Package ‘IDSL.NPA’

January 23, 2023

Type Package

Title Nominal Peak Analysis (NPA)

Version 1.0

Depends R (>= 4.0)

Imports IDSL.MXP (>= 1.8), IDSL.IPA (>= 2.6), IDSL.FSA, doParallel, foreach, readxl

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 pipeline for processing nominal mass spectrometry data to create .msp files for untargeted nominal mass data processing.

License MIT + file LICENSE

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

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

Encoding UTF-8

Archs i386, x64

NeedsCompilation no

Repository CRAN

Date/Publication 2023-01-23 09:30:06 UTC

R topics documented:

IDSL.NPA_MSPgenerator 2
IDSL.NPA_referenceMSPgenerator 2
IDSL.NPA_workflow 3
IDSL.NPA_xlsxAnalyzer 3
NPA_fragmentationPeakDetection 4
NPA_peakDeconvolution 5
NPA_reference_xlsxAnalyzer 5
NPA_workflow 6
NPA_xlsxAnalyzer 7

Index **8**

`IDSL.NPA_MSPgenerator` *IDSL.NPA MSP Generator*

Description

This function creates standard .msp files that can also be used for Pepsearch.

Usage

```
IDSL.NPA_MSPgenerator(NPA_peaklist, number_processing_threads = 1)
```

Arguments

`NPA_peaklist`	A dataframe peaklist of co-detected peaks
`number_processing_threads`	Number of processing threads for multi-threaded processing

Value

A string of standard .msp file

`IDSL.NPA_referenceMSPgenerator` *IDSL.NPA Reference MSP Generator*

Description

This function creates reference standard .msp files.

Usage

```
IDSL.NPA_referenceMSPgenerator(NPA_peaklist, refNPAtable, selectedPeaks_IDref)
```

Arguments

`NPA_peaklist`	A dataframe peaklist of co-detected peaks
`refNPAtable`	reference NPA table
`selectedPeaks_IDref`	selectedPeaks_IDref

Value

A string of standard .msp file

`IDSL.NPA_workflow` *IDSL.NPA workflow*

Description

This function executes the NPA workflow.

Usage

```
IDSL.NPA_workflow(spreadsheet)
```

Arguments

`spreadsheet`	NPA spreadsheet

Value

This function organizes the IDSL.NPA file processing for better performance using the template spreadsheet.

`IDSL.NPA_xlsxAnalyzer` *IDSL.NPA workflow xlsx Analyzer*

Description

This function processes the spreadsheet of the NPA parameters to ensure the parameter inputs are consistent with the requirements of the IDSL.NPA pipeline.

Usage

```
IDSL.NPA_xlsxAnalyzer(spreadsheet)
```

Arguments

`spreadsheet`	'Start' tab of the parameter spreadsheet

Value

This function returns the NPA parameters to feed the 'IDSL.NPA_workflow' function.

`NPA_fragmentationPeakDetection`

NPA Fragmentation Peaks Detection

Description

This function detects fragmentation peaks for the NPA analysis

Usage

```
NPA_fragmentationPeakDetection(input_MS_path, MSfilename, smoothingWindow,
peakHeightThreshold, minSNRbaseline, RTtolerance, nSpline, topRatioPeakHeight,
minIonRangeDifference, minNumNPApeaks, pearsonRHOthreshold, outputNPAeic = NULL,
number_processing_threads = 1)
```

Arguments

`input_MS_path`	path to the MS files
`MSfilename`	MS file
`smoothingWindow`	number of scans for peak smoothing.
`peakHeightThreshold`	A minimum peak height threshold
`minSNRbaseline`	A minimum baseline S/N threshold
`RTtolerance`	retention time tolerance to detect common peaks
`nSpline`	number of points for further smoothing using a cubic spline smoothing method to add more points to calculate Pearson correlation rho values
`topRatioPeakHeight`	The top percentage of the chromatographic peak to calculate Pearson correlation rho values
`minIonRangeDifference`	Minimum distance (Da) between lowest and highest m/z to prevent clustering isotopic envelopes
`minNumNPApeaks`	Minumum number of ions in a NPA cluster
`pearsonRHOthreshold`	Minimum threshold for Pearson correlation rho values
`outputNPAeic`	When ‘NULL‘ NPA EICs are not plotted. ‘outputNPAeic‘ represents an address to save NPA EICs figures.
`number_processing_threads`	Number of processing threads for multi-threaded processing

Value

A dataframe peaklist of co-detected NPA analysis.

