
  format = NULL,
  type = NULL,
  transpose = NULL
)
```

Arguments

path	Path to the file.
name	Original file name (used for format detection).
format	File format; guessed with detect_format() if NULL.
type	Data type, one of data_types() ; guessed if NULL.
transpose	TRUE if samples are in columns and markers in rows. Guessed if NULL.

Value

A cl_data object.

Examples

```
f <- system.file("extdata", "demo_binary.csv", package = "clusnalyt")
read_markers(f)
```

read_metadata	<i>Read a sample metadata table</i>
---------------	-------------------------------------

Description

Read a sample metadata table

Usage

```
read_metadata(path, name = basename(path))
```

Arguments

path	Path to a CSV, TSV or Excel file whose first column holds sample IDs.
name	Original file name.

Value

A data frame with sample IDs as row names.

read_second_matrix	<i>Read a second matrix for a Mantel test</i>
--------------------	---

Description

A square table whose row and column names are the same samples is read as a distance matrix. Anything else is read as data with [read_markers\(\)](#) and turned into distances with the default measure for its type.

Usage

```
read_second_matrix(path, name = basename(path))
```

Arguments

path	Path to the file.
name	Original file name.

Value

A dist object.

read_vcf	<i>Read a VCF file as SNP dosages</i>
----------	---------------------------------------

Description

Read a VCF file as SNP dosages

Usage

```
read_vcf(path)
```

Arguments

path Path to a .vcf or .vcf.gz file.

Value

A cl_data object of type "snp".

run_analysis	<i>Run a whole analysis from settings</i>
--------------	---

Description

Run a whole analysis from settings

Usage

```
run_analysis(d, settings = default_settings(d), progress = NULL)
```

Arguments

d A cl_data object as read (before filtering).
 settings A list like `default_settings()`. Steps whose entry is NULL are skipped.
 progress Optional function called with a message at each step.

Value

A list with data, dist, tree, boot, kres, groups and ord.

Examples

```
res <- run_analysis(clusnalyst_demo("binary"))
res$tree
```

run_app	<i>Run the Clusnalyst app</i>
---------	-------------------------------

Description

Run the Clusnalyst app

Usage

```
run_app(  
  onStart = NULL,  
  options = list(),  
  enableBookmarking = NULL,  
  uiPattern = "/",  
  ...  
)
```

Arguments

onStart, options, enableBookmarking, uiPattern
Passed to `shiny::shinyApp()`.
... Options passed to the app as golem options.

Value

A Shiny app object.

Examples

```
if (interactive()) run_app()
```

sample_names	<i>Sample names of a Clusnalyst dataset</i>
--------------	---

Description

Sample names of a Clusnalyst dataset

Usage

```
sample_names(d)
```

Arguments

d A `cl_data` object.

Value

Character vector.

save_session	<i>Save and restore an analysis session</i>
--------------	---

Description

A session file is JSON: the data, the metadata and the settings. Results are recomputed when the session is opened again, so nothing executable is stored.

Usage

```
save_session(d, settings, path)
```

```
load_session(path)
```

Arguments

d	The cl_data as read.
settings	The settings list.
path	File to write or read.

Value

save_session() returns path invisibly; load_session() returns a list with data and settings.

silhouette_meaning	<i>How to read an average silhouette width</i>
--------------------	--

Description

Bands from Kaufman and Rousseeuw (1990).

Usage

```
silhouette_meaning(asw)
```

Arguments

asw	Average silhouette width.
-----	---------------------------

Value

A short phrase.

theme_clusnalyst	<i>ggplot2 theme used by Clusnalyst</i>
------------------	---

Description

ggplot2 theme used by Clusnalyst

Usage

```
theme_clusnalyst(base_size = 12, dark = FALSE)
```

Arguments

base_size	Base font size.
dark	Use colours for a dark background.

Value

A ggplot2 theme.

tree_choices	<i>Tree-building methods</i>
--------------	------------------------------

Description

Tree-building methods

Usage

```
tree_choices()
```

Value

A named character vector (labels as names).

Index

align_sequences, 3
attach_metadata, 3

bootstrap_tree, 4
bootstrap_tree(), 8, 21
build_report, 5
build_script, 5
build_script(), 10
build_tree, 6
build_tree(), 4

cl_palette, 7
clusnalyst_demo (demo_info), 11
compute_distance, 7
compute_distance(), 4
consensus_tree, 8
cophenetic_cor, 8
cophenetic_table, 9

data_types, 9
data_types(), 12, 24
default_distance, 10
default_settings, 10
default_settings(), 6, 26
demo_info, 11
detect_format, 11
detect_format(), 24
distance_choices, 12
distance_choices(), 7

filter_data, 12

genotypes_to_dosage, 13
ggiraph::girafe(), 22
guess_type, 14

has_numeric, 14

load_session (save_session), 28
mantel_test, 15

mantel_test(), 19
metadata_distance, 15

optimal_k, 16
optimal_k(), 18
ordinate, 16
ordinate(), 20, 21

partition, 17
plot_heatmap, 18
plot_k, 18
plot_mantel, 19
plot_ordination, 19
plot_qc, 20
plot_scree, 21
plot_support, 21
plot_tree, 22

qc_summary, 23
qc_summary(), 20

read_fasta, 23
read_hapmap, 24
read_markers, 24
read_markers(), 25
read_metadata, 25
read_metadata(), 4
read_second_matrix, 25
read_vcf, 26
run_analysis, 26
run_analysis(), 5, 10
run_app, 27

sample_names, 27
save_session, 28
shiny::shinyApp(), 27
silhouette_meaning, 28
stats::cutree(), 16

theme_clusnalyst, 29
tree_choices, 29
tree_choices(), 6