

# Package ‘IDSL.IPA’

August 8, 2022

**Type** Package

**Title** Intrinsic Peak Analysis (IPA) for HRMS Data

**Version** 2.1

**Depends** R (>= 4.0)

**Imports** IDSL.MXP (>= 1.4), xml2, RNetCDF, base64enc, grid, readxl, parallel, doParallel, foreach, ggplot2, gridExtra, png

**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 sophisticated pipeline for processing LC/HRMS data to extract signals of organic compounds. The package performs isotope pairing, peak detection, alignment, RT correction, gap filling, peak annotation and visualization of extracted ion chromatograms and total ion chromatograms.

**License** MIT + file LICENSE

**URL** <https://ipa.idsl.me>, <https://github.com/idslme/idsl.ipa>

**BugReports** <https://github.com/idslme/idsl.ipa/issues>

**Encoding** UTF-8

**LazyData** true

**Archs** i386, x64

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2022-08-08 11:50:02 UTC

## R topics documented:

asymmetry_factor . . . . .	3
baseline_developer . . . . .	3
chromatogram_builder . . . . .	4
chromatography_analysis . . . . .	5

derivative_skewness . . . . .	6
der_5points_stencil . . . . .	7
EIC_plotter . . . . .	7
fronting_tailing_resolver . . . . .	8
gaussianity_measurement . . . . .	9
IPA_CompoundsAnnotation . . . . .	10
IPA_GapFiller . . . . .	10
IPA_IsotopePairing . . . . .	11
IPA_MSdeconvoluter . . . . .	11
IPA_PeakAlignment . . . . .	12
IPA_PeakAnalyzer . . . . .	12
IPA_PeaklistAnnotation . . . . .	13
IPA_TargetedAnalysis . . . . .	13
IPA_Workflow . . . . .	14
IPA_xlsxAnalyzer . . . . .	15
islocalminimum . . . . .	16
islocaloptimum . . . . .	16
loadRdata . . . . .	17
mzRTindexer . . . . .	17
mz_clustering_xic . . . . .	18
opendir . . . . .	18
peak_alignment . . . . .	19
peak_area . . . . .	19
peak_detection . . . . .	20
peak_sharpness . . . . .	21
peak_spline . . . . .	21
peak_width . . . . .	22
peak_Xcol2 . . . . .	22
plot_mz_eic . . . . .	23
plot_simple_tic . . . . .	24
primary_peak_analyzer . . . . .	24
pseudomoments_symmetry . . . . .	26
recursive_mass_correction . . . . .	26
reference_peaks_detector . . . . .	28
sample_rt_corrector . . . . .	28
segment . . . . .	29
snr_rms . . . . .	30
snr_signal2baseline . . . . .	30
snr_xcms . . . . .	31
spectralList_filtering . . . . .	32
usp_tailing_factor . . . . .	32
XIC . . . . .	33
xlsxAnalyzer_EIC . . . . .	33

---

asymmetry_factor	<i>Asymmetry factor for a chromatographic peak</i>
------------------	--

---

**Description**

This function calculates an asymmetry factor for a chromatographic peak.

**Usage**

```
asymmetry_factor(rt, int)
```

**Arguments**

rt	a vector of retention times for the chromatographic peak.
int	a vector of intensities corresponding to the vector of retention times for the chromatographic peak.

**Value**

asymmetry of the chromatographic peak. 1 is for very symmetric peak.

**Examples**

```
data(peak_spline)
rt <- peak_spline[, 1]
int <- peak_spline[, 2] - peak_spline[, 3]
asymmetry_factor(rt, int)
```

---

baseline_developer	<i>Develop a baseline for the chromatogram using local minima</i>
--------------------	---

---

**Description**

This function generates a vector of baselines for the chromatogram using local minima. It also is capable of excluding outlier local minima to generate a realistic baseline including true baseline regions. This baseline may represent the local noise levels for the chromatogram.

**Usage**

```
baseline_developer(segment, int)
```

**Arguments**

segment	a matrix or a vector of adjusted scan number of local minima w/ or w/o redundant local minima. Adjusted scan numbers are the scan numbers but adjusted to start at 1.
int	a vector of intensities of the chromatogram.

**Value**

A vector of baselines in the same size of the "int" vector.

**Examples**

```
data(segment)
data(chromatogram_builder)
int <- chromatogram_builder$SmoothedChromatogram
baseline_developer(segment, int)
```

---

chromatogram\_builder *chromatogram builder for m/z = 263.1678 in 003.d from cord blood sample*

---

**Description**

illustrates a chromatogram and baseline vectors to indicate chromatogram development.

**Usage**

```
data("chromatogram_builder")
```

**Format**

A data frame with 219 observations on the following 6 variables.

ScanNumber a numeric vector

RetentionTime a numeric vector

SmoothedChromatogram a numeric vector

RawChromatogram a numeric vector

'<sup>12</sup>C/<sup>13</sup>C Isotopologue Pairs' a numeric vector

Baseline a numeric vector

**Examples**

```
data(chromatogram_builder)
```

---

`chromatography_analysis`*Chromatography analysis*

---

**Description**

This function detect individual chromatographic peaks and measures their peak qualification metrics.

