Package 'chemodiv'

January 12, 2023

Title Analysing Chemodiversity of Phytochemical Data

Version 0.2.0

Description Quantify and visualise various measures of chemical diversity and dissimilarity, for phytochemical compounds and other sets of chemical composition data. Importantly, these measures can incorporate biosynthetic and/or structural properties of the chemical compounds, resulting in a more comprehensive quantification of diversity and dissimilarity. For details, see Petrén, Köllner and Junker (2023) <doi:10.1111/nph.18685>.

License GPL (>= 3)

Encoding UTF-8

LazyData true

RoxygenNote 7.2.3

Suggests knitr, rmarkdown, testthat (>= 3.0.0)

Config/testthat/edition 3

Depends R (>= 2.10)

biocViews

Imports jsonlite, httr, vegan, webchem, fmcsR, ChemmineR, hillR, ape, GUniFrac, tidygraph, igraph, ggraph, ggplot2, gridExtra, ggdendro, tidyr, rlang, curl

VignetteBuilder knitr

URL https://github.com/hpetren/chemodiv

BugReports https://github.com/hpetren/chemodiv/issues

NeedsCompilation no

Author Hampus Petrén [aut, cre] (<https://orcid.org/0000-0001-6490-4517>), Tobias G. Köllner [aut] (<https://orcid.org/0000-0002-7037-904X>), Robert R. Junker [aut] (<https://orcid.org/0000-0002-7919-9678>)

Maintainer Hampus Petrén <hampus.petren@gmail.com>

Repository CRAN

Date/Publication 2023-01-12 09:30:06 UTC

R topics documented:

`alpinaCompData` *Arabis alpina floral scent compounds*

Description

A dataset listing the compounds in `alpinaSampData`.

Usage

```
alpinaCompData
```

Format

A data frame with 15 rows and 3 columns. Each row is a compound. First column is a common name of the compound, second column is the SMILES (Simplified Molecular-Input Line-Entry System) specification, third column is the InChIKey (International Chemical Identifier).

Source

Petren H, Torang P, Agren J, Friberg M. 2021. Evolution of floral scent in relation to self-incompatibility and capacity for autonomous self-pollination in the perennial herb *Arabis alpina*. Annals of Botany 127: 737-747.

`alpinaCompDis` *Arabis alpina floral scent compound dissimilarity matrix*

Description

A matrix with compound dissimilarities calculated using `compDis` with `type = "PubChemFingerprint"`, for the compounds in `alpinaCompData`.

Usage

```
alpinaCompDis
```

Format

A 15x15 compound dissimilarity matrix.

`alpinaMolNet` *Arabis alpina floral scent molecular network*

Description

A molecular network. Generated by the `molNet` function used on the `alpinaCompDis` dataset, with `cutOff = 0.75`.

Usage

```
alpinaMolNet
```

Format

A `tbl_graph` object with 15 nodes and 56 edges.

alpinaNPCTable *Arabis alpina floral scent NPClassifier table*

Description

A table with the NPClassifier pathways, superclasses and classes, along with the compound names, smiles and inchikey. Generated by the NPCTable function used on the alpinaCompData dataset.

Usage

```
alpinaNPCTable
```

Format

A dataframe with 15 compounds and their NPClassifier classifications.

alpinaPopData *Arabis alpina populations*

Description

A dataset listing what population each sample in alpinaSampData comes from.

Usage

```
alpinaPopData
```

Format

A data frame with 87 rows and 1 column. Each row represents the population each sample in alpinaSampData comes from (It8, S1 or G1).

Source

Petren H, Torang P, Agren J, Friberg M. 2021. Evolution of floral scent in relation to self-incompatibility and capacity for autonomous self-pollination in the perennial herb *Arabis alpina*. Annals of Botany 127: 737-747.

alpinaSampData *Arabis alpina floral scent data*

Description

A dataset with proportional floral scent data from three populations of the plant Arabis alpina.

Usage

```
alpinaSampData
```

Format

A data frame with 87 rows and 15 columns. Each row is a sample, each column is a floral scent compound.

Source

Petren H, Torang P, Agren J, Friberg M. 2021. Evolution of floral scent in relation to self-incompatibility and capacity for autonomous self-pollination in the perennial herb *Arabis alpina*. Annals of Botany 127: 737-747.

alpinaSampDis *Arabis alpina floral scent sample dissimilarity matrix*

Description

A matrix with sample dissimilarities calculated from `alpinaSampData` and `alpinaCompDis` using `sampDis` with `type = "GenUniFrac"` and `alpha = 0.5`.

Usage

```
alpinaSampDis
```

Format

A 87x87 sample dissimilarity matrix.

`calcBetaDiv` *Calculate beta diversity*

Description

Function to calculate beta diversity in the Hill diversity framework. This can be calculated as Hill beta diversity or Functional Hill beta diversity.

Usage

```
calcBetaDiv(sampleData, compDisMat = NULL, type = "HillDiv", q = 1)
```

Arguments

`sampleData`	Data frame with the relative concentration of each compound (column) in every sample (row).
`compDisMat`	Compound dissimilarity matrix, as calculated by `compDis`. Has to be supplied for calculations of Functional Hill beta diversity.
`type`	Type(s) of Hill beta diversity to calculate. `"HillDiv"` and/or `"FuncHillDiv"`.
`q`	Diversity order to use for the calculation of beta diversity. See `calcDiv` for further details on q.

