Package 'IDSL.SUFA'

March 21, 2022

Type Package

Title Simplified UFA

Version 1.0

Depends R (>= 3.5)

Author Sadjad Fakouri-Baygi [cre, aut]
(<https://orcid.org/0000-0002-6864-6911>),
Dinesh Barupal [aut] (<https://orcid.org/0000-0002-9954-8628>)

Maintainer Sadjad Fakouri-Baygi <sadjad.fakouri-baygi@mssm.edu>

Description
A simplified version of the 'IDSL.UFA' package to calculate isotopic profiles and adduct formulas from molecular formulas with no dependency on other R packages for online tools.

License MIT + file LICENSE

URL https://ufa.idsl.me, https://github.com/idslme/idsl.ufa

BugReports https://github.com/idslme/idsl.ufa/issues

Encoding UTF-8

LazyData true

Archs i386, x64

NeedsCompilation no

Repository CRAN

Date/Publication 2022-03-21 08:10:04 UTC

R topics documented:

`element_sorter` *Element Sorter*

Description

This function sorts 84 elements in the periodic table for molecular formula deconvolution and isotopic profile calculation.

Usage

```
element_sorter(ElementList = "all", ElementOrder = "alphabetical")
```

Arguments

`ElementList`	A string vector of elements needed for isotopic profile calculation. The default value for this parameter is a vector string of entire elements.
`ElementOrder`	ElementOrder = c("alphabetical", "same") where "alphabetical" should be used to sort the elements for elemental deconvolution (default value), "same" should be used to keep the input order.

Value

`OutputElements`	A string vector of elements (alphabetically sorted or unsorted)
`Elements_mass_abundance`	A list of isotopic mass and abundance of elements.
`valence`	A vector of electron valences.

Examples

```
EL_mass_abundance_val <- element_sorter()
```

`formula_adduct_calculator` *Formula Adduct Calculator*

Description

a function that takes a formula and an vecotr of ionization pathways and returns the adduct formulas.

Usage

```
formula_adduct_calculator(molecular_formula, IonPathways)
```

Arguments

molecular_formula	molecular formula
IonPathways	A vector of ionization pathways. Pathways should be like [Coeff*M+ADD1-DED1+...] where "Coeff" should be an integer between 1-9 and ADD1 and DED1 may be ionization pathways. ex: 'IonPathways <- c("[M]+", "[M+H]+", "[2M-Cl]-", "[3M+CO2-H2O+Na-KO2+HCl-NH4]-")'

Value

A vector of adduct formulas

Examples

```
molecular_formula = "C15H10O7"
IonPathways = c("[M+]","[M+H]","[M+H20+H]","[M+Na]")
Formula_adducts <- formula_adduct_calculator(molecular_formula, IonPathways)
```

---

formula_vector_generator

Molecular Formula Vector Generator

Description

This function convert a molecular formulas into a numerical vector

Usage

```
formula_vector_generator(molecular_formula, Elements, L_Elements = length(Elements))
```

Arguments

molecular_formula	molecular formula
Elements	a string vector of elements. This value must be driven from the 'element_sorter' function.
L_Elements	number of elements. To speed up loop calculations, consider to calculate number of elements outside of the loop.

Value

a numerical vector for the molecular formula. This function returns a vector of -Inf values when the molecular formula has elements not listed in the 'Elements' string vector.

Examples

```
molecular_formula <- "C12H2Br5Cl3O"
Elements_molecular_formula <- c("C", "H", "O", "Br", "Cl")
EL <- element_sorter(ElementList = Elements_molecular_formula)
Elements <- EL[[1]]
L_Elements <- length(Elements)
mol_vec <- formula_vector_generator(molecular_formula, Elements, L_Elements)
```

`ionization_pathway_deconvoluter` *Ionization Pathway Deconvoluter*

Description

This function deconvolutes ionization pathways into a coefficient and a numerical vector to simplify prediction ionization pathwyas.