**Usage**

```
chromatography_analysis(spec_scan_xic, smoothing_window,  
peak_resolving_power, min_nIsoPair, min_peak_height,  
min_ratio_IsoPair, max_rpw, min_snr_baseline,  
max_R13C_integrated_peak, max_percentage_missing_scans,  
mz_target, rt_target = 0, mass_accuracy_xic, spectralList,  
RetentionTime, n_spline)
```

**Arguments**

<code>spec_scan_xic</code>	a matrix consists of 5 columns. The column contents are the m/z of 12C isotopologues, intensity of 12C isotopologues, scan number (t), m/z of 13C isotopologues, and intensity of 13C isotopologues, respectively.
<code>smoothing_window</code>	number of scans for peak smoothing
<code>peak_resolving_power</code>	a value to represent peak resolving power
<code>min_nIsoPair</code>	minimum number of nIsoPair for an individual peak
<code>min_peak_height</code>	minimum peak height for an individual peak
<code>min_ratio_IsoPair</code>	minimum ratio of nIsoPair per number of available scans within an individual peak
<code>max_rpw</code>	maximum allowed value of ratio of peak width at half-height to baseline (RPW) for an individual peak
<code>min_snr_baseline</code>	minimum S/N baseline for an individual peak
<code>max_R13C_integrated_peak</code>	maximum allowed value of average R13C for an individual peak
<code>max_percentage_missing_scans</code>	maximum allowed value of percentage missing scans on the raw chromatogram for an individual peak.
<code>mz_target</code>	m/z value to perform chromatography analysis

rt_target	retention time value for a targeted peak to calculate the ancillary chromatography parameters. When this parameter set at 0, the ancillary chromatography parameters are calculated for the entire detected peaks.
mass_accuracy_xic	mass error to perform chromatography analysis
spectralList	a list of mass spectra in each chromatogram scan
RetentionTime	a vector of retention times vs. corresponding scan numbers
n_spline	number of points for further smoothing using a cubic spline smoothing method to calculate ancillary chromatographical parameters

**Value**

a data frame consisting of 24 columns representing chromatography and mass spectrometry parameters. Each row represents an individual separated chromatographic peak.

---

derivative\_skewness    *Derivative skewness*

---

**Description**

This function calculates skewness of a chromatographic peak using first order degree of numerical differentiation.

**Usage**

```
derivative_skewness(rt, int)
```

**Arguments**

rt                    a vector representing retention times of the chromatographic peak.  
int                    a vector representing intensities of the chromatographic peak.

**Value**

Skewness of a chromatographic peak. 1 is for very symmetric peak. Minimum is 0 from this function.

**Examples**

```
data(peak_spline)
rt <- peak_spline[, 1]
int <- peak_spline[, 2]
derivative_skewness(rt, int)
```

---

der\_5points\_stencil     *Numerical differentiation by five-point stencil method*

---

**Description**

This module performs numerical differentiation using the five-point stencil method.

**Usage**

```
der_5points_stencil(x, y, n)
```

**Arguments**

x	a vector of values for x.
y	a vector of values for y.
n	order of numerical differentiation (n=1-4).

**Value**

A matrix of 2 columns. The first column represents x and the second column represents numerical differentiation values. This matrix has four rows (two rows from the beginning and 2 rows from the end) less than length of x or y.

**Examples**

```
data(peak_spline)
rt <- peak_spline[, 1]
int <- peak_spline[, 2]
n <- 2 # second order derivative
der_5points_stencil(rt, int, n)
```

---

EIC\_plotter

*EIC plotter*

---

**Description**

This function plots the EIC figure and annex the chromatographic properties to the EIC figures.

**Usage**

```
EIC_plotter(spec_scan_xic, peak_property_xic, smoothing_window,
peak_resolving_power, mass_accuracy_xic, spectralList, RetentionTime,
mz_target, rt_target, file_name, legend_EIC)
```

**Arguments**

spec_scan_xic	a matrix consists of 5 columns. The column contents are the m/z of 12C isotopologues, intensity of 12C isotopologues, scan number (t), m/z of 13C isotopologues, and intensity of 13C isotopologues.
peak_property_xic	a data frame representing chromatographic peak properties.
smoothing_window	number of scans for peak smoothing.
peak_resolving_power	a value to represent peak resolving power.
mass_accuracy_xic	a mass accuracy value to perform chromatography analysis.
spectralList	a list of mass spectra in each chromatogram scan.
RetentionTime	a vector of retention times vs. corresponding scan numbers.
mz_target	an m/z value to perform chromatography analysis.
rt_target	the retention time value of the candidate peak.
file_name	name of HRMS file used for peak construction.
legend_EIC	A file to attach the legends on the EIC figures.

**Value**

A figure to show the EIC and its property table.

---

fronting\_tailing\_resolver

*Fronting and tailing peaks resolver*

---

**Description**

This function attempts to resolve peak tailings or frontings into the main peak in case they were detected as separate peaks.

