Package ‘PHclust’

July 21, 2025

Type Package

Title Poisson Hurdle Clustering for Sparse Microbiome Data

Version 0.1.0

Author Zhili Qiao

Maintainer Zhili Qiao <zlqiao@iastate.edu>

Description Clustering analysis for sparse microbiome data, based on a Poisson hurdle model.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

VignetteBuilder knitr

Depends R (>= 2.10)

Config/testthat/edition 3

NeedsCompilation no

Repository CRAN

Date/Publication 2022-02-08 16:20:11 UTC

Contents

Hybrid *Calculate optimal number of clusters.*

Description

This function estimates the optimal number of clusters for a given dataset.

Usage

```
Hybrid(data, absolute = FALSE, Kstart = NULL, Treatment)
```

Arguments

`data`	Data matrix with dimension N*P indicating N features and P samples.
`absolute`	Logical. Whether we should use absolute (TRUE) or relative (FALSE) abundance of features to determine clusters.
`Kstart`	Positive integer. The number of clusters for starting the hybrid merging algorithm. Should be relatively large to ensure that Kstart > optimal number of clusters. Uses *max(50, sqrt(N))* by default.
`Treatment`	Vector of length p, indicating replicates of different treatment groups. For example, *Treatment* = c(1,1,2,2,3,3) indicates 3 treatment groups, each with 2 replicates.

Value

A positive integer indicating the optimal number of clusters

Examples

```
######## Run the following codes in order:
##
## This is a sample data set which has 100 features, and 4 treatment groups with 4 replicates each.
data('sample_data')
head(sample_data)
set.seed(1)
##
## Finding the optimal number of clusters
K <- Hybrid(sample_data, Kstart = 4, Treatment = rep(c(1,2,3,4), each = 4))
##
## Clustering result from EM algorithm
result <- PHcluster(sample_data, rep(c(1,2,3,4), each = 4), K, method = 'EM', nstart = 1)
print(result$cluster)
##
## Plot the feature abundance level for each cluster
plot_abundance(result, sample_data, Treatment = rep(c(1,2,3,4), each = 4))
```

PHcluster *Poisson hurdle clustering*

Description

This function gives the clustering result based on a Poisson hurdle model.

Usage

```
PHcluster(
  data,
  Treatment,
  nK,
  method = c("EM", "SA"),
  absolute = FALSE,
  cool = 0.9,
  nstart = 1
)
```

Arguments

`data`	Data matrix with dimension N*P indicating N features and P samples. The cluster analysis is done feature-wised.
`Treatment`	Vector of length P. Indicating replicates of different treatment groups. For example, *Treatment* = c(1,1,2,2,3,3) indicates 3 treatment groups, each with 2 replicates.
`nK`	Positive integer. Number of clusters.
`method`	Method for the algorithm. Can choose between *"EM"* as Expectation Maximization or *"SA"* as Simulated Annealing.
`absolute`	Logical. Whether we should use absolute (TRUE) or relative (False) abundance of features to determine clusters.
`cool`	Real number between (0, 1). Cooling rate for the *"SA"* algorithm. Uses 0.9 by default.
`nstart`	Positive integer. Number of starts for the entire algorithm. Note that as *nstart* increases the computational time also grows linearly. Uses 1 by default.

Value

cluster Vector of length N consisting of integers from 1 to nK. Indicating final clustering result. For evaluating the clustering result please check NMI for *Normalized Mutual Information*.

prob N*nK matrix. The (i, j)th element representing the probability that observation i belongs to cluster j.

log_l Scaler. The Poisson hurdle log-likelihood of the final clustering result.

alpha Vector of length N. The geometric mean abundance level for each feature, across all treatment groups.

Normalizer vector of length P. The normalizing constant of sequencing depth for each sample.

Examples

```
######## Run the following codes in order:
##
## This is a sample data set which has 100 features, and 4 treatment groups with 4 replicates each.
data('sample_data')
head(sample_data)
set.seed(1)
##
## Finding the optimal number of clusters
K <- Hybrid(sample_data, Kstart = 4, Treatment = rep(c(1,2,3,4), each = 4))
##
## Clustering result from EM algorithm
result <- PHcluster(sample_data, rep(c(1,2,3,4), each = 4), K, method = 'EM', nstart = 1)
print(result$cluster)
##
## Plot the feature abundance level for each cluster
plot_abundance(result, sample_data, Treatment = rep(c(1,2,3,4), each = 4))
```

---

plot_abundance *Plot of feature abundance level*

Description

This function plots the feature abundance level for each cluster, after extracting the effect of sample-wise normalization factors and feature-wise geometric mean.

Usage

```
plot_abundance(result, data, Treatment)
```

Arguments

`result`	Clustering result from function PHclust().
`data`	Data matrix with dimension N*P indicating N features and P samples.
`Treatment`	Vector of length P. Indicating replicates of different treatment groups. For example, *Treatment* = c(1,1,2,2,3,3) indicates 3 treatment groups, each with 2 replicates.

Value

A plot for feature abundance level will be shown. No value is returned.

Examples

```
######## Run the following codes in order:
##
## This is a sample data set which has 100 features, and 4 treatment groups with 4 replicates each.
data('sample_data')
head(sample_data)
set.seed(1)
##
## Finding the optimal number of clusters
K <- Hybrid(sample_data, Kstart = 4, Treatment = rep(c(1,2,3,4), each = 4))
##
## Clustering result from EM algorithm
result <- PHcluster(sample_data, rep(c(1,2,3,4), each = 4), K, method = 'EM', nstart = 1)
print(result$cluster)
##
## Plot the feature abundance level for each cluster
plot_abundance(result, sample_data, Treatment = rep(c(1,2,3,4), each = 4))
```

sample_data *Sample of sparse microbiome count data*

Description

A sample data matrix with 100 features in 2 true clusters, 4 treatment groups with 4 replicates in each group.

Usage

```
sample_data
```

Format

The dataset contains 16 columns, indexed as A1 ~ A4, B1 ~ B4, C1 ~ C4, D1 ~ D4 to represent 4 treatment groups.

Examples

```
head(sample_data)
```

Index