Package 'lans2r'

July 22, 2025

Title Work with Look at NanoSIMS Data in R

Description R interface for working with nanometer scale secondary ion mass spectrometry (NanoSIMS) data exported from Look at NanoSIMS.

Version 1.2.0

URL https://github.com/KopfLab/lans2r

BugReports https://github.com/KopfLab/lans2r/issues

Depends R (>= 3.1)

Imports dplyr, tidyr, reshape2, ggplot2, rlang, R.matlab

Suggests covr, testthat, knitr, rmarkdown

License GPL-3 | file LICENSE

Encoding UTF-8

VignetteBuilder knitr, rmarkdown

RoxygenNote 7.2.3

NeedsCompilation no

Author Sebastian Kopf [aut, cre]

Maintainer Sebastian Kopf <sebastian.kopf@colorado.edu>

Repository CRAN

Date/Publication 2023-02-19 07:20:02 UTC

Contents

`calculate` *Calculate derived data*

Description

This function allows easy calculation of any quantities derived from other variables. The new quantities can be assigned to a specific data_type and values, errors as well as the resulting variable names are calculated/constructed based on custom functions that can be provided via the function parameters. calculate_sums, calculate_ratios and calculate_abundances are all based on this and provide an easy way for common standard calculations.

Usage

```
calculate(
  data,
  data_type,
  ...,
  value_fun,
  error_fun = function(...) return(NA),
  name_fun = default_name,
  filter_new = NULL,
  quiet = FALSE
)
```

Arguments

`data`	a data frame with lans2r data, can be grouped to do calculations within individual groups
`data_type`	what to call the new data type
`...`	the parameters to send to the value, error and naming function for each derived value. These are always expressions that can include references to variable columns, arithmetic and constants, e.g. c(`12C`, `13C`) or c("test", 100*(`12C`+`13C`)). The number of parameters needs to match those expected by the value, error and name functions. Error values of different columns (say for classical error propagation) can be addressed using the suffix "sigma", e.g. c(`12C`, `12C sigma`) would pass both the value and error of this variable to the functions.
`value_fun`	a custom function used to calculate the derived value - needs to match the sets of parameters provided through ...
`error_fun`	a custom function used to calculate the error (sigma) for the derived value - needs to match the sets of parameters provided through ...
`name_fun`	a custom function used to construct the variable name for the derived quantity - needs to match the sets of parameters provided through ...
`filter_new`	an expression to apply as a filter on the new data rows (e.g. plane == "all")
`quiet`	whether the function should output information messages or be quiet (default is to output)

Value

the original data frame with the newly calculated information appended (data_type == "ion_sum")

See Also

Other calculations: `calculate_abundances()`, `calculate_ratios()`, `calculate_sums()`

`calculate_abundances` *Calculate isotope fractional abundances*

Description

This function calculates the isotope abundances (in %!) and resulting counting statistics error from the raw ion counts. It can be applied to data from both LANS_summary and LANS_maps loading but can be slow if LANS_maps is combined from many analyses.

Usage

```
calculate_abundances(data, ..., name_fun = default_name, quiet = FALSE)
```

Arguments

`data`	a data frame with raw ion counts retrieved from `load_LANS_summary`
`...`	the fractional abundances to calculate, each entry is for one fractional abundance with major isotope first, then minor isotope, e.g. c(`13C`, `12C`), c(`15N12C`, `14C12C`), ...
`name_fun`	the naming function, receives ... from the top level, default concatenates 'F' + minor ion name
`quiet`	whether the function should output information messages or be quiet (default is to output)

Value

the original data frame with the fractional abundance information appended (all fractional abundances are in % and have data_type == "abundance")

See Also

Other calculations: `calculate_ratios()`, `calculate_sums()`, `calculate()`

calculate_ratios *Calculate isotope ratios*

Description

This function calculates the ratios and resulting counting statistics error from the raw ion counts. It can be applied to data from both LANS_summary and LANS_maps loading but can be slow if LANS_maps is combined from many analyses. It can also be applied to ion_sums generate by calculate_sums to calculate elemental ratios (careful, ionization efficiencies skew their scaling!)

Usage

```
calculate_ratios(data, ..., name_fun = default_name, quiet = FALSE)
```

Arguments

data	a data frame with raw ion counts retrieved from load_LANS_summary
...	the ratios to calculate, each entry is one ratio with major isotope first, then minor isotope, e.g. c(13C, 12C), c(15N12C, 14C12C), ...
name_fun	the naming function, receives ... from the top level, default concatenates column names with '/'
quiet	whether the function should output information messages or be quiet (default is to output)

Value

the original data frame with the ratio information appended (all ratios have data_type == "ratio")

See Also

Other calculations: calculate_abundances(), calculate_sums(), calculate()

calculate_sums *Calculate ion sums*

Description

This function calculates the ion sums and resulting counting statistics error from multiple raw ion counts. It can be applied to data from both LANS_summary and LANS_maps loading but can be slow if LANS_maps is combined from many analyses. Careful about its error propagation, it assumes it is calculating sums of ions and uses the ion counts themselves for error calculation. This is not suitable for calculating other types of sums where other types of error propagation may be more appropriate.

Usage

```
calculate_sums(data, ..., name_fun = default_name, quiet = FALSE)
```

Arguments

data	a data frame with raw ion counts retrieved from load_LANS_summary
...	the ion sums to calculate, each entry is for one sum of as many ions as desired, e.g. c(13C, 12C), c(15N12C, 14C12C), ...
name_fun	the naming function, receives ... from the top level, default concatenates column names with '+'
quiet	whether the function should output information messages or be quiet (default is to output)

Value

the original data frame with the sums information appended (data_type == "ion_sum")

See Also

Other calculations: calculate_abundances(), calculate_ratios(), calculate()

---

load_HMR *Read data from HMR txt files*

Description

See HMR vignette for examples.