**Usage**

```
fronting_tailing_resolver(segment, int, max_space, peak_resolving_power)
```

**Arguments**

segment	a matrix or a vector of peak boundaries.
int	a vector of intensities of the entire chromatogram.
max_space	maximum scan number difference between peak tailing or fronting and the main peak.
peak_resolving_power	power of peak resolving tool.



**Value**

A matrix of 2 columns. Each row indicates peak boundary indices on the 'int' vector after resolving fronting and tailing peaks.

**Examples**

```
data(segment)
data(chromatogram_builder)
int <- chromatogram_builder$SmoothedChromatogram
max_space <- 7
peak_resolving_power <- 0.2
fronting_tailing_resolver(segment, int, max_space, peak_resolving_power)
```

---

gaussianity\_measurement

*gaussianity measurement*

---

**Description**

This module measures gaussianity of chromatographic peak using Pearson correlation coefficients ( $\rho$ ) at top 80 percent of peak.

**Usage**

```
gaussianity_measurement(RT, Int, BL, gauge = 0.8)
```

**Arguments**

RT	a vector of retention times of the chromatographic peak.
Int	a vector of intensities of the chromatographic peak.
BL	a vector of baseline of the chromatographic peak.
gauge	represents the gauge height of peak for Gaussianity measurement.

**Value**

Gaussianity of the chromatographic peak.

**Examples**

```
data("peak_spline")
RT <- peak_spline[, 1]
Int <- peak_spline[, 2]
BL <- peak_spline[, 3]
gaussianity_measurement(RT, Int, BL, gauge = 0.8)
```

---

IPA\_CompoundsAnnotation

*Compound-centric peak annotation*

---

### **Description**

This function performs compound-centric peak annotation.

### **Usage**

IPA\_CompoundsAnnotation(PARAM)

### **Arguments**

PARAM                    a data frame from IPA\_xlsxAnalyzer function containing the IPA parameters.

### **Value**

This function saves individual .CSV files for each compound in the "compound\_centeric\_annotation" folder.

---

IPA\_GapFiller

*IPA GapFiller*

---

### **Description**

This function fills the gaps on the peak table.

### **Usage**

IPA\_GapFiller(PARAM)

### **Arguments**

PARAM                    a data frame from IPA\_xlsxAnalyzer function containing the IPA parameters.

### **Value**

This function saves individual .CSV and .Rdata files for the gap-filled peak tables for peak height, area, and R13C properties in the "peak\_alignment" folder.

---

IPA\_IsotopePairing      *IPA Isotope Pairing*

---

### Description

This function pairs isotopologue in high-resolution mass spectral datasets

### Usage

```
IPA_IsotopePairing(spectralList, int_threshold, mass_accuracy_isotope_pair,
massDifferenceIsotopes)
```

### Arguments

spectralList      list of mass spectra in each chromatogram scan  
int\_threshold    intensity threshold at each chromatogram scan  
mass\_accuracy\_isotope\_pair  
                  mass error to detect pair isotopologues  
massDifferenceIsotopes  
                  mass difference to pair isotopologues. (Default =  $\Delta C = {}^{13}\text{C} - {}^{12}\text{C} = 1.003354835336$ ,  $\Delta S = {}^{34}\text{S} - {}^{32}\text{S} = 1.9957958356$ , or any numerical value.

### Value

A matrix consists of 5 columns. The column contents are the m/z of  ${}^{12}\text{C}$  isotopologues, intensity of  ${}^{12}\text{C}$  isotopologues, scan number (t), m/z of  ${}^{13}\text{C}$  isotopologues, and intensity of  ${}^{13}\text{C}$  isotopologues, respectively.

---

IPA\_MSdeconvoluter      *MS deconvoluter*

---

### Description

This function deconvolute mass spectrometry files into a list of mass spectrals and a vector of retention times.

### Usage

```
IPA_MSdeconvoluter(HRMS_path, MSfile, MS_level = 1)
```

### Arguments

HRMS\_path      address of the mass spectrometry file  
MSfile          mass spectrometry file.  
MS\_level        MS level to extract information.

**Value**

spectralList      a list of mass spectra.  
 RetentionTime    a vector of retention times for scan numbers.  
 MS\_polarity      mass spectrometry ionization mode (+/-)

---

IPA\_PeakAlignment      *IPA peak alignment*

---

**Description**

This function produce an aligned peak table from individual peaklists.

**Usage**

IPA\_PeakAlignment(PARAM)

**Arguments**

PARAM              is a data frame from IPA\_xlsxAnalyzer function.

**Value**

This function saves individual .CSV and .Rdata files for the aligned peak tables for peak height, area, and R13C properties in the "peak\_alignment" folder.

---

IPA\_PeakAnalyzer      *IPA Peak Analyzer*

---

**Description**

This function performs the IPA peak detection module.

**Usage**

IPA\_PeakAnalyzer(PARAM)

**Arguments**

PARAM              is a data frame from IPA\_xlsxAnalyzer function.

**Value**

This function saves individual peaklist files in .CSV and .Rdata formats for HRMS files in the "peaklists" folder.

---

IPA\_PeaklistAnnotation

*IPA Peaklist Annotation*

---

### Description

This function performs sample-centric peak annotation.

