

Package ‘disclapmix2’

July 22, 2025

Type Package

Title Mixtures of Discrete Laplace Distributions using Numerical Optimisation

Version 0.6.1

Date 2023-06-11

Description

Fit a mixture of Discrete Laplace distributions using plain numerical optimisation. This package has similar applications as the 'disclapmix' package that uses an EM algorithm.

License GPL (>= 2)

Imports Rcpp (>= 1.0.3), cluster

LinkingTo Rcpp

RoxygenNote 7.2.1

Encoding UTF-8

Suggests testthat, disclapmix, readxl

NeedsCompilation yes

Author Maarten Kruijver [aut, cre] (ORCID:

<<https://orcid.org/0000-0002-6890-7632>>),

Duncan Taylor [aut] (ORCID: <<https://orcid.org/0000-0003-0633-7424>>)

Maintainer Maarten Kruijver <maarten.kruijver@esr.cri.nz>

Repository CRAN

Date/Publication 2023-04-12 11:30:05 UTC

Contents

disclapmix2	2
haplotype_counts	3
profile_pr_by_locus_and_cluster	4
unique_haplotype_counts	5
Index	6

disclapmix2

*Discrete Laplace mixture inference using Numerical Optimisation***Description**

An extension to the **disclapmix** method in the **disclapmix** package that supports duplicated loci and other non-standard haplotypes.

Description of your package

Usage

```
disclapmix2(
  x,
  number_of_clusters,
  include_2_loci = FALSE,
  remove_non_standard_haplotypes = TRUE,
  use_stripped_data_for_initial_clustering = FALSE,
  initial_y_method = "pam",
  verbose = 0L
)
```

Arguments

<code>x</code>	DataFrame. Columns should be one character vector for each locus
<code>number_of_clusters</code>	The number of clusters to fit the model for.
<code>include_2_loci</code>	Should duplicated loci be included or excluded from the analysis?
<code>remove_non_standard_haplotypes</code>	Should observations that are not single integer alleles be removed?
<code>use_stripped_data_for_initial_clustering</code>	Should non_standard data be removed for the initial clustering?
<code>initial_y_method</code>	Which cluster method to use for finding initial central haplotypes, y: pam (recommended) or clara.
<code>verbose</code>	Set to 1 (or higher) to print optimisation details. Default is 0.

Value

List.

Author(s)

you

Examples

```
require(disclapmix)

data(danes)

x <- as.matrix(danes[rep(seq_len(nrow(danes)), danes$n), -ncol(danes)])
x2 <- as.data.frame(sapply(danes[rep(seq_len(nrow(danes)), danes$n), -ncol(danes)], as.character))

d1m_fit <- disclapmix(x, clusters = 3L)
d1m2_fit <- disclapmix2(x2, number_of_clusters = 3)

stopifnot(all.equal(d1m_fit$logL_marginal, d1m2_fit$log_lik))
```

haplotype_counts	<i>Count the number of times each haplotype occurs</i>
------------------	--

Description

Count the number of times each haplotype occurs

Usage

```
haplotype_counts(x)
```

Arguments

x DataFrame (by locus) of character vectors containing haplotypes (rows) where alleles are separated by comma's, e.g. "13,14.2" is a haplotype

Value

Integer vector with count for each row in DataFrame

Examples

```
# read haplotypes
h <- readxl::read_excel(system.file("extdata", "South_Australia.xlsx",
package = "disclapmix2"),
col_types = "text")[-c(1,2)]

# obtain counts
counts <- disclapmix2::haplotype_counts(h)

# all haplotypes in the dataset are unique
stopifnot(all(counts == 1))
```

profile_pr_by_locus_and_cluster

Compute Profile Probability from fit

Description

Compute the profile probability for a new profile that was not used in the original fit.

Usage

```
profile_pr_by_locus_and_cluster(x, fit)
```

Arguments

<code>x</code>	DataFrame. Columns should be one character vector for each locus
<code>fit</code>	Output from <code>disclapmix2</code>

Value

Numeric.

Examples

```
require(disclapmix)

data(danes)

x <- as.data.frame(sapply(danes[rep(seq_len(nrow(danes))), danes$n], ~ncol(danes)], as.character))

d1m2_fit <- disclapmix2(x, number_of_clusters = 3)

new_profile <- structure(list(DYS19 = "14", DYS389I = "13", DYS389II = "29",
                             DYS390 = "22", DYS391 = "9", DYS392 = "15", DYS393 = "13",
                             DYS437 = "14", DYS438 = "11", DYS439 = "12"),
                          row.names = 1L, class = "data.frame")

profile_pr_by_locus_and_cluster(x = new_profile, d1m2_fit)
```

`unique_haplotype_counts`*List unique haplotypes with their counts*

Description

List unique haplotypes with their counts

Usage

```
unique_haplotype_counts(x)
```

Arguments

`x` `DataFrame` (by locus) of character vectors containing haplotypes (rows) where alleles are separated by comma's, e.g. "13,14.2" is a haplotype

Value

`DataFrame` with unique rows and a `Count` column added at the end

Examples

```
# read haplotypes
h <- readxl::read_excel(system.file("extdata", "South_Australia.xlsx",
  package = "disclapmix2"),
  col_types = "text")[-c(1,2)]

# obtain counts
unique_counts <- disclapmix2::unique_haplotype_counts(h)

# all haplotypes in the dataset are unique
stopifnot(all(unique_counts$Count == 1))
```

Index

disclapmix2, [2](#)

haplotype_counts, [3](#)

profile_pr_by_locus_and_cluster, [4](#)

unique_haplotype_counts, [5](#)