Package: specprepper (via r-universe)

August 26, 2024

Title Plan and Apply Chained Preprocessing Operations on Spectra

Version 0.3.5

Description Schedule and perform common spectroscopic signal processing (preprocessing) methods using a recipe-style syntax. Combine different operations in sequence.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

Roxygen list(markdown = TRUE)

RoxygenNote 7.2.3

Imports prospectr, future.apply, data.table, matrixStats

Config/rextendr/version 0.3.1

Suggests Rarr

Repository https://spectral-cockpit.r-universe.dev

RemoteUrl https://github.com/spectral-cockpit/specprepper

RemoteRef HEAD

RemoteSha ac6dd84e9a0f8b92162d8f49a9413742937acd3e

Contents

colmean_group_apply *Compute the mean spectra per group label for all spectra in a collection*

Description

The function can be applied to spectral collections, dt_prep_sets. The list-column id_labels with lists of data.tables each containing a column named group must be present. See also ids_apply().

Usage

```
colmean_group_apply(dt_prep_sets, append_rows = FALSE)
```

Arguments

dt_prep_sets	A standardized data.table, i.e. returned from specprepper::*_apply() function. Contains labelled sets of preprocessed spectra: This argument allows to chain preprocessing in sequential manner, and i.e. apply variable Savitzky-Golay smoothers with a single function application.
append_rows	logical whether to append the newly processed rows, when dt_prep_sets is not NULL.

Details

A spectral collection typically represents an outcome of one or more specific preprocessing with methods and possibly associated parameters used. colmean_group_apply() only accepts collections with structural conventions of dt_prep_sets. It requires a id_labels list-column with a group column specifying the lables used for aggregation in each data.table element (one for each collection). Label columns such as row or id that were present before will be removed because they are assumed to be aggregated.

Value

- A "data.table" with as many rows as spectral collections. It contains at least the following columns:
  - prep_set: appends "-mean_group" to the exisiting character vector elements of the input data.
  - prep_label: appends "mean_group" to the exisiting character vector elements of the input data.
  - prep_params: A list-column with 1-row data.table's. Each data.table has a new column mean_group, contains the string "id_labels$group".
  - id_labels: This list-column now only contains a sliced version of the group column, that correspond to the new rows of the aggregated column means in spc_prep.
  - spc_prep: A list-column with data.tables that contain aggregated means of spectra by group for each spectral collection (row of dt_prep_sets)

ids_apply *Add atomic vector labels for row, id and group for all rows of spectra.*

Description

Adds labels to rows of all individual spectra in collections. Such labels are required for subsequent processing functions that aggregate spectral collections by group, for example `colmean_group_apply()`. It can also be used to initialize a single spectral collection with labels when inputting a single matrix, data frame or data.table.

Usage

```
ids_apply(X, dt_prep_sets = NULL, vec_row, vec_id, vec_group)
```

Arguments

`X`	`matrix`, `data.frame` or `data.table` for which label rows are to be applied.
`dt_prep_sets`	A standardized `data.table`, i.e. returned from `specprepper::*_apply()` function. Contains labelled sets of preprocessed spectra: This argument allows to chain preprocessing in sequential manner, and i.e. apply variable Savitzky-Golay smoothers with a single function application.
`vec_row`	atomic vector with row labels; need to have same length as `nrow(X)` or rows in all `spc_proc` list-column data.table's.
`vec_id`	atomic vector with id labels, needs to have same length as `nrow(X)` or rows in all `spc_proc` list-column data.table's. `id_vec` typically represents the smallest hiararchical unit in the measurement design, e.g., a replicate spectrum measured.
`vec_group`	atomic vector with group labels; needs to have same length as `nrow(X)` or rows in all `spc_proc` list-column data.table's. `id_group` typically represents the group to aggregate by in specific methods applied later. Currently, this is `colmean_group_apply()` that takes grouped means of spectra or collections of spectra.

Value

If `X` is specified:

- A one-row `"data.table"` with the following columns
  - `prep_set`: `"init_ids"`,
  - `prep_label`: `"prep_label"`
  - `prep_params`: list-column of length 1 with `"data.table"` containing `init_ids` = `NA`
  - `id_labels`: list-column (repeated across rows) with `"data.table"` containing columns with labels: `row` (from `vec_row`), `id` (from `vec_id`), and `group` (from `vec_group`). If `dt_prep_sets` is specified:

- A "data.table" with as many rows as spectral collections. A spectral collection typically represents an outcome of one or more specific preprocessing with methods and possibly associated parameters used. Specifically, it augments the input dt_prep_sets and outputs the following (list-)columns:
  - prep_set: appends "-init_ids to the input string that states what the main preprocessings done in previous steps.
  - prep_label: appends "-init_ids to the input string that states what was done with abbreviations of methods in previous steps.
  - prep_params: augments each data.table element in the list-column with a new non-specific column init_ids = NA (indicating a new label column but no direct effect on the processed spectra).
  - id_labels: new list-column that contains a set of labels that applies for all spectral collections nested within respective rows of the dt_prep_sets input. Each data.table in the list contains the label columns row (from vec_row),id (from vec_id), and group (from vec_group).
  - spec_prep: unmodified list-column with sets of already prepared, processed spectra. Each element is a data.table which rows corresponds to the row labels in id_labels.