`NPA_peakDeconvolution` *NPA Peak Deconvolution*

Description

This function detects fragmentation peaks for the NPA analysis.

Usage

```
NPA_peakDeconvolution(input_MS_path, MSfilename, smoothingWindow,
peakHeightThreshold, minSNRbaseline, number_processing_threads = 1)
```

Arguments

`input_MS_path`	path to the MS files
`MSfilename`	MS file
`smoothingWindow`	number of scans for peak smoothing.
`peakHeightThreshold`	A minimum peak height threshold
`minSNRbaseline`	A minimum baseline S/N threshold
`number_processing_threads`	Number of processing threads for multi-threaded processing

Value

A dataframe peaklist of co-detected DIA analysis.

`NPA_reference_xlsxAnalyzer` *NPA reference xlsxAnalyzer*

Description

NPA reference xlsxAnalyzer

Usage

```
NPA_reference_xlsxAnalyzer(ref_xlsx_file, input_path_hrms = NULL, PARAM = NULL,
PARAM_ID = "", checkpoint_parameter = TRUE)
```

Arguments

`ref_xlsx_file`	ref_xlsx_file
`input_path_hrms`	input_path_hrms
`PARAM`	PARAM
`PARAM_ID`	PARAM_ID
`checkpoint_parameter`	checkpoint_parameter

Value

`ref_table`	ref_table
`PARAM`	PARAM
`checkpoint_parameter`	checkpoint_parameter

`NPA_workflow` *FSA NPA Workflow*

Description

This function runs the NPA analysis.

Usage

```
NPA_workflow(PARAM_NPA)
```

Arguments

`PARAM_NPA`	PARAM_NPA

Value

This function stores .Rdata and .csv data from dataframe peaklist of co-detected NPA analysis.

Examples

```
s_path <- system.file("extdata", package = "IDSL.NPA")
SSh1 <- paste0(s_path,"/NPA_parameters.xlsx")
## To see the results, use a known folder instead of the `tempdir()` command
temp_wd <- tempdir()
temp_wd_zip <- paste0(temp_wd, "/idsl.npa_testfile.zip")
spreadsheet <- readxl::read_xlsx(SSh1, sheet = "NPA")
PARAM_NPA <- cbind(spreadsheet[, 2], spreadsheet[, 4])
download.file(paste0("https://github.com/idslme/IDSL.NPA/blob/main/",
                     "NPA_educational_files/idsl.npa_testfile.zip?raw=true"),
```

```
              destfile = temp_wd_zip, mode = "wb")
unzip(temp_wd_zip, exdir = temp_wd)
PARAM_NPA[2, 2] <- "NO"
PARAM_NPA[4, 2] <- temp_wd
PARAM_NPA[8, 2] <- temp_wd
## To ensure `PARAM_NPA` is consistent with the `NPA_workflow`
PARAM_NPA <- NPA_xlsxAnalyzer(PARAM_NPA)
##
NPA_workflow(PARAM_NPA)
```

`NPA_xlsxAnalyzer` *xlsx Analyzer for NPA analysis*

Description

This function processes the spreadsheet of the NPA spreadsheet tab to ensure the parameter inputs are in agreement with requirements of the NPA analysis.

Usage

```
NPA_xlsxAnalyzer(spreadsheet)
```

Arguments

`spreadsheet`	NPA spreadsheet tab

Value

parameters to feed the ‘NPA_workflow‘ function.

Index

IDSL.NPA_MSPgenerator, 2
IDSL.NPA_referenceMSPgenerator, 2
IDSL.NPA_workflow, 3
IDSL.NPA_xlsxAnalyzer, 3

NPA_fragmentationPeakDetection, 4
NPA_peakDeconvolution, 5
NPA_reference_xlsxAnalyzer, 5
NPA_workflow, 6
NPA_xlsxAnalyzer, 7