Package: disclapmix2 (via r-universe)

September 4, 2024

Type Package

Title Mixtures of Discrete Laplace Distributions using Numerical Optimisation

Version 0.6.2

Date 2023-06-12

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

Repository https://mkruijver.r-universe.dev

RemoteUrl https://github.com/mkruijver/disclapmix2

RemoteRef HEAD

RemoteSha 3905f8fd1d8dc52c5c55781a3cac50e89eb9a828

Contents

`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.

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

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

Value

List.

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))


dlm_fit <- disclapmix(x, clusters = 3L)
dlm2_fit <- disclapmix2(x2, number_of_clusters = 3)

stopifnot(all.equal(dlm_fit$logL_marginal, dlm2_fit$log_lik))
```

---

haplotype_counts *Count the number of times each haplotype occurs*

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

`x`	DataFrame. Columns should be one character vector for each locus
`fit`	Output from disclapmix2

Value

Numeric.

Examples

```
require(disclapmix)

data(danes)

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

dlm2_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, dlm2_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