Package 'MetChem'

July 6, 2023

Version 0.4

Date 2023-07-05

Maintainer Stefano Cacciatore <tkcaccia@gmail.com>

Title Chemical Structural Similarity Analysis

Description A new pipeline to explore chemical structural similarity across metabolite. It allows to classify metabolites in structurally-related modules and identify common shared functional groups. KODAMA algorithm is used to highlight structural similarity between metabolites. See Cacciatore S, Tenori L, Luchinat C, Bennett PR, MacIntyre DA. (2017) Bioinformatics <doi:10.1093/bioinformatics/btw705>, Cacciatore S, Luchinat C, Tenori L. (2014) Proc Natl Acad Sci USA <doi:10.1073/pnas.1220873111>, and Abdel-Shafy EA, Melak T, MacIntyre DA, Zadra G, Zerbini LF, Piazza S, Cacciatore S. (2023) Bioinformatics Advances <doi:10.1093/bioadv/vbad053>.

Depends R (>= 3.5.0), stats, KODAMA (>= 2.3), httr, XML, fingerprint, rcdk (>= 3.4.3)

Suggests knitr, rmarkdown, readxl, impute, pheatmap, RColorBrewer

VignetteBuilder knitr

SuggestsNote No suggestions

License GPL (>= 2)

NeedsCompilation no

Repository CRAN

Author Ebtesam Abdel-Shafy [aut],
Tadele Melak [aut],
David A. MacIntyre [aut],
Giorgia Zadra [aut],
Luiz F. Zerbini [aut],
Silvano Piazza [aut],
Stefano Cacciatore [aut, cre]

Date/Publication 2023-07-06 07:43:05 UTC

R topics documented:

`allbranches` *Cut a Tree into Groups of Data*

Description

Cuts a tree as resulting from `hclust` function, into groups (a.k.a. modules).

Usage

```
allbranches(hh,minlen=5)
```

Arguments

`hh`	a tree as produced by `hclust` function.
`minlen`	The minimum number of elements in each module.

Value

A list contains vectors of module memberships.

See Also

`cutree`, `hclust`, `clusters.detection`

Examples

```
data(Metabolites)

data=Metabolites$readMet$concentration
hh=hclust(dist(data),method="ward.D")
res=allbranches(hh)
```

`chemical.dissimilarity` *Chemical dissimilarity.*

Description

This function calculates the structural dissimilarity between different metabolites using the simplified molecular-input line-entry system (SMILE) of each metabolite as input.

Usage

```
chemical.dissimilarity (smiles,method="tanimoto",type="extended")
```

Arguments

`smiles`	A vector of smile notations.
`method`	The method used to calculated the distance between molecular fingerprint ("tanimoto" as default). For more information see `fp.sim.matrix` function.
`type`	The type of fingerprint applied to the SMILEs ("extended" as default). For more information see `get.fingerprint` function.

Value

A list contains distance between fingerprints .

See Also

`fp.sim.matrix`, `get.fingerprint`,

Examples

```
data(Metabolites)
d=chemical.dissimilarity(Metabolites$SMILES[1:50])
```

ChemRICH *ChemRICH Dataset*

Description

This dataset consists of a list of the metabolites names download from https://chemrich.fiehnlab.ucdavis.edu/.
HMDB IDs were retrieved from PubChem Identifier Exchange Service (https://pubchem.ncbi.nlm.nih.gov/idexchange/idexch
and manually curated.

Usage

```
data(ChemRICH)
```

Value

A list with the following elements in the variable `ChemRICH`:

`name`	A vector of metabolite's names.
`SMILES`	A vector of SMILES represenation of each metabolite.
`HMDB`	A vector containing HMDB IDs of each metabolite.

Examples

```
data(ChemRICH)
```

clusters.detection *Detection of clusters.*

Description

This function calculates the structural similarity between different metabolites and perform hierarchical clustering using the KODAMA algorithm and detect the optimal number of clusters. The procedure is repeated to ensure the robustness of the detection.