Details

The function calculates a single beta diversity value for the supplied `sampleData`. This is calculated as *beta = gamma / alpha*. Gamma diversity represents the diversity of the pooled data set, alpha diversity represents the mean diversity across individual samples, and beta diversity represents turnover or variability among samples. With `type = "HillDiv"` and `q = 0` the calculated beta diversity is equal to the well-known and most simple measure of beta diversity introduced by Whittaker 1960, where *beta = gamma / alpha*, based only on the number of species (here compounds).

Value

Data frame with type of Hill beta diversity calculated, q, and values for gamma diversity, mean alpha diversity and beta diversity.

References

Chao A, Chiu C-H, Jost L. 2014. Unifying Species Diversity, Phylogenetic Diversity, Functional Diversity, and Related Similarity and Differentiation Measures Through Hill Numbers. Annual Review of Ecology, Evolution, and Systematics 45: 297-324.

Jost L. 2007. Partitioning diversity into independent alpha and beta components. Ecology 88: 2427-2439.

Whittaker RH. 1960. Vegetation of the Siskiyou Mountains, Oregon and California. Ecological Monographs 30: 279-338.

Examples

```
data(minimalSampData)
data(minimalCompDis)
calcBetaDiv(sampleData = minimalSampData)
calcBetaDiv(sampleData = minimalSampData, compDisMat = minimalCompDis,
type = c("HillDiv", "FuncHillDiv"), q = 2)

data(alpinaSampData)
data(alpinaCompDis)
calcBetaDiv(sampleData = alpinaSampData, compDisMat = alpinaCompDis,
type = "FuncHillDiv")
```

`calcDiv` *Calculate various diversity and evenness measures*

Description

Function to calculate different common measures of diversity and evenness. This includes Hill diversity, Functional Hill diversity, Shannon's diversity, Simpson diversity, Rao's Q, Pielou's evenness and Hill evenness.

Usage

```
calcDiv(sampleData, compDisMat = NULL, type = "HillDiv", q = 1)
```

Arguments

`sampleData`	Data frame with the relative concentration of each compound (column) in every sample (row).
`compDisMat`	Compound dissimilarity matrix, as calculated by `compDis`. Has to be supplied for calculations of Functional Hill Diversity and Rao's Q.
`type`	Type(s) of diversity or evenness to calculate. Any of `"Shannon"`, `"Simpson"`, `"HillDiv"`, `"FuncHillDiv"`, `"RaoQ"`,`"PielouEven"`, `"HillEven"`.
`q`	Diversity order to use for Hill diversity, Functional Hill Diversity and Hill Evenness. q should be equal to or larger than zero. This parameter determines the sensitivity of the (Functional) Hill Diversity measure to the relative frequencies of compounds. Commonly set to 0, 1 or 2, although any value > 0 may be used. For `q = 0` compound proportions are not taken into account. For `q = 1` (default) compounds are weighed according to their proportion in the sample. For `q = 2`, more weight is put on compounds with high proportions.

Details

The function calculates diversity and/or evenness for each sample in `sampleData`. It can calculate the following indices:

- `Shannon`. Shannon's Diversity.

- `Simpson`. Simpson Diversity, often referred to as the Inverse Simpson Index.
- `HillDiv`. Hill Diversity. Equation 4a/4b in Chao et al. 2014. Also referred to as the Hill number or the effective number of species (here compounds). The parameter q determines the sensitivity of the measure to the relative frequencies of compounds (see above for details). For `q = 0`, this equals the number of compounds in a sample. For `q = 1`, this equals the exponential of Shannon's Diversity. For `q = 2`, this equals the Simpson Diversity.
- `FuncHillDiv`. Functional Hill Diversity. Equation 4b/6b in Chiu & Chao 2014, which is the measure called "total functional diversity". Requires a compound dissimilarity matrix. Functional Hill Diversity quantifies the effective total dissimilarity between compounds in the sample. The parameter q determines the sensitivity of the measure to the relative frequencies of compounds (see above for details). For `q = 1`, this is a measure sensitive to compound richness, evenness and dissimilarity, and is therefore the most comprehensive measure of diversity. For `q = 0`, this is equal to Functional Attribute Diversity (FAD) which is the sum of all dissimilarities in the dissimilarity matrix. FAD divided by *n(n-1)*, where n is the number of compounds and hence the number of rows/columns in the dissimilarity matrix, is equal to the Mean Pairwise Dissimilarity (MPD). This value is the mean of the pairwise dissimilarities in the compound dissimilarity matrix (excluding the 0 values in the diagonal), and is therefore in contrast to FAD not dependent on the number of compounds.
- `RaoQ`. Rao's quadratic entropy index Q. The perhaps most common measure of functional diversity. Requires a compound dissimilarity matrix. Rao's Q represents the average dissimilarity of two randomly selected (weighed by their proportions) compounds in the sample.
- `PielouEven`. Pielou's Evenness, also referred to as Shannon's equitability. This is perhaps the most common evenness measure. Equal to the Shannon's Diversity divided by the natural logarithm of the number of compounds. In other words, it expresses evenness with the observed Shannon's diversity as a proportion of the maximum Shannon's diversity where all compounds are equally abundant. Therefore, this is a relative measure with a minimum value of 0 and a maximum value of 1. This measure of evenness is not replication invariant.
- `HillEven`. Hill Evenness, as defined by equation 8 in Tuomisto 2012. This is equal to the Hill Diversity, at a given value of q, divided by the number of compounds, and therefore has a minimum value of 1 / number of compounds and maximum value of 1. This measure of evenness is replication invariant. This measure can be normalized to range from 0 to 1 (equation 13 in Tuomisto 2012).

Value

Data frame with calculated diversity/evenness values for each sample.