### Usage

```
IPA_PeaklistAnnotation(PARAM)
```

### Arguments

PARAM            a data frame from IPA\_xlsxAnalyzer function.

### Value

This function saves individual .CSV files for peak height, area, and R13C properties in the "sample\_centeric\_annotation" folder.

---

IPA\_TargetedAnalysis    *IPA Targeted Analysis*

---

### Description

This function plots extracted ion chromatogram (EIC) figures in the targeted mode.

### Usage

```
IPA_TargetedAnalysis(spreadsheet, mzCandidate, rtCandidate, exportEIC = TRUE,  
exportTable = FALSE)
```

### Arguments

spreadsheet      a spreadsheet containing the parameters.  
mzCandidate      a vector of candidate m/z values.  
rtCandidate      a vector of candidate RT values.  
exportEIC        TRUE by default. To plot and save EICs.  
exportTable      FALSE by default. To return the whole peaklists for the m/z and RT vectors,  
select TRUE.

### Value

This function saves extracted ion chromatograms in .png format in the "EICs" folder when "exportEIC = TRUE", and it save a table of peak properties when "exportTable = TRUE".

## Examples

```
s_path <- system.file("extdata", package = "IDSL.IPA")
SSh1 <- paste0(s_path, "/IPA_parameters.xlsx")
spreadsheet <- readxl::read_xlsx(SSh1, sheet = 'IPA_targeted')
temp_wd <- tempdir()
temp_wd_zip <- paste0(temp_wd, "/idsl_ipa_test_files.zip")
download.file(paste0("https://github.com/idslme/IDSL.IPA/blob/main/",
"IPA_educational_files/idsl_ipa_test_files.zip?raw=true"),
destfile = temp_wd_zip)
unzip(temp_wd_zip, exdir = temp_wd)
spreadsheet[2, 4] <- temp_wd
spreadsheet[5, 4] <- temp_wd
mzCandidate <- c(53.01853, 61.00759)
rtCandidate <- c(0.951, 0.961)
IDSL.IPA::IPA_TargetedAnalysis(spreadsheet, mzCandidate, rtCandidate)
```

---

IPA\_Workflow

*IPA Workflow*

---

## Description

This function executes the IPA workflow in order.

## Usage

```
IPA_Workflow(spreadsheet)
```

## Arguments

spreadsheet      IPA spreadsheet

## Value

This function organizes the IPA file processing for a better performance using the template spreadsheet.

## See Also

<https://ipa.idsl.me/home>

## Examples

```
library(IDSL.IPA)
s_path <- system.file("extdata", package = "IDSL.IPA")
SSh1 <- paste0(s_path, "/IPA_parameters.xlsx")
temp_wd <- tempdir()
```

```
temp_wd_zip <- paste0(temp_wd, "/idsl_ipa_test_files.zip")
spreadsheet <- readxl::read_xlsx(SSH1)
download.file(paste0("https://github.com/idslme/IDSL.IPA/blob/main/",
"IPA_educational_files/idsl_ipa_test_files.zip?raw=true"),
destfile = temp_wd_zip)
unzip(temp_wd_zip, exdir = temp_wd)
spreadsheet[7, 4] <- temp_wd
spreadsheet[40, 4] <- s_path
spreadsheet[10, 4] <- temp_wd
IPA_Workflow(spreadsheet)
```

---

IPA\_xlsxAnalyzer      *IPA xlsx Analyzer*

---

## Description

This function processes the spreadsheet of the IPA parameters to ensure the parameter inputs are in agreement with the IPA requirements.

## Usage

```
IPA_xlsxAnalyzer(spreadsheet)
```

## Arguments

spreadsheet      IPA spreadsheet

## Value

This function returns the IPA parameters to feed the IPA\_Workflow, IPA\_CompoundsAnnotation, IPA\_GapFiller, IPA\_PeakAlignment, IPA\_PeakAnalyzer, and IPA\_PeaklistAnnotation functions.

## Examples

```
s_path <- system.file("extdata", package = "IDSL.IPA")
SSH1 <- paste0(s_path, "/IPA_parameters.xlsx")
temp_wd <- tempdir()
temp_wd_zip <- paste0(temp_wd, "/idsl_ipa_test_files.zip")
spreadsheet <- readxl::read_xlsx(SSH1)
download.file(paste0("https://github.com/idslme/IDSL.IPA/blob/main/",
"IPA_educational_files/idsl_ipa_test_files.zip?raw=true"),
destfile = temp_wd_zip)
unzip(temp_wd_zip, exdir = temp_wd)
spreadsheet[7, 4] <- temp_wd
spreadsheet[40, 4] <- s_path # reference file location
spreadsheet[10, 4] <- temp_wd # output data location
PARAM <- IDSL.IPA::IPA_xlsxAnalyzer(spreadsheet)
```

---

<code>islocalminimum</code>	<i>islocalminimum</i>
-----------------------------	-----------------------

---

**Description**

This function returns indices of local minimum points on a curve.

**Usage**

```
islocalminimum(y)
```

**Arguments**

`y` is a vector of y values.