Usage

```
clusters.detection  (smiles,
                     repetition=10,
                     k=50,
                     seed=12345,
                     max_nc = 30,
                     dissimilarity.parameters=list(),
                     kodama.matrix.parameters=list(),
```

```
                    kodama.visualization.parameters=list(),
                    hclust.parameters=list(method="ward.D"),
                    verbose = TRUE)
```

Arguments

smiles	A list of smile notations for the study metabolites dataset.
repetition	The number of time the KODAMA analysis is repeated.
k	A number of components of multidimensional scaling.
seed	Seed for the generation of random numbers.
max_nc	Maximum number of clusters.
dissimilarity.parameters	Optional parameters for chemical.dissimilarity function.
kodama.matrix.parameters	Optional parameters for KODAMA.matrix function.
kodama.visualization.parameters	Optional parameters for KODAMA.visualization function.
hclust.parameters	Optional parameters for hclust function.
verbose	If verbose is TRUE, it displays the progress for each iteration.

Value

A list contains all results of KODAMA chemical similarity analysis and hierarchical clustering.

See Also

KODAMA.matrix

Examples

```
data(Metabolites)

res=clusters.detection(Metabolites$SMILES)
```

diseasesMet *Metabolite-associated Diseases*

Description

This function correlates metabolites to associated diseases.

Usage

```
diseasesMet(doc)
```

Arguments

doc	A list of metabolites information produced by readMet function.

Value

A data frame contains the diseases associated with each metabolite.

See Also

pathwaysMet, taxonomyMet, enzymesMet

Examples

```
data(Metabolites)
dis=diseasesMet(Metabolites$readMet)
```

enzymesMet *Metabolite-associated Enzymes*

Description

This function finds the metabolite related enzymes.

Usage

```
enzymesMet(doc)
```

Arguments

doc	A list of metabolites information produced by readMet function.

Value

A data frame contains the enzymes associated with each metabolite.

See Also

pathwaysMet , taxonomyMet, diseasesMet

Examples

```
data(Metabolites)
enz=enzymesMet(Metabolites$readMet)
```

---

features *Cluster features extraction*

Description

This function finds features associated with each cluster.

Usage

```
features(doc,cla,cl,HMDB_ID)
```

Arguments

doc	The output of the readMet function.
cla	The output of diseasesMet, enzymesMet, pathwaysMet, propertiesMet, substituentsMet, or taxonomyMet functions.
cl	The output of the allbranches function containing the module memberships.
HMDB_ID	A vector of HMDB IDs associated with their chemical name.

Value

A list of p-value calculated using Fisher test for cluster associted features.

See Also

KODAMA.chem.sim, tree.cutting, substituentsMet

Examples

```
data(Metabolites)
SMILES=Metabolites$SMILES
names(SMILES)=Metabolites$name
HMDB=Metabolites$HMDB
names(HMDB)=Metabolites$name
res=KODAMA.chem.sim(SMILES)
cl=allbranches(res$hclust)
cla=substituentsMet(Metabolites$readMet)
f=features(Metabolites$readMet,cla,cl,HMDB)
```

HFD *HFD Dataset*

Description

This dataset is dataframe of metabolite dataset contains only chemical information.

Usage

```
data(HFD)
```

Value

A list with the following elements in the variable HFD:

SMILES	A vector of SMILES represenation of each metabolite.
CHEMICAL_ID	A vector of chemical ID number or each metabolite.
PUBCHEM	A vector of identifier ID number from PUBCHEM database for chemical molecules and their activities in biological assays.
CHEMSPIDER	A vector of a unique identifier from CHEMSPIDER database each molecule.
HMDB	A vector containing HMDB IDs of each metabolite.

Examples

```
data(HFD)
```

KODAMA.chem.sim *KODAMA chemical similarity.*

Description

This function calculates the structural similarity between different metabolites and perform hierarchical clustering using the KODAMA algorithm.