References

Chao A, Chiu C-H, Jost L. 2014. Unifying Species Diversity, Phylogenetic Diversity, Functional Diversity, and Related Similarity and Differentiation Measures Through Hill Numbers. Annual Review of Ecology, Evolution, and Systematics 45: 297-324.

Chiu C-H, Chao A. 2014. Distance-Based Functional Diversity Measures and Their Decomposition: A Framework Based on Hill Numbers. PLoS ONE 9: e100014.

Hill MO. 1973. Diversity and Evenness: A Unifying Notation and Its Consequences. Ecology 54: 427-432.

Tuomisto H. 2012. An updated consumer's guide to evenness and related indices. Oikos 121: 1203-1218

Examples

```
data(minimalSampData)
data(minimalCompDis)
calcDiv(sampleData = minimalSampData)
calcDiv(sampleData = minimalSampData, type = c("HillDiv", "HillEven"))
calcDiv(sampleData = minimalSampData, compDisMat = minimalCompDis,
type = "FuncHillDiv", q = 2)

data(alpinaSampData)
data(alpinaCompDis)
calcDiv(sampleData = alpinaSampData, compDisMat = alpinaCompDis,
type = "FuncHillDiv")
```

`calcDivProf` *Calculate a diversity profile*

Description

Function to calculate a diversity profile, i.e. calculate Hill diversity or Functional Hill Diversity for a range of q values.

Usage

```
calcDivProf(
  sampleData,
  compDisMat = NULL,
  type = "HillDiv",
  qMin = 0,
  qMax = 3,
  step = 0.1
)
```

Arguments

`sampleData`	Data frame with the relative concentration of each compound (column) in every sample (row).
`compDisMat`	Compound distance matrix, as calculated by `compDis`. Has to be supplied for calculations of Functional Hill diversity.
`type`	Type of Hill Diversity to calculate for the diversity profile. `"HillDiv"` or `"FuncHillDiv"`.
`qMin`	Minimum value of q.
`qMax`	Maximum value of q.
`step`	Increment by which q will be calculated between `qMin` and `qMax`.

Details

The function calculates a diversity profile for each sample in `sampleData`. A diversity profile is a calculation of Hill Diversity or Functional Hill Diversity for a range of different values of q. This function performs the calculations, while `chemoDivPlot` can be used to conveniently create the diversity profile plot, where Hill Diversity is plotted as a function of q within the chosen range. The shape of the diversity profile curve reflects the evenness of compound proportions in the sample. For a perfectly even sample the curve is flat. The more uneven the compound proportions are, the more steep is the decline of the curve. A common range, used as default, of q values is between `qMin = 0` and `qMax = 3`, as diversity should change little beyond `qMax = 3`. See `calcDiv` for further details on q.

Value

List with a diversity profile data frame with samples as rows and the Hill diversity or Functional Hill diversity for different q values as columns; and values for `type`, `qMin`, `qMax` and `step`.

References

Chao A, Chiu C-H, Jost L. 2014. Unifying Species Diversity, Phylogenetic Diversity, Functional Diversity, and Related Similarity and Differentiation Measures Through Hill Numbers. Annual Review of Ecology, Evolution, and Systematics 45: 297-324.

Examples

```
data(minimalSampData)
data(minimalCompDis)
calcDivProf(sampleData = minimalSampData)
calcDivProf(sampleData = minimalSampData, compDisMat = minimalCompDis,
type = "FuncHillDiv")

data(alpinaCompData)
data(alpinaCompDis)
calcDivProf(sampleData = alpinaSampData, compDisMat = alpinaCompDis,
type = "FuncHillDiv", qMin = 1, qMax = 2, step = 0.2)
```

`chemodiv` *chemodiv: A package for analysing phytochemical diversity*

Description

chemodiv is an R package for analysing the chemodiversity of phytochemical data. The package includes a number of functions that enables quantification and visualization of phytochemical diversity and dissimilarity for any type of phytochemical (and similar) samples, such as herbivore defence compounds, volatiles and similar. Importantly, calculations of diversity and dissimilarity can incorporate biosynthetic and/or structural properties of the phytochemical compounds, resulting in more comprehensive quantifications of diversity and dissimilarity. Functions in the R-package will work best for sets of data, commonly generated by chemical ecologists using GC-MS, LC-MS or similar, where all or most compounds in the samples have been confidently identified. See Petren et al. 2023 for a detailed description of the package.

Details

Two datasets are needed to use the full set of analyses included in the package.

The first dataset should contain data on the relative abundance/concentration (i.e. proportion) of different compounds (columns) in different samples (rows). See the included dataset `minimalSampData` for a basic example. Note that all calculations of diversity, and most calculations of dissimilarity, are only performed on relative, rather than absolute, values.

The second dataset should contain, in each of three columns in a data frame, the compound name, SMILES and InChIKey IDs of all the compounds present in the first dataset. See the included dataset `minimalCompData` for a basic example. SMILES and InChIKey are chemical identifiers that are easily obtained for each compound by searching for it in PubChem https://pubchem.ncbi.nlm.nih.gov/. Here, a search with a common name will bring up the compound's record in the database, where the (isomeric/canonical) SMILES and InChIKey are included. Various automated tools such as the PubChem Identifier Exchange Service https://pubchem.ncbi.nlm.nih.gov/idexchange/idexchange.cgi or The Chemical Translation Service https://cts.fiehnlab.ucdavis.edu/ can also be used. The user is intentionally required to compile the chemical identifiers manually to ensure these are correct, as lists of compounds very often contain compounds wrongly named, wrongly formatted, under various synonyms etc. which prevents easy automatic translation of compound names to SMILES and InChIKey. Note that SMILES IDs might contain the character combination `"\C"`. If SMILES are entered manually directly in R, this is interpreted as an unrecognized escape and results in an error. In this case, an extra backslash has to be added: `"\\C"`. If the dataset is instead imported into R as a csv-file or txt-file (recommended), this is done automatically and no manual edits has to be done.