**Value**

A vector in the same size of vector 'y'. Local minimum arrays represented by -1.

**Examples**

```
data(chromatogram_builder)
int <- chromatogram_builder$SmoothedChromatogram
islocalminimum(int)
```

---

<code>islocaloptimum</code>	<i>islocaloptimum</i>
-----------------------------	-----------------------

---

**Description**

This function returns indices of local minimum and maximum points on a curve.

**Usage**

```
islocaloptimum(y)
```

**Arguments**

`y` is a vector of y values.

**Value**

A vector in the same size of vector 'y'. Local minimum and maximum arrays represented by -1 and +1, respectively.



**Examples**

```
data(chromatogram_builder)
int <- chromatogram_builder$SmoothedChromatogram
islocaloptimum(int)
```

---

loadRdata	<i>loadRdata</i>
-----------	------------------

---

**Description**

This function loads .Rdata files into a variable.

**Usage**

```
loadRdata(fileName)
```

**Arguments**

fileName is an .Rdata file.

**Value**

The called variable into the new assigned variable name.

---

mzRTindexer	<i>m/z - RT Indexer</i>
-------------	-------------------------

---

**Description**

This function locate the closeset pair of a reference (m/z - RT) pair in a 2-D grid of 'm/z' and 'RT' vectors.

**Usage**

```
mzRTindexer(MZvec, RTvec, MZref, RTref, MZtolerance, RTtolerance)
```

**Arguments**

MZvec	m/z vector
RTvec	RT vector
MZref	a reference m/z
RTref	a reference RT
MZtolerance	m/z tolerance
RTtolerance	RT tolerance

**Value**

index of closest pair to the reference (m/z - RT) pair

**Note**

This function return NULL in case no match is detected.

---

mz_clustering_xic	<i>mz clustering XIC</i>
-------------------	--------------------------

---

**Description**

This function clusters related 12C m/z values.

**Usage**

```
mz_clustering_xic(spec_scan, mass_accuracy_xic, min_peak_height, min_nIsoPair)
```

**Arguments**

spec_scan	a matrix consists of 3 columns. The column contents are the m/z of 12C isotopologues, intensity of 12C isotopologues, and scan number (t).
mass_accuracy_xic	mass accuracy to detect related 12C m/z values.
min_nIsoPair	minimum number of nIsoPair for an individual peak.
min_peak_height	minimum peak height for an individual peak.

**Value**

This function returns an list on index numbers of EICs for the "spec\_scan" variable.

---

opendir	<i>opendir</i>
---------	----------------

---

**Description**

This function opens the directory.

**Usage**

```
opendir(dir)
```

**Arguments**

dir	full address of the directory.
-----	--------------------------------

**Value**

This function opens its input directory for the user.

---

peak_alignment	<i>Peak alignment</i>
----------------	-----------------------

---

**Description**

This function aligns peaks from multiple peaklists and produce a peak table to find common peaks among multiple samples.

**Usage**

```
peak_alignment(input_path_pl, file_names_pl, RT_pl, mz_error, rt_tol,
n_quantile, number_processing_threads = 1)
```

**Arguments**

input_path_pl	path to directory of peaklists.
file_names_pl	name of peaklists for peak table production.
RT_pl	a list of corrected or uncorrected retention times for each peaklist.
mz_error	mass error to detect common peaks.
rt_tol	retention time tolerance to detect common peaks.
n_quantile	number of total m/z quantiles to split the whole table for faster processing.
number_processing_threads	number of processing threads

**Value**

This function returns an aligned peak table with index numbers from individual peaklists for each peak.

---

peak_area	<i>peak area</i>
-----------	------------------

---

**Description**

This function calculates area under the curve using a trapezoid method.

**Usage**

```
peak_area(x, y)
```

**Arguments**

x is a vector of x values.  
y is a vector of y values.

**Value**

A number for the integrated peak area.

**Examples**

```
data("peak_spline")
rt <- peak_spline[, 1]
int <- peak_spline[, 2]
peak_area(rt, int)
```

---

peak\_detection      *peak detection*

---

**Description**

This function detects separated chromatographical peaks on the chromatogram.

**Usage**

```
peak_detection(int)
```

**Arguments**

int a vector of intensities of the chromatogram.

**Value**

A matrix of 2 columns. Each row indicates peak boundary indices on the 'int' vector.

**Examples**

```
data(chromatogram_builder)
int <- chromatogram_builder$SmoothedChromatogram
peak_detection(int)
```

---

peak_sharpness	<i>Peak sharpness</i>
----------------	-----------------------

---

**Description**

This function measures sharpness of a chromatographic peak

**Usage**

```
peak_sharpness(int)
```

**Arguments**

int                    a vector of intensities of the chromatographic peak.

**Value**

A number representing peak sharpness. The higher values indicate higher sharpness.