Usage

```
KODAMA.chem.sim (smiles,
                 d=NULL,
                 k=50,
                 dissimilarity.parameters=list(),
                 kodama.matrix.parameters=list(),
                 kodama.visualization.parameters=list(),
                 hclust.parameters=list(method="ward.D"))
```

Arguments

smiles	A list of smile notations for the study metabolites dataset.
d	A distance structure such as that returned by dist or a full symmetric matrix containing the dissimilarities. If NULL (default), then the dissimilarity matrix will be generated by chemical.dissimilarity function. Otherwise, d will be considered as the dissimilarity matrix.
k	A number of components of multidimensional scaling.
dissimilarity.parameters	Optional parameters for chemical.dissimilarity function.
kodama.matrix.parameters	Optional parameters for KODAMA.matrix function.
kodama.visualization.parameters	Optional parameters for KODAMA.visualization function.
hclust.parameters	Optional parameters for hclust function.

Value

A list contains all results of KODAMA chemical similarity analysis and hierarchical clustering for the KODAMA dimensions.

See Also

KODAMA.matrix

Examples

```
data(Metabolites)

res=KODAMA.chem.sim(Metabolites$SMILES)
plot(res$kodama$visualization)
```

`Metabolites` *Metabolomic Dataset*

Description

This dataset consists of a list of the metabolites as returned by the function `readMet` and concentration value of each metabolites.

Usage

```
data(Metabolites)
```

Value

A list with the following elements in the variable `Metabolites`:

`concentration`	A matrix containing the concentration of each metabolites.
`name`	A vector of metabolite's names.
`SMILES`	A vector of SMILES represenation of each metabolite.
`HMDB`	A vector containing HMDB IDs of each metabolite.
`readMet`	A list of metabolites information produced by `readMet` function.

Examples

```
data(Metabolites)
```

nameMet *Name of metabolites*

Description

This function extracts the metabolite's names from the list generated by readMet function.

Usage

```
nameMet(doc)
```

Arguments

doc	A list of metabolites information produced by readMet function.

Value

A data frame contains the names of each metabolite.

See Also

readMet

Examples

```
data(Metabolites)
nam=nameMet(Metabolites$readMet)
```

pathwaysMet *Metabolic Pathways*

Description

This function finds the metabolite related pathways.

Usage

```
pathwaysMet(doc)
```

Arguments

doc	A list of metabolites information produced by readMetfunction.

Value

A data frame contains the pathways associated with each metabolite.

See Also

readMet, taxonomyMet, enzymesMet, diseasesMet

Examples

```
data(Metabolites)
pat=pathwaysMet(Metabolites$readMet)
```

---

propertiesMet *Physical Proprieties of metabolites*

Description

This function finds the Physical Proprieties of metabolites.

Usage

```
propertiesMet(doc)
```

Arguments

doc	A list of metabolites information produced by readMet function.

Value

A data frame contains the properties associated with each metabolite.

See Also

readMet, taxonomyMet, substituentsMet, propertiesMet

Examples

```
data(Metabolites)
pro=propertiesMet(Metabolites$readMet)
```

readMet *Metabolite Cards Reading*

Description

This function extract metabocards of your metabolites dataset from http://www.hmdb.ca/metabolites/ database and store all of this information in a list.

Usage

```
readMet(ID, address =  c("http://www.hmdb.ca/metabolites/"),remove=TRUE)
```

Arguments

ID	A vector containg the HMDBcodes (i.e., metabolite IDs) of metabolites dataset.
address	Optional address where the MetaboCards are located. The default address is http://www.hmdb.ca/metabolites/.
remove	A logic value. If true, missing and wrong HMDB IDs are removed.

Value

A list containing all the information related to the metabocards.

See Also

nameMet

Examples

```
ID=c("HMDB0000122","HMDB0000124","HMDB0000243","HMDB0000263")
doc=readMet(ID)
```

selectionMet *Metabolites selection*

Description

This function select metabolites from the list generated by readMet function.