The second dataset with the chemical IDs is primarily used to construct one or more dissimilarity matrices with pairwise dissimilarities between chemical compounds, which can then be used in calculations of phytochemical diversity and dissimilarity. As noted above, to do this, the compounds in the samples have to be identified and their chemical IDs listed. If some compounds in a dataset are unknown, these can be handled in different ways decided by the user, see `compDis` for details. If many or all compounds are unknown, as is common for more metabolomic type datasets, phytochemical diversity and dissimilarity can still be calculated using indices that do not consider compound dissimilarities. Alternatively, other ways to calculate compound dissimilarities, not based on knowing compound identities, can be used. For example, cosine dissimilarities between tandem (MS/MS) mass spectra of metabolomic features can be calculated in the GNPS framework https://gnps.ucsd.edu (Wang et al. 2016). A dissimilarity matrix of such dissimilarities can then be used for the `compDisMat` argument in various functions in the package, thereby enabling comprehensive quantification of phytochemical diversity and dissimilarity also for datasets consisting of unidentified compounds.

Once the dataset with samples and the dataset with compounds are prepared, these should be imported/constructed as separate data frames in R, and all analyses in the package can then be performed, in largely the same order as they appear in the list below.

Data format checks

`chemoDivCheck`

Compound classification and dissimilarity

`NPCTable` `compDis`

Diversity calculations

calcDiv calcBetaDiv calcDivProf

Sample dissimilarities

sampDis

Molecular network and properties

molNet

Chemodiversity and network plots

molNetPlot chemoDivPlot

Shortcut function

quickChemoDiv

Author(s)

Hampus Petren, Tobias G. Koellner, Robert R. Junker

References

Petren H., Koellner T.G., Junker R.R. 2023. Quantifying chemodiversity considering biochemical and structural properties of compounds with the R package *chemodiv*. New Phytologist doi: 10.1111/nph.18685.

Wang M, Carver JJ, Phelan VV, et al. 2016. Sharing and community curation of mass spectrometry data with Global Natural Products Social Molecular Networking. Nature Biotechnology 34: 828-837.

See Also

https://github.com/hpetren/chemodiv

---

chemoDivCheck *Check data formatting*

Description

Function to check that the datasets used by other functions in the *chemodiv* package are correctly formatted.

Usage

```
chemoDivCheck(sampleData, compoundData)
```

Arguments

`sampleData`	Data frame with the relative concentration of each compound (column) in every sample (row).
`compoundData`	Data frame with the compounds in `sampleData` as rows. Should have a column named "compound" with common names of the compounds, a column named "smiles" with SMILES IDs of the compounds, and a column named "inchikey" with the InChIKey IDs for the compounds.

Details

The function performs a number of checks on the two main datasets used as input data, to make sure datasets are formatted in a way suitable for the other functions in the package. This should make it easier for users to correctly construct datasets before starting with analyses.

Two datasets are needed to use the full set of analyses included in the package, and these can be checked for formatting issues. The first dataset should contain data on the proportions of different compounds (columns) in different samples (rows). Note that all calculations of diversity, and most calculations of dissimilarity, are only performed on relative, rather than absolute, values. The second dataset should contain, in each of three columns in a data frame, the compound name, SMILES and InChIKey IDs of all the compounds present in the first dataset. See `chemodiv` for details on obtaining SMILES and InChIKey IDs. Avoid including Greek letters or other special characters in the compound names.

Value

One or several messages pointing out problems with data formatting, or a message informing that the datasets appear to be correctly formatted.

Examples

```
data(minimalSampData)
data(minimalCompData)
chemoDivCheck(minimalSampData, minimalCompData) # Correct format
chemoDivCheck(minimalSampData, minimalCompData[c(2,3,1),]) # Incorrect format

data(alpinaSampData)
data(alpinaCompData)
chemoDivCheck(sampleData = alpinaSampData, compoundData = alpinaCompData)
```

`chemoDivPlot` *Plot chemodiversity*

Description

Function to conveniently create basic plots of the different types of chemodiversity measurements calculated by functions in the package.

Usage

```
chemoDivPlot(
  compDisMat = NULL,
  divData = NULL,
  divProfData = NULL,
  sampDisMat = NULL,
  groupData = NULL
)
```

Arguments

compDisMat	Compound dissimilarity matrix, generated by the compDis function. Note that only a single matrix should be supplied, and not the whole list.
divData	Diversity/evenness data frame, generated by the calcDiv function. This data frame can contain a single or multiple columns with diversity/evenness measures.
divProfData	Diversity profile, generated by the calcDivProf function. Note that the whole list outputted by the calcDivProf function should be supplied.
sampDisMat	Sample dissimilarity matrix, generated by the sampDis function. This can be either the list of one or both matrices outputted by the function, or a single matrix directly.
groupData	Grouping data. Should be either a vector or a data frame with a single column.

Details

The function can create four different types of plots, (using ggplot2) depending on which input data is supplied:

- Function argument compDisMat. A compound dissimilarity matrix will be plotted as a dendrogram visualizing how structurally/biosynthetically similar different compounds are to each other.
- Function argument divData. Diversity/evenness values will be plotted as a boxplot.
- Function argument divProfData. A diversity profile, plotting (Functional) Hill diversity at different values of q will be plotted as a line plot.
- Function argument sampDisMat. A sample dissimilarity matrix will be plotted as an NMDS plot.
- Function argument groupData. Grouping data (e.g. population, species etc.) may be supplied, to plot each group in different boxes/lines/colours.

