Package ‘IDSL.SUFA’

March 23, 2023

Type Package

Title Simplified UFA

Version 1.3

Depends R (>= 3.5)

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

Maintainer Dinesh Barupal <dinesh.barupal@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 and educational mass spectrometry courses. The 'IDSL.SUFA' package also provides an ancillary module to process user-defined adduct formulas.

License MIT + file LICENSE

URL https://github.com/idslme/idsl.sufa

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

Encoding UTF-8

Archs i386, x64

NeedsCompilation no

Repository CRAN

Date/Publication 2023-03-23 22:32:15 UTC

R topics documented:

element_sorter *Element Sorter*

Description

This module sorts 84 non-labeled and 14 labeled elements in the periodic table for molecular formula deconvolution and isotopic profile calculation.

Usage

```
element_sorter(ElementList = "all", alphabeticalOrder = TRUE)
```

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.
`alphabeticalOrder`	‘TRUE‘ should be used to sort the elements for elemental deconvolution (default value), ‘FALSE‘ should be used to keep the input order.

Value

`Elements`	A string vector of elements (alphabetically sorted or unsorted)
`massAbundanceList`	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

This function takes a formula and a vector of ionization pathways and returns the adduct formulas.

Usage

```
formula_adduct_calculator(molecular_formula, IonPathway)
```

Arguments

`molecular_formula`	molecular formula
`IonPathway`	An ionization pathway. 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: 'IonPathway <- c("[M]+", "[M+H]+", "[2M-Cl]-", "[3M+CO2-H2O+Na-KO2+HCl-NH4]-")'

Value

A vector of adduct formulas

Examples

```
molecular_formula = "C15H10O7"
IonPathway = c("[M]+","[M+H]","[M+H2O+H]","[M+Na]")
Formula_adducts <- formula_adduct_calculator(molecular_formula, IonPathway)
```

---

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, LElements = length(Elements),
allowedRedundantElements = FALSE)
```

Arguments

`molecular_formula`	molecular formula
`Elements`	a string vector of elements. This value must be driven from the 'element_sorter' function.
`LElements`	number of elements. To speed up loop calculations, consider calculating the number of elements outside of the loop.
`allowedRedundantElements`	'TRUE' should be used to deconvolute molecular formulas with redundant elements (e.g. CO2CH3O), and 'FALSE' should be used to skip such complex molecular formulas.(default value)

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 <- "[13]C2C12H2Br5Cl3O"
Elements_molecular_formula <- c("[13]C", "C", "H", "O", "Br", "Cl")
EL <- element_sorter(ElementList = Elements_molecular_formula, alphabeticalOrder = TRUE)
Elements <- EL[["Elements"]]
LElements <- length(Elements)
##
mol_vec <- formula_vector_generator(molecular_formula, Elements, LElements)
##
regenerated_molecular_formula <- SUFA_hill_molecular_formula_printer(Elements, mol_vec)
```

`ionization_pathway_deconvoluter` *Ionization Pathway Deconvoluter*

Description

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

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()[["Elements"]]
IonPathways <- c("[M]+", "[M+H]+", "[2M-Cl]-", "[3M+CO2-H2O+2Na-KO2+HCl-2NH4]-")
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. Nonetheless, this function may suit more complicated tasks with complex biological compounds. 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, massAbundanceList, peak_spacing,
intensity_cutoff, UFA_IP_memeory_variables = c(1e30, 1e-12, 100))
```

Arguments

`MoleFormVec`	A numerical vector of the molecular formula
`massAbundanceList`	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 intensity threshold for isotopic profiles in percentage
`UFA_IP_memeory_variables`	A vector of three variables. Default values are c(1e30, 1e-12, 100) to manage memory usage. UFA_IP_memeory_variables[1] is used to control the overall size of isotopic combinations. UFA_IP_memeory_variables[2] indicates the minimum relative abundance (RA calculated by eq(1) in the reference [1]) of an isotopologue to include in the isotopic profile calculations. UFA_IP_memeory_variables[3] is the maximum elapsed time to calculate the isotopic profile on the ‘setTimeLimit‘ function of base R.

Value

A matrix of isotopic profile. The first and second column represents the mass and intensity 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(alphabeticalOrder = TRUE)
Elements <- EL[["Elements"]]
massAbundanceList <- EL[["massAbundanceList"]]
peak_spacing <- 0.005 # mDa
intensity_cutoff <- 1 # (in percentage)
MoleFormVec <- formula_vector_generator("C8H10N4O2", Elements)
IP <- isotopic_profile_calculator(MoleFormVec, massAbundanceList, peak_spacing,
intensity_cutoff)
```

`isotopic_profile_molecular_formula_feeder`
Isotopic Profile Molecular Formula Feeder

Description

A function to calculate isotopic profiles from a molecular formulas

Usage

```
isotopic_profile_molecular_formula_feeder(molecular_formula, peak_spacing = 0,
intensity_cutoff = 1, IonPathway = "[M]", UFA_IP_memeory_variables = c(1e30, 1e-12, 100),
plotProfile = TRUE, allowedVerbose = TRUE)
```

Arguments

`molecular_formula`	A molecular formulas
`peak_spacing`	A maximum space between isotopologues in Da to merge neighboring isotopologues.
`intensity_cutoff`	A minimum intensity threshold for isotopic profiles in percentage.
`IonPathway`	An ionization pathway. 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: 'IonPathway <- c("[M]+", "[M+H]+", "[2M-Cl]-", "[3M+CO2-H2O+Na-KO2+HCl-NH4]-")'
`UFA_IP_memeory_variables`	A vector of three variables. Default values are c(1e30, 1e-12, 100) to manage memory usage. UFA_IP_memeory_variables[1] is used to control the overall size of isotopic combinations. UFA_IP_memeory_variables[2] indicates the minimum relative abundance (RA calculated by eq(1) in the reference [1]) of an isotopologue to include in the isotopic profile calculations. UFA_IP_memeory_variables[3] is the maximum elapsed time to calculate the isotopic profile on the ‘setTimeLimit‘ function of base R.

`plotProfile`	c(TRUE, FALSE). A 'TRUE' plotProfile generates a spectra plot.
`allowedVerbose`	c(TRUE, FALSE). A 'TRUE' allowedVerbose provides messages about the flow of the function.

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 isotopologues
intensity_cutoff <- 1
IonPathway <- "[M+H]+"
isotopic_profile <- isotopic_profile_molecular_formula_feeder(molecular_formula,
peak_spacing, intensity_cutoff, IonPathway)
```

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, ignore.case = FALSE, perl = FALSE, fixed = FALSE,
useBytes = FALSE)
```

Arguments

string	a string as character
pattern	a pattern to screen
ignore.case	ignore.case
perl	perl
fixed	fixed
useBytes	useBytes

Details

This function returns ‘NULL‘ when no matches are detected for the pattern.

Value

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

Examples

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

Index