Usage

```
ionization_pathway_deconvoluter(IonPathways, Elements)
```

Arguments

`IonPathways`	A vector of ionization pathways. Pathways should be like [Coeff*M+ADD1-DED1+...] where "Coeff" should be an integer between 1-9 and ADD1 and DED1 may be ionization pathways. ex: 'IonPathways <- c("[M]+", "[M+H]+", "[2M-Cl]-", "[3M+CO2-H2O+Na-KO2+HCl-NH4]-")'
`Elements`	A vector string of the used elements

Value

A list of adduct calculation values for each ionization pathway.

Examples

```
Elements <- element_sorter()[[1]]
IonPathways <- c("[M]+", "[M+H]+", "[2M-Cl]-", "[3M+CO2-H2O+Na-KO2+HCl-NH4]-")
Ion_DC <- ionization_pathway_deconvoluter(IonPathways, Elements)
```

`isotopic_profile_calculator` *Isotopic Profile Calculator*

Description

This function was designed to calculate isotopic profile distributions for small molecules with masses <= 1200 Da. Details of the equations used in this function are available in the reference[1]. In this function, neighboring isotopologues are merged using the satellite clustering merging (SCM) method described in the reference[2].

Usage

```
isotopic_profile_calculator(MoleFormVec, Elements_mass_abundance, peak_spacing,
intensity_cutoff, UFA_IP_memeory_variables = c(1e30, 1e-12))
```

Arguments

`MoleFormVec`	A numerical vector of the molecular formula
`Elements_mass_abundance`	A list of isotopic mass and abundance of elements obtained from the 'element_sorter' function
`peak_spacing`	A maximum space between two isotopologues in Da
`intensity_cutoff`	A minimum intesnity threshold for isotopic profiles in percentage
`UFA_IP_memeory_variables`	A vector of three variables. Default values are c(1e30, 1e-12). Memory may be an issue when the entire isotopologues are calculated; therefore, memeory_variables[1] is used to adjust memory usage. memeory_variables[2] indicates the minimum relative abundance (RA calcaulated by eq(1) in the reference [1]) of an isotopologue to include in the isotopic profile calculations.

Value

A matrix of isotopic profile. The first and second column represents the mass and instensity profiles, respectively.

References

[1] Fakouri Baygi, S., Crimmins, B.S., Hopke, P.K. Holsen, T.M. (2016). Comprehensive emerging chemical discovery: novel polyfluorinated compounds in Lake Michigan trout. *Environmental Science and Technology*, 50(17), 9460-9468, doi: 10.1021/acs.est.6b01349.

[2] Fakouri Baygi, S., Fernando, S., Hopke, P.K., Holsen, T.M. and Crimmins, B.S. (2019). Automated Isotopic Profile Deconvolution for High Resolution Mass Spectrometric Data (APGC-QToF) from Biological Matrices. *Analytical chemistry*, 91(24), 15509-15517, doi: 10.1021/acs.analchem.9b03335.

See Also

https://ipc.idsl.me/

Examples

```
EL <- element_sorter()
Elements <- EL[[1]]
Elements_mass_abundance <- EL[[2]]
peak_spacing <- 0.005 # mDa
intensity_cutoff <- 1 # (in percentage)
MoleFormVec <- formula_vector_generator("C8H10N4O2", Elements)
IP <- isotopic_profile_calculator(MoleFormVec, Elements_mass_abundance,
                                  peak_spacing, intensity_cutoff)
```

---

isotopic_profile_molecular_formula_feeder_simplified

Isotopic Profile Molecular Formula Feeder

Description

A function to calculate isotopic profiles from a molecular formulas

Usage

```
isotopic_profile_molecular_formula_feeder_simplified(molecular_formula,
IonPathways = "[M]+", peak_spacing = 0, intensity_cutoff = 1,
UFA_IP_memeory_variables = c(1e30, 1e-12)
)
```