Note that this function can take any combination of the four arguments as input, and argument names should always be specified to ensure each dataset is correctly plotted. If including the function argument sampDisMat, a Nonmetric Multidimensional Scaling (NMDS) will be performed, which may take time for larger datasets.

Value

The specified chemodiversity plots.

Examples

```
minimalDiv <- calcDiv(minimalSampData, minimalCompDis, type = "FuncHillDiv")
groups <- c("A", "A", "B", "B")
chemoDivPlot(divData = minimalDiv, groupData = groups)

data(alpinaCompDis)
data(alpinaSampDis)
data(alpinaPopData)
alpinaDiv <- calcDiv(sampleData = alpinaSampData, compDisMat = alpinaCompDis,
type = "FuncHillDiv")
alpinaDivProf <- calcDivProf(sampleData = alpinaSampData,
compDisMat = alpinaCompDis, type = "FuncHillDiv",
qMin = 0, qMax = 2, step = 0.2)
chemoDivPlot(compDisMat = alpinaCompDis, divData = alpinaDiv,
divProfData = alpinaDivProf, sampDisMat = alpinaSampDis,
groupData = alpinaPopData)
```

`compDis` *Calculate compound dissimilarities*

Description

Function to quantify dissimilarities between phytochemical compounds.

Usage

```
compDis(
  compoundData,
  type = "PubChemFingerprint",
  npcTable = NULL,
  unknownCompoundsMean = FALSE
)
```

Arguments

`compoundData`	Data frame with the chemical compounds of interest, usually the compounds found in the sample dataset. Should have a column named "compound" with common names of the compounds, a column named "smiles" with SMILES IDs of the compounds, and a column named "inchikey" with the InChIKey IDs for the compounds.
`type`	Type of data compound dissimilarity calculations will be based on: `NPClassifier`, `PubChemFingerprint` or `fMCS`. If more than one is chosen, a matrix with mean values of the other matrices will also calculated.
`npcTable`	A data frame already generated by `NPCTable` can optionally be supplied, if compound dissimilarities are to be calculated using `type = "NPClassifier"`.
`unknownCompoundsMean`	If unknown compounds, i.e. ones without SMILES or InChIKey, should be given mean dissimilarity values. If not, these will have dissimilarity 1 to all other compounds.

Details

This function calculates matrices with pairwise dissimilarities between the chemical compounds in `compoundData`, to quantify how different the molecules are to each other. It does so in three different ways, based on the biosynthetic classification or molecular structure of the molecules:

1. Using the classification from the *NPClassifier* tool, `type = "NPClassifier"`. *NPClassifier* (Kim et al. 2021) is a deep-learning tool that automatically classifies natural products (i.e. phytochemical compounds) into a hierarchical classification of three levels: pathway, superclass and class. This classification largely corresponds to the biosynthetic groups/pathways the compounds are produced in. Classifications are downloaded from https://npclassifier.ucsd.edu/. *NPClassifier* does not always manage to classify every compound into all three hierarchical levels. In such cases, it might be beneficial to first run `NPCTable`, manually edit the resulting data frame with probable classifications if possible (with help from the Supporting Information in Kim et al. 2021), and then supply this classification to the `compDis` function with the `npcTable` argument. This will ensure that compound dissimilarities are computed optimally.
2. Using PubChem Fingerprints, `type = "PubChemFingerprint"`. This is a binary substructure fingerprint with 881 binary variables describing the chemical structure of a compound. With this method, compounds are therefore compared based on how structurally dissimilar the molecules are. See https://pubchem.ncbi.nlm.nih.gov/docs/data-specification for more information. (There are many other types of fingerprints, and ways of calculating compound dissimilarities based on them, see e.g. packages `fingerprint` and `rcdk`). Fingerprint data for molecules is downloaded from PubChem. In association with this, there might be a Warning message about closing unused connections, which is not important.
3. fMCS, flexible Maximum Common Substructure, `type = "fMCS"`. This is a pairwise graph matching concept. The fMCS of two compounds is the largest substructure that occurs in both compounds allowing for atom and/or bond mismatches (Wang et al 2013). As with the fingerprints, compounds are compared based on how structurally dissimilar the molecules are. While potentially a very accurate similarity measure, fMCS is much more computationally demanding than the other methods, and will take a significant amount of time for larger data sets. Data on molecules is downloaded from PubChem. In association with this, there might be a Warning message about closing unused connections, which is not important.

Dissimilarities using NPClassifier and PubChem Fingerprints are generated by calculating Jaccard (Tanimoto) dissimilarities from a 0/1 table with compounds as rows and group (NPClassifier) or binary fingerprint variable (PubChem Fingerprints) as columns. fMCS generates dissimilarity values by calculating Jaccard dissimilarities based on the number of atoms in the maximum common substructure, allowing for one atom and one bond mismatch. Dissimilarities are outputted as dissimilarity matrices.

If dissimilarities are calculated with more than one method, the function will output additional dissimilarity matrices. This always includes a matrix with the mean dissimilarity values of the selected methods. If `"NPClassifier"` is included in `type`, a matrix of "mix" values is also calculated. The values in this matrix are the dissimilarities from NPClassifier when these are > 0. For pairs of compounds where dissimilarities from NPClassifier equals 0 (i.e. when the compounds belong to the same pathway, superclass and class), values are equal to half of the (mean) value(s) of the structural dissimilarity/-ies from PubChem Fingerprints and/or fMCS. With this method, compound dissimilarities are primarily based on NPClassifier, but instead of compounds with identical classification having 0 dissimilarity, these have a dissimilarity based on PubChem Fingerprints and/or fMCS,

scaled to always be less (< 0.5) than compounds being in the same pathway and superclass, but different class.