Usage

```
selectionMet(doc, sel)
```

Arguments

doc	A list of metabolites information produced by readMet function.
sel	A vector of metabolite's HMDBcode that will be selected

Value

doc	A doc list contains only the selcted metabolites.

See Also

readMet, nameMet

Examples

```
data(Metabolites)
doc=selectionMet(Metabolites$readMet,c("HMDB0000299","HMDB0000881"))
nameMet(doc)
```

---

substituentsMet *Metabolite substituents*

Description

This function finds the metabolite related substituents.

Usage

```
substituentsMet(doc)
```

Arguments

doc	A list of metabolites information produced by readMet function

.

Value

A data frame contains the substituents of each metabolite.

See Also

readMet, nameMet, propertiesMet

Examples

```
data(Metabolites)
sub=substituentsMet(Metabolites$readMet)
```

taxonomyMet *Metabolite Taxonomy*

Description

This function finds the metabolite related taxonomy.

Usage

```
taxonomyMet(doc)
```

Arguments

doc	A list of metabolites information produced by readMet function.

Value

A data frame contains the taxonomy of each metabolite.

See Also

readMet, propertiesMet, enzymesMet, diseasesMet

Examples

```
data(Metabolites)
tax=taxonomyMet(Metabolites$readMet)
```

tree.cutting *Optimal cluster number calculation.*

Description

This function helps to estimate the optimal cluster number that fit the metabolites dataset. It applies different optimal cluster number calculating algorithms to cut clutering tree of hclust function. and return a list contains index corresponding to each cluster number.

Usage

```
tree.cutting (res,max_nc=20)
```

Arguments

res	A list produced by KODAMA.chem.sim function.
max_nc	The maximum number of cluster (default = 20).

Value

A list contains the calculation for each clustering of Rousseeuw's Silhouette index.

See Also

KODAMA.chem.sim, WMCSA

Examples

```
data(Metabolites)

res=KODAMA.chem.sim(Metabolites$SMILES)
clu=tree.cutting(res,max_nc = 30)
plot(clu$min_nc:clu$max_nc,clu$res.S)
```

WMCSA *Weighted Metabolite Chemical Structural Analysis*

Description

Summarize metabolites concetration in each of identified clusters using the module eigenvalue (eigen-metabolite) for calculating module membership measures.

Usage

```
WMCSA(data,cl)
```

Arguments

data	dataset of different metabolite concentration in differnt samples.
cl	The output of the allbranches function containing the module memberships.

Value

This function returns a matrix as output represent similarity score of metabolites within the same module among different samples.

See Also

KODAMA.chem.sim, tree.cutting

Examples

```
data(Metabolites)

SMILES=Metabolites$SMILES
names(SMILES)=Metabolites$name
res=KODAMA.chem.sim(SMILES)
cl=allbranches(res$hclust)
ww=WMCSA(Metabolites$concentration,cl)
```

`write.cls` *Write a CLS file*

Description

This function write a file in the format CLS defined by GenePattern.

Usage

```
write.cls(es, address)
```

Arguments

`es`	A matrix.
`address`	The address of the file should be saved.

Value

No return value. If an invalid address is inserted, the function will generate an error.

See Also

`write.gmt`, `write.gct`

write.gct *Write a GCT file*

Description

This function write a file in the format GCT defined by GenePattern.

Usage

```
write.gct(es, address)
```

Arguments

es	A matrix.
address	The address of the file should be saved.

Value

No return value. If an invalid address is inserted, the function will generate an error.

See Also

write.gmt, write.cls

write.gmt *Write a GMT file*

Description

This function write a file containing the Metabolite Set informtation in the format GMT defined by GenePattern.

Usage

```
write.gmt(sub,address,min_entry=2,max_entry=50)
```

Arguments

sub	A matrix.
address	The address of the file should be saved.
min_entry	The minimum number of metabolites for each metabolite set.
max_entry	The maximum number of metabolites for each metabolite set.

Value

No return value. If an invalid address is inserted, the function will generate an error.

See Also

write.gct, write.cls

Index