Arguments

molecular_formula	A molecular formulas
IonPathways	An ionization pathways. Pathways should be like [Coeff*M+ADD1-DED1+...] where "Coeff" should be an integer between 1-9 and ADD1 and DED1 may be ionization pathways. ex: 'IonPathways <- c("[M]+", "[M+H]+", "[2M-Cl]-", "[3M+CO2-H2O+Na-KO2+HCl-NH4]-")'
peak_spacing	A maximum space between isotopologues in Da to merge neighboring isotoplogues.
intensity_cutoff	A minimum intesnity threshold for isotopic profiles in percentage.
UFA_IP_memeory_variables	A vector of two variables. Default values are c(1e30, 1e-12). Memory may be an issue when the entire isotopologues are calculated; therefore, memeory_variables[1] is used to adjust memory usage. memeory_variables[2] indicates the minimum relative abundance (RA calcaulated by eq(1) in the reference [1]) of an isotopologue to include in the isotopic profile calculations.

Value

A list of isotopic profiles

References

[1] Fakouri Baygi, S., Crimmins, B.S., Hopke, P.K. Holsen, T.M. (2016). Comprehensive emerging chemical discovery: novel polyfluorinated compounds in Lake Michigan trout. *Environmental Science and Technology*, 50(17), 9460-9468, doi: 10.1021/acs.est.6b01349.

See Also

https://ipc.idsl.me/

Examples

```
molecular_formula <- "C12Cl10"
peak_spacing <- 0.005 # in Da for QToF instruments
# Use this piece of code for intensity cutoff to preserve significant isotoplogues
intensity_cutoff <- 1
IonPathways <- "[M+H]+"
isotopic_profile <- isotopic_profile_molecular_formula_feeder_simplified(molecular_formula,
IonPathways, peak_spacing, intensity_cutoff)
```

IUPAC_Isotopes *IUPAC Isotopes*

Description

This data consists of element, mass, abundance and valence of 289 stable isotopes for 84 elements.

Usage

```
data("IUPAC_Isotopes")
```

Format

A data frame with 289 observations on the following 4 variables.

`element` a character vector

`mass` a character vector

`abundance` a character vector

`valence` a character vector

Note

The PubChem source for isotopes abunadnce and mass data is IUPAC.

References

Kim S, Chen J, Cheng T, Gindulyte A, He J, He S, Li Q, Shoemaker BA, Thiessen PA, Yu B, Zaslavsky L, Zhang J, Bolton EE. PubChem in 2021: new data content and improved web interfaces. Nucleic Acids Res. 2021 Jan 8; 49(D1):D1388–D1395. doi:10.1093/nar/gkaa971.

Examples

```
data(IUPAC_Isotopes)
```

`SUFA_hill_molecular_formula_printer`
Print Hill Molecular Formula

Description

This function produces molecular formulas from a list numerical vectors in the Hill notation system

Usage

```
SUFA_hill_molecular_formula_printer(Elements, MolVecMat)
```

Arguments

`Elements`	A vector string of the used elements.
`MolVecMat`	A matrix of numerical vectors of molecular formulas in each row.

Value

A vector of molecular formulas

Examples

```
Elements <- c("C", "H", "O", "N", "Br", "Cl")
MoleFormVec1 <- c(2, 6, 1, 0, 0, 0) # C2H6O
MoleFormVec2 <- c(8, 10, 2, 4, 0 ,0) # C8H10N4O2
MoleFormVec3 <- c(12, 2, 1, 0, 5, 3) # C12H2Br5Cl3O
MolVecMat <- rbind(MoleFormVec1, MoleFormVec2, MoleFormVec3)
H_MolF <- SUFA_hill_molecular_formula_printer(Elements, MolVecMat)
```

UFA_locate_regex *UFA Locate regex*

Description

Locate indices of the pattern in the string

Usage

```
UFA_locate_regex(string, pattern)
```

Arguments

string	a string as character
pattern	a pattern to screen

Details

This function returns 'NA' when no matches is detected for the pattern.

Value

A 2-column matrix of location indices. The first and second columns represent start postions and end positions, respectively.

Examples

```
pattern <- "Cl"
string <- "NaCl.5HCl"
Location_Cl <- UFA_locate_regex(string, pattern)
```

Index