**Examples**

```
data("peak_spline")
int <- peak_spline[, 2]
peak_sharpness(int)
```

---

peak_spline	<i>peak spline</i>
-------------	--------------------

---

**Description**

illustrates a smoothed peak using cubic spline smoothing method

**Usage**

```
data("peak_spline")
```

**Format**

A data frame with 100 observations on the following 3 variables.

rt\_spline a numeric vector  
int\_spline a numeric vector  
bl\_approx a numeric vector

**Examples**

```
data(peak_spline)
```

---

peak_width	<i>peak width measuement</i>
------------	------------------------------

---

**Description**

This function measures peak width at different peak heights.

**Usage**

```
peak_width(rt, int, gauge)
```

**Arguments**

rt	a vector of retention times of the chromatographic peak.
int	a vector of intensities of the chromatographic peak.
gauge	a height gauge to measure the peak width. This parameter should be between 0-1.

**Value**

A peak width at the guaged height.

**Examples**

```
data("peak_spline")
rt <- peak_spline[, 1]
int <- peak_spline[, 2] - peak_spline[, 3]
gauge <- 0.5
peak_width(rt, int, gauge)
```

---

peak_Xcol2	<i>Peak table producer</i>
------------	----------------------------

---

**Description**

This function fills the peak table from individual peaklists.

**Usage**

```
peak_Xcol2(input_path_peaklist, file_names_peaklist, peak_Xcol)
```

**Arguments**

input_path_peaklist	address of the peaklists.
file_names_peaklist	a vector of the peaklist file names.
peak_Xcol	a matrix of index numbers in individual peaklists for each peak (m/z-RT).

**Value**

peak_height	peak table for height values
peak_area	peak table for area values
peak_R13C	peak table for R13C values

---

plot_mz_eic	<i>plot_mz_eic</i>
-------------	--------------------

---

**Description**

plot\_mz\_eic

**Usage**

```
plot_mz_eic(filelist, filelocation, mztarget, mzdelta,
            numberOfcores, rtstart = 0, rtend = 0, plotTitle = "")
```

**Arguments**

filelist	filelist
filelocation	filelocation
mztarget	mztarget
mzdelta	mzdelta
numberOfcores	numberOfcores
rtstart	rtstart
rtend	rtend
plotTitle	plotTitle

**Value**

plot\_mz\_eic

plot\_simple\_tic      *plot\_simple\_tic*

---

### Description

plot\_simple\_tic

### Usage

```
plot_simple_tic(filelist, filelocation, numberOfcores,  
plotTitle = "Total Ion Chromatogram")
```

### Arguments

filelist	filelist
filelocation	filelocation
numberOfcores	numberOfcores
plotTitle	plotTitle

### Value

plot\_simple\_tic

---

primary\_peak\_analyzer    *Primary peak analyzer*

---

### Description

This function performs the first round of the chromatography analysis.

### Usage

```
primary_peak_analyzer(spec_scan, index_xic, scan_tol,  
spectralList, RetentionTime, mass_accuracy_xic,  
smoothing_window, peak_resolving_power, min_nIsoPair,  
min_peak_height, min_ratio_IsoPair, max_rpw, min_snr_baseline,  
max_R13C_integrated_peak, max_percentage_missing_scans,  
n_spline)
```



**Arguments**

spec_scan	a matrix consists of 5 columns. The column contents are the m/z of 12C isotopologues, intensity of 12C isotopologues, scan number (t), m/z of 13C isotopologues, and intensity of 13C isotopologues.
index_xic	a list of indices of candidate 12C m/z values from spec_scan matrix.
scan_tol	scan tolerance to extend the chromatogram for better calculations.
spectralList	a list of mass spectra in each chromatogram scan.
RetentionTime	a vector of retention times vs. corresponding scan numbers.
mass_accuracy_xic	a m/z value to perform chromatography analysis.
smoothing_window	number of scans for peak smoothing.
peak_resolving_power	a value to represent peak resolving power.
min_nIsoPair	minimum number of nIsoPair for an individual peak.
min_peak_height	minimum peak height for an individual peak.
min_ratio_IsoPair	minimum ratio of nIsoPair per number of available scans within an individual peak.
max_rpw	maximum allowed value of ratio of peak width at half-height to baseline (RPW) for an individual peak.
min_snr_baseline	minimum S/N baseline for an individual peak.
max_R13C_integrated_peak	maximum allowed value of average R13C for an individual peak.
max_percentage_missing_scans	maximum allowed value of percentage missing scans on the raw chromatogram for an individual peak.
n_spline	number of points for further smoothing using a cubic spline smoothing method.

**Value**

a data frame consisting of 24 columns representing chromatography and mass spectrometry parameters. Each row represents an individual separated chromatographic peak.

---

pseudomoments\_symmetry  
*pseudomoments symmetry*

---

### Description

This function measures peak symmetry and skewness using the inflection points of the peak on both sides.

### Usage

```
pseudomoments_symmetry(rt, int)
```

### Arguments

rt	a vector of retention times for the chromatographic peak.
int	a vector of intensities corresponding to the vector of retention times for the chromatographic peak.

### Value

PeakSymmetry	peak symmetry for the chromatographic peak.
Skewness	skewness for the chromatographic peak.

### Examples

```
data("peak_spline")
rt <- peak_spline[, 1]
int <- peak_spline[, 2] - peak_spline[, 3]
pseudomoments_symmetry(rt, int)
```

---

recursive\_mass\_correction  
*recursive mass correction*

---

### Description

This function performs recursive mass correction.