If there are unknown compounds, which do not have a corresponding SMILES or InChIKey, this can be handled in three different ways. First, these can be completely removed from the list of compounds and the sample data set, and hence excluded from all analyses. Second, if `unknownCompoundsMean = FALSE`, unknown compounds will be given a dissimilarity value of 1 to all other compounds. Third, if `unknownCompoundsMean = TRUE`, unknown compounds will be given a dissimilarity value to all other compounds which equals the mean dissimilarity value between all known compounds. See `chemodiv` for alternative methods that can be used when most or all compounds are unknown.

Value

List with compound dissimilarity matrices. A list is always outputted, even if only one matrix is calculated. Downstream functions, including `calcDiv`, `calcBetaDiv`, `calcDivProf`, `sampDis`, `molNet` and `chemoDivPlot` require only the matrix as input (e.g. as `fullList$specificMatrix`) rather than the whole list.

References

Kim HW, Wang M, Leber CA, Nothias L-F, Reher R, Kang KB, van der Hooft JJJ, Dorrestein PC, Gerwick WH, Cottrell GW. 2021. NPClassifier: A Deep Neural Network-Based Structural Classification Tool for Natural Products. Journal of Natural Products 84: 2795-2807.

Wang Y, Backman TWH, Horan K, Girke T. 2013. fmcsR: mismatch tolerant maximum common substructure searching in R. Bioinformatics 29: 2792-2794.

Examples

```
data(minimalCompData)
data(minimalNPCTable)
compDis(minimalCompData, type = "NPClassifier",
npcTable = minimalNPCTable) # Dissimilarity based on NPClassifier

## Not run: compDis(minimalCompData) # Dissimilarity based on Fingerprints

data(alpinaCompData)
data(alpinaNPCTable)
compDis(compoundData = alpinaCompData, type = "NPClassifier",
npcTable = alpinaNPCTable) # Dissimilarity based on NPClassifier
```

`minimalCompData` *Minimal compound dataset*

Description

A small dataset with three phytochemical compounds.

Usage

```
minimalCompData
```

Format

A data frame with 3 rows and 3 columns. Each row is a phytochemical compound. First column is a common name of the compound, second column is the SMILES (Simplified Molecular-Input Line-Entry System) specification, third column is the InChIKey (International Chemical Identifier).

---

minimalCompDis *Minimal compound dissimilarity matrix*

Description

A matrix with compound dissimilarities calculated using `compDis` with `type = "PubChemFingerprint"`, for the compounds in `minimalCompData`.

Usage

```
minimalCompDis
```

Format

A 3x3 compound dissimilarity matrix.

---

minimalMolNet *Minimal molecular network*

Description

A molecular network. Generated by the `molNet` function used on the `minimalCompDis` dataset, with `cutOff = "median"`.

Usage

```
minimalMolNet
```

Format

A `tbl_graph` object with 3 nodes and 4 edges.

minimalNPCTable *Minimal NPClassifier table*

Description

A table with the NPClassifier pathways, superclasses and classes, along with the compound names, smiles and inchikey. Generated by the `NPCTable` function used on the `minimalCompData` dataset.

Usage

```
minimalNPCTable
```

Format

A dataframe with 3 compounds and their NPClassifier classifications.

minimalSampData *Minimal sample dataset*

Description

A small made up dataset with phytochemical data.

Usage

```
minimalSampData
```

Format

A data frame with 4 rows and 3 columns. Each row is a sample, each column is a phytochemical compound.

minimalSampDis *Minimal sample dissimilarity matrix*

Description

A matrix with sample dissimilarities calculated from `minimalSampData` and `minimalCompDis` using `sampDis` with `type = "GenUniFrac"` and `alpha = 0.5`.

Usage

```
minimalSampDis
```

Format

A 4x4 sample dissimilarity matrix.

molNet *Generate a molecular network with some properties*

Description

Function to generate a molecular network object, and some basic properties of the network.

Usage

```
molNet(compDisMat, npcTable = NULL, cutOff = "median")
```

Arguments

compDisMat	Compound dissimilarity matrix, as calculated by compDis. Note that the supplied dissimilarity matrix is transformed into a similarity matrix, and this is what cutOff values are set for. Note also that compDis always outputs a list of one or more matrices, while molNet requires a single matrix as input. Therefore, a specific matrix has to be selected from this list, as compDisOutput$matrix.
npcTable	A data frame generated by NPCTable can be supplied for calculations of the number of NPC pathways and network modularity.
cutOff	Cut-off value for compound similarities. Any similarity lower than this value will be set to zero when the network is generated, which strongly affects the look of the network. The value can be set manually to any value between 0 and 1; to the median similarity value from the compDisMat; or, if an NPCTable is supplied, to minPathway, the lowest within-pathway similarity (which allows all within-NPC-pathway similarities to be kept).

Details

Molecular networks can be used to illustrate the biosynthetic/structural similarity of phytochemical compounds in a sample, while simultaneously visualizing their relative concentrations. molNet creates the network, and molNetPlot can subsequently be used to create a plot of the network.

Value

List with a (tbl_graph) graph object, the number of compounds, number of NPC pathways and a measure of the modularity of the network (see modularity).

