

Package ‘metaCluster’

May 8, 2026

Type Package

Title Metagenomic Clustering

Version 0.1.1

Author Dipro Sinha [aut, cre],
Sayanti Guha Majumdar [aut],
Anu Sharma [aut],
Dwijesh Chandra Mishra [aut],
Md Yeasin [aut]

Maintainer Dipro Sinha <diprosinha@gmail.com>

Description Clustering in metagenomics is the process of grouping of microbial contigs in species specific bins. This package contains functions that extract genomic features from metagenome data, find the number of clusters for that given data and find the best clustering algorithm for binning.

License GPL-3

Encoding UTF-8

LazyData true

RoxygenNote 7.1.2

Imports factoextra, cluster, dbscan, dplyr, seqinr, Biostrings

Depends R (>= 3.6)

NeedsCompilation no

Repository CRAN

Date/Publication 2024-06-20 05:20:02 UTC

Contents

clust.suite	2
GC.content	3
metafeatures	4
oligo.freq	6
opt.clust.num	7

Index	8
--------------	----------

clust.suite *Determination of Suitable Clustering Algorithm for Metagenomics Data*

Description

This function will give the best clustering algorithm for a given metagenomics data based on silhouette index for kmeans clustering, kmedoids clustering, fuzzy kmeans clustering, DBSCAN clustering and hierarchical clustering.

Usage

```
clust.suite(data, k, eps, minpts)
```

Arguments

data	Feature matrix consisting of different genomic features. Each row represents features corresponding to a particular individual or contig and each column represents different genomic features.
k	Optimum number of clusters
eps	Radius value for DBSCAN clustering
minpts	Minimum point value of DBSCAN clustering

Value

kmeans	Output of kmeans clustering
kmedoids	Output of kmedoids clustering
fkmeans	Output of fuzzy kmeans clustering
dbscan	Output of dbscan clustering
hierarchical	Output of hierarchical clustering
silhouette.kmeans	Silhouette plot of kmeans clustering
silhouette.kmedoids	Silhouette plot of kmedoids clustering
silhouette.fkmeans	Silhouette plot of fuzzy kmeans clustering
silhouette.dbscan	Silhouette plot of dbscan clustering
silhouette.hierarchical	Silhouette plot of hierarchical clustering
best.clustering.method	Best clustering algorithm based on silhouette index
silhouette.summary	Average silhouette width of each clustering algorithm

Author(s)

Dipro Sinha <<diprosinha@gmail.com>>, Sayanti Guha Majumdar, Anu Sharma, Dwijesh Chandra Mishra

Examples

```
library(metaCluster)
data(metafeatures)
result <- clust.suite(metafeatures[1:200,], 8, 0.5, 10)
```

GC.content

Calculation of GC content

Description

This function will calculate GC content from each sequence or contigs of a FASTA file.

Usage

```
GC.content(fasta_file)
```

Arguments

fasta_file Name of the fasta or multifasta file

Value

Value of the GC content of each sequence or contig.

Author(s)

Dipro Sinha <<diprosinha@gmail.com>>, Sayanti Guha Majumdar, Anu Sharma, Dwijesh Chandra Mishra

Examples

```
library(metaCluster)
library(seqinr)
sample_data <- read.fasta(file = system.file("extdata/sample1.fasta", package = "metaCluster"),
seqtype = "DNA")
gc <- GC.content(sample_data)
```

 metafeatures

Metagenomic data

Description

Feature matrix consisting of different genomic features. Each row represents features corresponding to a particular individual or contig and each column represents different genomic features.