### Usage

```
recursive_mass_correction(peaklist, spec_scan, scan_tol,
spectralList, RetentionTime, mass_accuracy_xic, smoothing_window,
peak_resolving_power, min_nIsoPair, min_peak_height, min_ratio_IsoPair,
max_rpw, min_snr_baseline, max_R13C_integrated_peak,
max_percentage_missing_scans, n_spline)
```

**Arguments**

peaklist	an IPA peaklist from 'primary_peak_analyzer' function.
spec_scan	a matrix consists of 5 columns. The column contents are the m/z of 12C isotopologues, intensity of 12C isotopologues, scan number (t), m/z of 13C isotopologues, and intensity of 13C isotopologues.
scan_tol	a scan tolerance to extend the chromatogram for better calculations.
spectralList	a list of mass spectra in each chromatogram scan.
RetentionTime	a vector of retention times for corresponding scan numbers.
mass_accuracy_xic	an m/z value to perform chromatography analysis.
smoothing_window	a number of scans for peak smoothing.
peak_resolving_power	a value to represent peak resolving power.
min_nIsoPair	minimum number of nIsoPair for an individual peak.
min_peak_height	minimum peak height for an individual peak.
min_ratio_IsoPair	minimum ratio of nIsoPair per number of available scans within an individual peak.
max_rpw	maximum allowed value of ratio of peak width at half-height to baseline (RPW) for an individual peak.
min_snr_baseline	minimum S/N baseline for an individual peak.
max_R13C_integrated_peak	maximum allowed value of average R13C for an individual peak.
max_percentage_missing_scans	maximum allowed value of percentage missing scans on the raw chromatogram for an individual peak.
n_spline	number of points for further smoothing using a cubic spline smoothing method to calculate ancillary chromatographical parameters.

**Value**

a data frame consisting of 24 columns representing chromatography and mass spectrometry parameters. Each row represents an individual separated chromatographic peak.

---

reference\_peaks\_detector

*Reference peaks detector*

---

### Description

This function detects recurring reference peaks (m/z-RT) for retention time correction.

### Usage

```
reference_peaks_detector(input_path_peaklist, file_names_peaklist_ref,  
min_frequency_ref_peaks, mz_error, rt_tol, n_quantile, number_processing_threads = 1)
```

### Arguments

`input_path_peaklist` path to directory of peaklists.

`file_names_peaklist_ref` name of peaklists files to detect recurring reference peaks (m/z-RT).

`min_frequency_ref_peaks` minimum frequency of the recurring reference peaks (m/z-RT) in the reference files.

`mz_error` mass error to detect common peaks.

`rt_tol` retention time tolerance to detect common peaks.

`n_quantile` number of total m/z quantiles to split the whole table for faster processing.

`number_processing_threads` number of processing threads

### Value

`reference_mz_rt_peaks` a matrix of two columns of m/z and RT of common peaks in the reference samples.

`listRefRT` a list of corrected or uncorrected retention times for each peaklist.

---

sample\_rt\_corrector *sample retention time corrector*

---

### Description

This function calculates corrected retention times for the peaklists.

**Usage**

```
sample_rt_corrector(reference_mz_rt_peaks, peaklist, mz_error,
  rt_correction_method, reference_peak_tol = 1, polynomial_degree = 3)
```

**Arguments**

`reference_mz_rt_peaks`  
a matrix of reference peaks for retention time correction.

`peaklist`  
an IPA peaklist.

`mz_error`  
mass error to detect common reference peaks.

`rt_correction_method`  
This parameter can be either 'RetentionIndex' or 'Polynomial'.

`reference_peak_tol`  
number of reference peaks for retention time correction using 'RetentionIndex' method.

`polynomial_degree`  
polynomial degree for retention time correction using 'Polynomial' method.

**Value**

a list of corrected retention times for each peaklist.

---

segment	<i>segment</i>
---------	----------------

---

**Description**

illustrates an output matrix of chromatogram peak detection module from the "chromatogram\_builder.rda" object.

**Usage**

```
data("segment")
```

**Format**

The format is: num [1:16, 1:2] 7 15 23 33 38 46 67 86 102 118 ...

**Examples**

```
data(segment)
```

---

snr_rms	<i>SNR RMS</i>
---------	----------------

---

**Description**

This function calculates signal-to-noise ratio using root mean square.

**Usage**

```
snr_rms(int, baseline, gauge)
```

**Arguments**

int	is the vector of intensities corresponding to the vector of retention times for the chromatographic peak.
baseline	is a vector of baseline of the chromatographic peak.
gauge	represents the gauge height of peak for gaussianity measurement.

**Value**

S/N value

**Examples**

```
data("peak_spline")
int <- peak_spline[, 2]
baseline <- peak_spline[, 3]
gauge <- 0.8
snr_rms(int, baseline, gauge)
```

---

snr_signal2baseline	<i>SNR baseline</i>
---------------------	---------------------

---

**Description**

This function calculates S/N using local noise levels from baseline,

**Usage**

```
snr_signal2baseline(int, baseline)
```

**Arguments**

int	a vector of intensities corresponding to the vector of retention times for the chromatographic peak.
baseline	a vector of baseline of the chromatographic peak.