Examples

```
data(minimalCompDis)
data(minimalNPCTable)
molNet(minimalCompDis, minimalNPCTable, cutOff = 0)

data(alpinaCompDis)
molNet(compDisMat = alpinaCompDis, cutOff = 0.75)
```

molNetPlot *Plot molecular networks*

Description

Function to conveniently create a basic plot of the molecular network created by the molNet function. Molecular networks can be used to illustrate the biosynthetic/structural similarity of phytochemical compounds in a sample, while simultaneously visualizing their relative concentrations. In the network, nodes are compounds, with node sizes or node colours representing the relative concentrations of compounds. Edges connects nodes, with edge widths representing compound similarity.

Usage

```
molNetPlot(
  sampleData,
  networkObject,
  groupData = NULL,
  npcTable = NULL,
  plotNames = FALSE,
  layout = "kk"
)
```

Arguments

sampleData	Data frame with the relative concentration of each compound (column) in every sample (row).
networkObject	A network object, as created by the molNet function. Note that this is only the network object, which is one of the elements in the list outputted by molNet. The network is extracted as molNetOutput$networkObject.
groupData	Grouping data (e.g. population, species etc.). If supplied, a separate network will be created for each group. Should be either a vector, or a data frame with a single column.
npcTable	It is optional but recommended to supply an NPCTable. This will result in network nodes being coloured by their NPC pathway classification.
plotNames	Indicates if compounds names should be included in the molecular network plot.
layout	Layout used by ggraph when creating the network. The default chosen here, "kk", is the the Kamada-Kawai layout algorithm which in most cases should produce a visually pleasing network. Another useful option is "circle", which puts all nodes in a circle, for easier comparisons between different networks.

Details

The network object from molNet and sampleData have to be supplied. In addition, groupData and/or an NPCTable can be supplied. If an NPCTable is supplied, which is recommended, node colours will represent NPC pathways, and node sizes the relative concentration of the compounds.

Edge widths represent compound similarity, and only edges with similarity values above the `cutOff` value in the `molNet` function will be plotted. If `groupData` is supplied, one network will be created for each group. When `groupData` but not an `NPCTable` is supplied, compounds missing (i.e. having a mean of 0) from specific groups are plotted as triangles. When `groupData` and an `NPCTable` is supplied, compounds missing from specific groups have a white fill. Additionally, in both cases, edges connecting to missing compounds are lighter coloured. These graphical styles are done so that networks are more easy to compare across groups.

Value

A plot with one or more molecular networks.

Examples

```
data(minimalSampData)
data(minimalNPCTable)
data(minimalMolNet)
groups <- c("A", "A", "B", "B")
molNetPlot(minimalSampData, minimalMolNet)
molNetPlot(minimalSampData, minimalMolNet, groups)
molNetPlot(minimalSampData, minimalMolNet, plotNames = TRUE)

data(alpinaSampData)
data(alpinaPopData)
data(alpinaMolNet)
data(alpinaNPCTable)
molNetPlot(sampleData = alpinaSampData, networkObject = alpinaMolNet,
npcTable = alpinaNPCTable)
```

NPCTable *Generate NPClassifier classification*

Description

Function to classify compounds with *NPClassifier*, and put the results in a data frame containing the pathway, superclass and class for each compound.

Usage

```
NPCTable(compoundData)
```

Arguments

`compoundData`	Data frame with the chemical compounds of interest, usually the compounds found in the sample dataset. Should include a column named "compound" with common names of the compounds and a column named "smiles" with SMILES IDs of the compounds.

Details

NPClassifier (Kim et al. 2021) is a deep-learning tool that automatically classifies natural products (i.e. phytochemical compounds) into a hierarchical classification of three levels: pathway, superclass and class. This classification largely corresponds to the biosynthetic groups/pathways the compounds are produced in. The `NPCTable` function conveniently performs this classification directly in R on the compounds in `compoundData`, by accessing the tool at https://npclassifier.ucsd.edu/ and downloading the classifications.

Value

Data frame with the NPClassifier classification for each compound as pathway, superclass and class. Note that compounds may be classified in more than one group, or no group, at each level of classification.

References

Kim HW, Wang M, Leber CA, Nothias L-F, Reher R, Kang KB, van der Hooft JJJ, Dorrestein PC, Gerwick WH, Cottrell GW. 2021. NPClassifier: A Deep Neural Network-Based Structural Classification Tool for Natural Products. Journal of Natural Products 84: 2795-2807.

Examples

```
data(minimalCompData)
NPCTable(minimalCompData)

data(alpinaCompData)
NPCTable(compoundData = alpinaCompData[1:3,]) # First three compounds only
```

---

quickChemoDiv *Quickly calculate or plot chemodiversity*

Description

This function is a shortcut that makes use of many of the other functions in the package. In one simple step chemodiversity is calculated, and if requested also plotted, for users wanting to quickly explore their data using standard parameters.

Usage

```
quickChemoDiv(
  sampleData,
  compoundData = NULL,
  groupData = NULL,
  outputType = "plots"
)
```

Arguments

`sampleData`	Data frame with the relative concentration of each compound (column) in every sample (row).
`compoundData`	Data frame with the compounds in `sampleData` as rows. Should have a column named "compound" with common names of the compounds, a column named "smiles" with SMILES IDs of the compounds, and a column named "inchikey" with the InChIKey IDs for the compounds. See `chemodiv` for details on obtaining SMILES and InChIKey IDs.
`groupData`	Grouping data (e.g. population, species etc.). Should be either a vector, or a data frame with a single column.
`outputType`	Type of output that should be returned: either `data` to output a list with different types of chemodiversity data, or `plots` to instead produce standard plots of this data.