Usage

```
data("metafeatures")
```

Format

A data frame with 1196 observations on the following 8 variables.

```
class a factor with levels  contig-0 contig-1000000 contig-10000000 contig-100000000
  contig-1000000000 contig-1001000000 contig-1002000000 contig-1003000000 contig-1004000000
  contig-1005000000 contig-1006000000 contig-1007000000 contig-1008000000 contig-1009000000
  contig-101000000 contig-1010000000 contig-1011000000 contig-1012000000 contig-1013000000
  contig-1014000000 contig-1015000000 contig-1016000000 contig-1017000000 contig-1018000000
  contig-1019000000 contig-102000000 contig-1020000000 contig-1021000000 contig-1022000000
  contig-1023000000 contig-1024000000 contig-1025000000 contig-1026000000 contig-1027000000
  contig-1028000000 contig-1029000000 contig-103000000 contig-1030000000 contig-1031000000
  contig-1032000000 contig-1033000000 contig-1034000000 contig-1035000000 contig-1036000000
  contig-1037000000 contig-1038000000 contig-1039000000 contig-104000000 contig-1040000000
  contig-1041000000 contig-1042000000 contig-1043000000 contig-1044000000 contig-1045000000
  contig-1046000000 contig-1047000000 contig-1048000000 contig-1049000000 contig-105000000
  contig-1050000000 contig-1051000000 contig-1052000000 contig-1053000000 contig-1054000000
  contig-1055000000 contig-1056000000 contig-1057000000 contig-1058000000 contig-1059000000
  contig-106000000 contig-1060000000 contig-1061000000 contig-1062000000 contig-1063000000
  contig-1064000000 contig-1065000000 contig-1066000000 contig-1067000000 contig-1068000000
  contig-1069000000 contig-107000000 contig-1070000000 contig-1071000000 contig-1072000000
  contig-1073000000 contig-1074000000 contig-1075000000 contig-1076000000 contig-1077000000
  contig-1078000000 contig-1079000000 contig-108000000 contig-1080000000 contig-1081000000
  contig-1082000000 contig-1083000000 contig-1084000000 contig-1085000000 contig-1086000000
  contig-1087000000 contig-1088000000 contig-1089000000 contig-109000000 contig-1090000000
  contig-1091000000 contig-1092000000 contig-1093000000 contig-1094000000 contig-1095000000
  contig-1096000000 contig-1097000000 contig-1098000000 contig-1099000000 contig-11000000
  contig-110000000 contig-1100000000 contig-1101000000 contig-1102000000 contig-1103000000
  contig-1104000000 contig-1105000000 contig-1106000000 contig-1107000000 contig-1108000000
  contig-1109000000 contig-111000000 contig-1110000000 contig-1111000000 contig-1112000000
  contig-1113000000 contig-1114000000 contig-1115000000 contig-1116000000 contig-1117000000
  contig-1118000000 contig-1119000000 contig-112000000 contig-1120000000 contig-1121000000
  contig-1122000000 contig-1123000000 contig-1124000000 contig-1125000000 contig-1126000000
  contig-1127000000 contig-1128000000 contig-1129000000 contig-113000000 contig-1130000000
  contig-1131000000 contig-1132000000 contig-1133000000 contig-1134000000 contig-1135000000
```



```

contig-1352000000 contig-1353000000 contig-1354000000 contig-1355000000 contig-1356000000
contig-1357000000 contig-1358000000 contig-1359000000 contig-1360000000 contig-1360000000
contig-1361000000 contig-1362000000 contig-1363000000 contig-1364000000 contig-1365000000
contig-1366000000 contig-1367000000 contig-1368000000 contig-1369000000 contig-1370000000
contig-1370000000 contig-1371000000 contig-1372000000 contig-1373000000 contig-1374000000
contig-1375000000 contig-1376000000 contig-1377000000 contig-1378000000 contig-1379000000
contig-1380000000 contig-1380000000 contig-1381000000 contig-1382000000 contig-1383000000
contig-1384000000 contig-1385000000 contig-1386000000 contig-1387000000 contig-1388000000
contig-1389000000 contig-1390000000 contig-1390000000 contig-1391000000 contig-1392000000
contig-1393000000 contig-1394000000 contig-1395000000 contig-1396000000 contig-1397000000
contig-1398000000 contig-1399000000 contig-1400000000 contig-1400000000 contig-1400000000
contig-1401000000 contig-1402000000 contig-1403000000 contig-1404000000 contig-1405000000
contig-1406000000 contig-1407000000 contig-1408000000 contig-1409000000 contig-1410000000
contig-1410000000 contig-1411000000 contig-1412000000 contig-1413000000 contig-1414000000
contig-1415000000 contig-1416000000 contig-1417000000 contig-1418000000 contig-1419000000
contig-1420000000 contig-1420000000 contig-1421000000 contig-1422000000 contig-1423000000
contig-1424000000 contig-1425000000 contig-1426000000 contig-1427000000 contig-1428000000
contig-1429000000 contig-1430000000 contig-1430000000 contig-1431000000 contig-1432000000
contig-1433000000 contig-1434000000 contig-1435000000 contig-1436000000 contig-1437000000
contig-1438000000 contig-1439000000 contig-1440000000 contig-1440000000 contig-1441000000
contig-1442000000 contig-1443000000 contig-1444000000 contig-1445000000 contig-1446000000
contig-1447000000

```

Dim.1 a numeric vector

Dim.2 a numeric vector

Dim.3 a numeric vector

Dim.4 a numeric vector

Dim.5 a numeric vector

Dim.6 a numeric vector

gc a numeric vector

oligo.freq

Oligonucleotide Frequency

Description

This function will calculate oligonucleotide frequency of each sequence or contig from a FASTA file.

Usage

```
oligo.freq(fasta_file, f)
```

Arguments

fasta_file	Name of the fasta or multifasta file
f	Length of the oligonucleotide

Value

Frequency value of each oligonucleotide of length specified by the user

Author(s)

Dipro Sinha <<diprosinha@gmail.com>>, Sayanti Guha Majumdar, Anu Sharma, Dwijesh Chandra Mishra

Examples

```
library(metaCluster)
freq <- oligo.freq(fasta_file = system.file("extdata/sample1.fasta", package = "metaCluster"), 4)
```

opt.clust.num *Finding Optimum Number of Cluster for Metagenomics Data*

Description

This function will give optimum number of clusters based on Within Sum of Squares (wss) plot.

Usage

```
opt.clust.num(data, nc, seed = 1234)
```

Arguments

data	Feature matrix consisting of different genomic features. Each row represents features corresponding to a particular individual or contig and each column represents different genomic features.
nc	Probable number of clusters
seed	Seed value for iteration

Value

WSS plot

Author(s)

Dipro Sinha <<diprosinha@gmail.com>>, Sayanti Guha Majumdar, Anu Sharma, Dwijesh Chandra Mishra

Examples

```
library(metaCluster)
data(metafeatures)
wss_plot <- opt.clust.num(metafeatures[1:200,], nc=10, seed = 1234)
```

Index

- * **Binning**
 - clust.suite, 2
 - * **DBSCAN**
 - clust.suite, 2
 - * **Fuzzy-kmeans**
 - clust.suite, 2
 - * **GC content**
 - GC.content, 3
 - * **Hierarchical**
 - clust.suite, 2
 - * **Metagenomics**
 - clust.suite, 2
 - * **WSS**
 - opt.clust.num, 7
 - * **datasets**
 - metafeatures, 4
 - * **kmeans**
 - clust.suite, 2
 - * **kmedoids**
 - clust.suite, 2
 - * **oligonucleotide frequency**
 - oligo.freq, 6
- clust.suite, 2
- GC.content, 3
- metafeatures, 4
- oligo.freq, 6
- opt.clust.num, 7