# Package ‘IDSL.SUFA’

March 21, 2022

**Type**  Package  
**Title**  Simplified UFA  
**Version**  1.0  
**Depends**  R (>= 3.5)  
**Author**  Sajjad Fakouri-Baygi [cre, aut] (<https://orcid.org/0000-0002-6864-6911>),  
Dinesh Barupal [aut] (<https://orcid.org/0000-0002-9954-8628>)  
**Maintainer**  Sajjad Fakouri-Baygi <sajjad.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                     2
  formula_adduct_calculator          2
  formula_vector_generator           3
  ionization_pathway_deconvoluter    4
  isotopic_profile_calculator        5
  isotopic_profile_molecular_formula_feeder_simplified 6
  IUPAC_Isotopes                     7
  SUFA_hill_molecular_formula_printer 8
  UFA_locate_regex                   9
```
element_sorter  

**Description**

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

**Usage**

```r
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**

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

---

formula_adduct_calculator

**Description**

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

**Usage**

```r
formula_adduct_calculator(molecular_formula, IonPathways)
```
**formula_vector_generator**

**Description**

This function converts a molecular formula into a numerical vector.

**Usage**

```r
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

```r
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**

**Description**

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

**Usage**

```r
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**

```r
Elements <- element_sorter()[[1]]
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**

```r
isotopic_profile_calculator(MoleFormVec, Elements_mass_abundance, peak_spacing, intensity_cutoff, UFA_IP_memory_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 intensity threshold for isotopic profiles in percentage

- **UFA_IP_memory_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, memory_variables[1] is used to adjust memory usage. memory_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.

**Value**

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

**References**


isotopic_profile_molecular_formula_feeder_simplified

Isotopic Profile Molecular Formula Feeder

Description

A function to calculate isotopic profiles from a molecular formulas

Usage

```r
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 isotopologues.
- `intensity_cutoff`: A minimum intensity 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, memory_variables[1] is used to adjust memory usage. memory_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.

See Also

https://ipc.idsl.me/

Examples

```r
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)
```
Value

A list of isotopic profiles

References


See Also

https://ipc.idsl.me/

Examples

```r
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
IonPathways <- "[M+H]+"
isotopic_profile <- isotopic_profile_molecular_formula_feeder_simplified(molecular_formula, IonPathways, peak_spacing, intensity_cutoff)
```

---

### IUPAC_Isotopes

<table>
<thead>
<tr>
<th>IUPAC_Isotopes</th>
<th>IUPAC Isotopes</th>
</tr>
</thead>
</table>

**Description**

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

**Usage**

```r
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 abundance and mass data is IUPAC.
References


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)
**Description**

Locate indices of the pattern in the string

**Usage**

\[
\text{UFA\_locate\_regex}(\text{string}, \text{pattern})
\]

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>string</td>
<td>a string as character</td>
</tr>
<tr>
<td>pattern</td>
<td>a pattern to screen</td>
</tr>
</tbody>
</table>

**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 positions and end positions, respectively.

**Examples**

```r
pattern <- "Cl"
string <- "NaCl.5HCl"
Location_Cl <- UFA\_locate\_regex(string, pattern)
```
Index

* datasets
  IUPAC_Isotopes, 7

element_sorter, 2
  formula_adduct_calculator, 2
  formula_vector_generator, 3
  ionization_pathway_deconvoluter, 4
  isotopic_profile_calculator, 5
  isotopic_profile_molecular_formula_feeder_simplified, 6
  IUPAC_Isotopes, 7

SUFA_hill_molecular_formula_printer, 8

UFA_locate_regex, 9