

# Package ‘IDSL.NPA’

July 21, 2025

**Type** Package

**Title** Nominal Peak Analysis (NPA)

**Version** 1.2

**Depends** R (>= 4.0)

**Imports** IDSL.MXP, IDSL.IPA, IDSL.FSA, readxl

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

**Maintainer** Dinesh Barupal <dinesh.barupal@mssm.edu>

**Description**

A pipeline to process nominal mass spectrometry data to create .msp files for untargeted analyses.

**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** 2023-06-29 14:00:05 UTC

## Contents

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, pearsonRH0threshold, 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	Minimum number of ions in a NPA cluster
pearsonRH0threshold	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	<i>FSA NPA Workflow</i>
--------------	-------------------------

---

**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](#)