Usage

```
load_HMR(folder, prefix, suffix = ".hmr_txt")
```

Arguments

folder	• the folder where the HMR files are
prefix	• the prefix for the set of hmr files
suffix	• the suffix for the set of hmr files

load_LANS_maps *Load LANS ion map data*

Description

Load the full ion map data (incl. ROI locations) exported from LANS analyses and attach additional information to each analysis. Uses read_map_data to read individual matlab export files.

Usage

```
load_LANS_maps(
  analysis,
  ...,
  base_dir = ".",
  ion_data_only = TRUE,
  quiet = FALSE
)
```

Arguments

analysis	• vector of LANS analysis folder names
...	• vectors of additional information to attach to each analysis, each argument has to have the same length as the 'analysis' parameter (or length 1)
base_dir	• the directory where all the analysis folders are located (defaults to current directory)
ion_data_only	• whether to import only ion data (TRUE by default), rather than any derived files (e.g. ratios calculated within LANS). Recommend using calculate_ratios and calculate_abundances to process the raw ion counts in easy format and good error propagation.
quiet	• whether to report information on the loaded data or not

load_LANS_summary *Load LANS summary data*

Description

Load the ROI summary information exported from LANS analyses and attach additional information to each analysis. Uses read_roi_data to read individual files.

Usage

```
load_LANS_summary(
  analysis,
  ...,
  base_dir = ".",
  ion_data_only = TRUE,
  load_zstacks = TRUE,
  quiet = FALSE
)
```

Arguments

analysis	• vector of LANS analysis folder names
...	• vectors of additional information to attach to each analysis, each argument has to have the same length as the 'analysis' parameter (or length 1)
base_dir	• the directory where all the analysis folders are located (defaults to current directory)
ion_data_only	• whether to import only ion data (TRUE by default), rather than any derived files (e.g. ratios calculated within LANS). Recommend using calculate_ratios and calculate_abundances to process the raw ion counts in easy format and good error propagation.
load_zstacks	• whether to load the planes data (ion-z.dat files need to be exported from LANS for this to be possible - they are created when the "Display depth profiles in ROI" is checked during "Display masses")
quiet	• whether to report information on the loaded data or not

---

plot_maps *Plot NanoSIMS ion maps*

Description

Helps to plot the ion maps exported and loaded from LANS. Can overlay the ROI boundaries for clarity as well. Note that this does not currently support any smoothing yet so plotting ratios or abundances will most likely not work well because individual pixels have extreme values.

Usage

```
plot_maps(
  data,
  draw_ROIs = TRUE,
  normalize = TRUE,
  color_scale = c("black", "white")
)
```

Arguments

data	the ion maps data frame
draw_ROIs	whether to draw in the ROIs or not (default TRUE)
normalize	whether to normalize the intensity scale for each panel (default TRUE)
color_scale	what color scale to use for high and low intensity, default is black & white

---

read_map_data *Read LANS full ion map data (.mat) files*

Description

Reads the full matlab data files (.mat) that contain the complete ion maps and ROI outlines for the given LANS analysis folder and returns the data in a concatenated data frame with identifier column 'variable' (=ion), data columns value (ion count) and sigma (error based on counting statistics). Additionally, the column 'ROI' indicates which ROI each pixel belongs to with a value of 0 indicating that it does not belong to any ROI. Note that this only reads ion data files by default and not any derived data files (any ratio or other formulas evaluated by LANS). It does also not currently support z-stacks yet.

Usage

```
read_map_data(mat_folder, ion_data_only = TRUE, quiet = FALSE)
```

Arguments

mat_folder	• the LANS mat directory with the ions' .mat files
ion_data_only	• by default TRUE, i.e. ignores all non-ion data files
quiet	• whether to report information on the loaded data or not

Value

concatenated data_frame with the full ion maps data

read_roi_data *Read LANS data summary (.dac/.dat) files*

Description

Reads the ion data (.dac/.dat) files for the given LANS analysis folder and returns the ROIs data in a concatenated data frame with identifier columns 'ROI' and 'variable' (=ions). Note that this only reads ion data files by default and not any derived data files (any ratio or other formulas evaluated by LANS). If zstacks (i.e. individual planes) are exported from LANS, they can be loaded as well (and are by default). The resulting data frame has a 'plane' column that keeps track of the plane, the value 'all' identifies the combined data for the ROI from all planes.

Usage

```
read_roi_data(
  dat_folder,
  ion_data_only = TRUE,
  load_zstacks = TRUE,
  quiet = FALSE
)
```

Arguments

dat_folder	• the LANS dat directory with the ions' .dac files
ion_data_only	• by default TRUE, i.e. ignores all non-ion data files
load_zstacks	• whether to load any z_stacks found, by default TRUE
quiet	• whether to report information on the loaded data or not

Value

concatenated data_frame with all the ROIs' data, with identifier columns 'plane', 'ROI' and 'variable'

spread_data *Spread data into wide format*

Description

This function allows easy spreading into wide format.

Usage

```
spread_data(data, values = TRUE, errors = TRUE)
```

Arguments

`data`	a data frame with lans2r data
`values`	whether to include the values in wide format
`errors`	whether to include the errors in wide format

Value

the original data frame but in wide format

Index