Details

The function requires sample data as input, and can also include compound data. `chemoDivCheck` can be used to ensure these datasets are correctly formatted, see `chemodiv` for further details on data formatting. If only sample data is supplied, phytochemical diversity and dissimilarity will be calculated as Hill diversity and Bray-Curtis dissimilarity, respectively. If sample data and compound data is supplied, phytochemical diversity and dissimilarity will be calculated as Functional Hill diversity and Generalized UniFrac dissimilarity, respectively. This function then uses the following other functions in the package:

- `compDis` is used to calculate compound dissimilarities using PubChem Fingerprints, if compound data is supplied.
- `calcDiv` is used to calculate (Functional) Hill Diversity for q = 1.
- `calcDivProf` is used to calculate a diversity profile with (Functional) Hill Diversity for q = 0-3.
- `sampDis` is used to calculate Bray-Curtis or Generalized UniFrac dissimilarities between samples.
- `chemoDivPlot` is used to create different chemodiversity plots if requested.

`quickChemoDiv` is designed to provide an easy way to calculate and visualize the most central measures of phytochemical diversity. It uses default parameters to do so, which should be reasonable in most cases. However, for detailed analyses it is recommended to use the separate functions to allow for full control of function input, arguments and output.

Value

Different types of chemodiversity measures, either as elements in a list or as separate plots. If `outputType = "data"`, function returns a compound dissimilarity matrix (if compound data was supplied), a data frame with (Functional) Hill Diversity at q = 1, a data frame with a (Functional) Hill Diversity profile for q = 0-3, and a sample dissimilarity matrix. If `outputType = "plots"`, these data sets are plotted as a dendrogram (if compound data was supplied), a boxplot, a diversity profile plot and an NMDS plot, respectively.

Examples

```
data(minimalCompData)
data(minimalSampData)
groups <- c("A", "A", "B", "B")
quickChemoDiv(sampleData = minimalSampData, groupData = groups,
outputType = "data") # Without compound data

data(alpinaSampData)
data(alpinaPopData)
quickChemoDiv(sampleData = alpinaSampData, outputType = "plots",
groupData = alpinaPopData) # Without compound data
```

`sampDis` *Calculate sample dissimilarities*

Description

Function to calculate dissimilarities between samples. Either Bray-Curtis dissimilarities and/or Generalized UniFrac dissimilarities are calculated.

Usage

```
sampDis(sampleData, compDisMat = NULL, type = "BrayCurtis", alpha = 1)
```

Arguments

`sampleData`	Data frame with the relative concentration of each compound (column) in every sample (row).
`compDisMat`	Compound dissimilarity matrix, as calculated by `compDis`. If this is supplied, Generalized UniFrac dissimilarities can be calculated.
`type`	Type of sample dissimilarities to be calculated. This is Bray-Curtis dissimilarities, `type = "BrayCurtis"`, and/or Generalized UniFrac dissimilarities, `type = "GenUniFrac"`.
`alpha`	Parameter used in calculations of Generalized UniFrac dissimilarities. alpha can be set between 0 and 1. With `alpha = 0`, equal weight is put on every branch in the dendrogram. With `alpha = 1`, branches are weighted by their abundance, and hence more emphasis is put on high abundance branches. `alpha = 0.5` strikes a balance between the two. alpha 0.5 or 1 is recommended, with `alpha = 1` as default. See Chen et al. 2012 for details.

Details

The function calculates a dissimilarity matrix for all the samples in `sampleData`, for the given dissimilarity index/indices. Bray-Curtis dissimilarities are calculated using only the `sampleData`. This is the most commonly calculated dissimilarity index used for phytochemical data (other types of dissimilarities are easily calculated using the `vegdist` function in the `vegan` package).

If a compound dissimilarity matrix, `compDisMat`, is supplied, Generalized UniFrac dissimilarities can be calculated, which also use the compound dissimilarity matrix for the sample dissimilarity calculations. For the calculation of Generalized UniFrac dissimilarities (Chen et al. 2012), the compound dissimilarity matrix is transformed into a dendrogram using hierarchical clustering (with the UPGMA method). Calculations of UniFrac dissimilarities quantifies the fraction of the total branch length of the dendrogram that leads to compounds present in either sample, but not both. The (weighted) Generalized UniFrac dissimilarities implemented here additionally take compound abundances into account. In this way, both the relative proportions of compounds and the biosynthetic/structural dissimilarities of the compounds are accounted for in the calculations of sample dissimilarities, such that two samples containing more biosynthetically/structurally different compounds have a higher pairwise dissimilarity than two samples containing more biosynthetically/structurally similar compounds. As with Bray-Curtis dissimilarities, Generalized UniFrac dissimilarities range in value from 0 to 1.

Value

List with sample dissimilarity matrices. A list is always outputted, even if only one matrix is calculated.

References

Bray JR, Curtis JT. 1957. An Ordination of the Upland Forest Communities of Southern Wisconsin. Ecological Monographs 27: 325-349.

Chen J, Bittinger K, Charlson ES, et al. 2012. Associating microbiome composition with environmental covariates using generalized UniFrac distances. Bioinformatics 28: 2106-2113.

Lozupone C, Knight R. 2005. UniFrac: a New Phylogenetic Method for Comparing Microbial Communities. Applied and Environmental Microbiology 71: 8228-8235.

Examples

```
data(minimalSampData)
data(minimalCompDis)
sampDis(minimalSampData)
sampDis(sampleData = minimalSampData, compDisMat = minimalCompDis,
type = c("BrayCurtis", "GenUniFrac"), alpha = 0.5)

data(alpinaSampData)
data(alpinaCompDis)
sampDis(sampleData = alpinaSampData, compDisMat = alpinaCompDis,
type = "GenUniFrac")
```

Index