**Value**

S/N value

**Examples**

```
data("peak_spline")
int <- peak_spline[, 2]
baseline <- peak_spline[, 3]
snr_signal2baseline(int, baseline)
```

---

snr\_xcms

*SNR xcms*

---

**Description**

This function calculates S/N values using a method suggested in the xcms paper (Tautenhahn, 2008).

**Usage**

```
snr_xcms(int)
```

**Arguments**

**int** a vector of intensities corresponding to the vector of retention times for the chromatographic peak.

**Value**

S/N value

**References**

Tautenhahn, R., Böttcher, C. and Neumann, S. (2008). Highly sensitive feature detection for high resolution LC/MS. *BMC bioinformatics*, 9(1), 1-16, doi: [10.1186/147121059504](https://doi.org/10.1186/147121059504).

**Examples**

```
data(peak_spline)
int <- peak_spline[, 2]
snr_xcms(int)
```

---

spectralList\_filtering *spectralList filtering*

---

**Description**

This function reduces the size of the spectralList value by removing m/z values with no correspondence to 12C/13C isotopologue pairs.

**Usage**

```
spectralList_filtering(spec_scan.xic, spectralList, rounding_digit)
```

**Arguments**

spec\_scan.xic a matrix of any size, but the first column containing the m/z of 12C isotopologues are used.

spectralList a list of mass spectra in each chromatogram scan.

rounding\_digit rounding digit to choose power of size reduction.

**Value**

a list of mass spectrals

---

usp\_tailing\_factor *USP tailing factor*

---

**Description**

This function calculates USP tailing factor at above 10 percent of the height.

**Usage**

```
usp_tailing_factor(rt, int)
```

**Arguments**

rt a vector of retention times for the chromatographic peak.

int a vector of intensities corresponding to the vector of retention times for the chromatographic peak.

**Value**

USP tailing factor for the chromatographic peak.



**Examples**

```
data(peak_spline)
rt <- peak_spline[, 1]
int <- peak_spline[, 2] - peak_spline[, 3]
usp_tailing_factor(rt, int)
```

---

XIC

*XIC*


---

**Description**

XIC

**Usage**

```
XIC(spectralList.xic, scan_number_start = 1, mz_target, mass_accuracy_xic)
```

**Arguments**

`spectralList.xic` a list of mass spectra in each chromatogram scan.

`scan_number_start` the first scan number.

`mz_target` an m/z value to perform XIC analysis.

`mass_accuracy_xic` a mass error to perform XIC analysis.

**Value**

A matrix of three columns representing scan number, m/z, and intensity.

---

xlsxAnalyzer\_EIC

*xlsxAnalyzer EIC*


---

**Description**

This function processes the spreadsheet of the IPA parameters to ensure the parameter inputs are in agreement with the IPA\_EIC requirements.

**Usage**

```
xlsxAnalyzer_EIC(spreadsheet)
```

**Arguments**

`spreadsheet` contains the IPA parameters.

**Value**

This function returns the IPA parameters to feed the IPA\_TargetedAnalysis function.

# Index

## \* datasets

- chromatogram\_builder, 4
- peak\_spline, 21
- segment, 29
  
- asymmetry\_factor, 3
  
- baseline\_developer, 3
  
- chromatogram\_builder, 4
- chromatography\_analysis, 5
  
- der\_5points\_stencil, 7
- derivative\_skewness, 6
  
- EIC\_plotter, 7
  
- fronting\_tailing\_resolver, 8
  
- gaussianity\_measurement, 9
  
- IPA\_CompoundsAnnotation, 10
- IPA\_GapFiller, 10
- IPA\_IsotopePairing, 11
- IPA\_MSdeconvoluter, 11
- IPA\_PeakAlignment, 12
- IPA\_PeakAnalyzer, 12
- IPA\_PeaklistAnnotation, 13
- IPA\_TargetedAnalysis, 13
- IPA\_Workflow, 14
- IPA\_xlsxAnalyzer, 15
- islocalminimum, 16
- islocaloptimum, 16
  
- loadRdata, 17
  
- mz\_clustering\_xic, 18
- mzRTindexer, 17
  
- opendir, 18
  
- peak\_alignment, 19
  
- peak\_area, 19
- peak\_detection, 20
- peak\_sharpness, 21
- peak\_spline, 21
- peak\_width, 22
- peak\_Xcol2, 22
- plot\_mz\_eic, 23
- plot\_simple\_tic, 24
- primary\_peak\_analyzer, 24
- pseudomoments\_symmetry, 26
  
- recursive\_mass\_correction, 26
- reference\_peaks\_detector, 28
  
- sample\_rt\_corrector, 28
- segment, 29
- snr\_rms, 30
- snr\_signal2baseline, 30
- snr\_xcms, 31
- spectralist\_filtering, 32
  
- usp\_tailing\_factor, 32
  
- XIC, 33
- xlsxAnalyzer\_EIC, 33