---

sg_apply *Combinatory Savitzky-Golay Filtering*

Description

Apply Savitzky-Golay filtering at variable combinations of parameter sets for set(s) of spectra.

Usage

```
sg_apply(
  X,
  dt_sg_plan,
  dt_prep_sets = NULL,
  nest_params = TRUE,
  append_rows = FALSE
)
```

Arguments

X	matrix, data.frame or data.table with spectra to be preprocessed according to plan (see dt_sg_plan).
dt_sg_plan	A standardized data.table with the Savitzky-Golay parameter sets, which can be generated with sg_plan(). It must at least contain the following columns: • prep_set (character) • prep_label (character) • m (integer): order of the derivative; m = 0 signifies no derivative • p (integer): polynomal order

- w (integer): window size in number of spectral points; must be uneven m is the , p is the polynomial order that should be bigger than the derivative order, and w is the window size in number of spectral points (must be uneven). See section *Savitzky-Golay Plan* for templating the required object and prospectr::savitzkyGolay() for further the original Savitzky-Golay algorithm.

dt_prep_sets	A standardized data.table, i.e. returned from specprepper::*_apply() function. Contains labelled sets of preprocessed spectra: This argument allows to chain preprocessing in sequential manner, and i.e. apply variable Savitzky-Golay smoothers with a single function application.
nest_params	logical whether to nest the Savitzky-Golay parameters in a prep_params list-column.
append_rows	logical whether to append the newly processed rows, when dt_prep_sets is not NULL.

Details

Design principles:

Savitzky-Golay transformation (moving window polynomial least-squares) prior modeling can help to reduce noise and enhance signals in spectra. This can allowing models to extract parsimonious predictable information from spectra for more accurate estimation. However, this process requires empirical optimization and fine-tuning of the parameters that control the nature and degree of smoothing and hence noise removal for calibration task at hand, which is often not done. For example, systematically varying the size of the smoothing window control the amount of information filtered and potential artefacts created. Nonetheless, non-stationary noise as opposed to white gaussian noise and informative fluctuations in chemically-driven spectral dynamics (e.g. slope changes and different absorption peak widths and compositional complexity) can make a simple nonrecursive application of the original Savitzky-Golay algorithm less appropriate to filter noise.

Templating code for sequential and/or recursive branching of preprocessing methods with variation their parameters, if applicable, can be repetitive and cumbersome. This is where the specprep package with combinatory planning and application tools jumps in.

The combinatory power of the sg_apply() function stems from the ability to map Savitzly Golay both over row-wise sets of parametrizations (see subsection *Savitzky-Golay Plan*) and previous preprocessing rounds that yielded set(s) of (differently) processed spectra to be processed again (see section *Set(s) of Previously Processed Spectra*). Since data.tables are structured consistently across the specprep::*_apply type of functions, their inputs and outputs are interoperable. This allows flexiblity for applying combinations of preprocessing methods.

Savitzky-Golay Plan:

dt_sg_plan is most conventiently built with sg_plan(). It parametrizes Savitzky-Golay preprocessing scheduled on either X or on all sets of already processed spectra contained in dt_prep_sets. Each row lays out one preprocessing step, linking the following data across columns:

- prep_set: this string identifies the name of general preprocessing method that is chained to sets of spectra.

Set(s) of Previously Processed Spectra:

tbd

Value

data.table with the following (list)columns:by #to be filled

Author(s)

Philipp Baumann

`sg_make_plan` *Generate a data.frame with Savitzky-Golay parameters*

Description

Make a full-factorial combination of Savitzky-Golay parameters.

Usage

```
sg_make_plan(param_list)
```

Arguments

`param_list`	A list of

Value

data.frame

Author(s)

Philipp Baumann

`snv_apply` *Compute the standard normal variate for collections of spectra*

Description

Compute the standard normal variate for collections of spectra

Usage

```
snv_apply(X, dt_prep_sets = NULL, append_rows = FALSE)
```

Arguments

X	matrix, data.frame or data.table with spectra used as input to compute standard normal variate (SNV)
dt_prep_sets	A standardized data.table, i.e. returned from specprepper::*_apply() function. Contains labelled sets of preprocessed spectra: This argument allows to chain preprocessing in sequential manner, and i.e. apply variable Savitzky-Golay smoothers with a single function application.
append_rows	logical whether to append the newly processed rows, when dt_prep_sets is not NULL.

Index