

Package: sapid (via r-universe)

October 13, 2026

Title A Strategy to Analyze Plant Extracts Taste In Depth

Version 0.0.0.9002

Author Adriano Rutz

Maintainer Adriano Rutz <adafede@gmail.com>

Description This package provides the infrastructure to implement a Strategy to Analyze Plant Extracts Taste In Depth.

License GPL (>= 3)

URL <https://github.com/adafede/sapid>, <https://adafede.github.io/sapid>

BugReports <https://github.com/adafede/sapid/issues>

Depends R (>= 4.5.0)

Imports AlpsNMR (>= 4.12.0), BSDA (>= 1.2.2), cascade (>= 0.0.0.9002), dendextend (>= 1.19.1), FactoMineR (>= 2.16), forcats (>= 1.0.1), ggplot2 (>= 4.0.3), ggpubr (>= 1.0.0), ggraph (>= 2.2.2), ggrepel (>= 0.9.8), igraph (>= 2.3.3), khroma (>= 1.17.0), purrr (>= 1.2.2), readxl (>= 1.5.0), scales (>= 1.4.0), SensoMineR (>= 1.28), stats, tibble (>= 3.3.1), tidytibble (>= 0.11.2), treemapify (>= 2.6.1), utils

Remotes adafede/cascade, taxonomicallyinformedannotation/tima

Suggests mzR (>= 2.44.0), NMRphasing (>= 1.0.7), R.utils (>= 2.13.0), tinytest (>= 1.4.3), withr (>= 3.0.2)

ByteCompile TRUE

Collate 'colors.R' 'cluster_nmr.R' 'correlate_ion_taste_intensities.R' 'data.R' 'geom_sigmoid.R' 'get_session_info.R' 'globals.R' 'harmonize_terms.R' 'harmonize_terms_df.R' 'load_consistent_profiles.R' 'load_session.R' 'plot_chasselas_modulation.R' 'plot_chasselas_variation.R' 'plot_chromatograms.R' 'plot_concentration.R' 'plot_correlations.R' 'plot_descriptors_network.R' 'plot_informed_tasting.R' 'plot_matrices.R' 'plot_napping.R' 'plot_profiles.R' 'prepare_chasselas.R' 'prepare_chemical_profiles.R' 'prepare_concentration.R' 'prepare_napping.R' 'prepare_profiles.R' 'sapid-package.R'

Config/roxygen2/markdown TRUE
Config/roxygen2/version 8.1.0
Encoding UTF-8
Language en-US
LazyData TRUE
LazyDataCompression bzip2
biocViews computationalMetabolomics, naturalExtracts, taste
X-schema.org-keywords computationalMetabolomics, naturalExtracts, taste
Config/pak/sysreqs cmake libfontconfig1-dev libfreetype6-dev libglpk-dev libglu1-mesa-dev make texlive libarchive-dev libicu-dev libjpeg-dev libpng-dev libuv1-dev libxml2-dev libnetcdf-dev libgl1-mesa-dev libssl-dev python3 libnode-dev zlib1g-dev
Repository <https://adafede.r-universe.dev>
Date/Publication 2026-10-13 18:59:50 UTC
RemoteUrl <https://github.com/adafede/sapid>
RemoteRef main
RemoteSha 79e80857471a4d1ffd6662e321cd561fbf4731f5

Contents

chasselas	3
cluster_nmr	4
concentration_afc	5
correlate_ion_taste_intensities	6
dictionary_generic	7
dictionary_napping	7
dictionary_specific	8
fractions	9
geom_sigmoid	10
get_session_info	11
groups	11
harmonize_terms	12
harmonize_terms_df	13
load_consistent_profiles	14
load_session	15
napping_coordinates	15
napping_descriptors	16
plot_chasselas_modulation	17
plot_chasselas_variation	17
plot_chromatograms	18
plot_concentration	19
plot_correlations	19
plot_descriptors_network	21
plot_matrices	21

plot_napping	22
plot_profiles	22
prepare_chasselas	24
prepare_chemical_profiles	24
prepare_concentration	25
prepare_napping	26
prepare_profiles	27
profiles	27
sapid-data	28
sigmoid	29
Index	30

chasselas	<i>Chasselas Sensory Panel Taste Ratings</i>
-----------	--

Description

A dataset of sensory panel ratings for Chasselas wine samples, collected during multiple sessions with multiple jurors.

Usage

chasselas

Format

- A data frame with 1,314 rows and 6 columns:
- date** ISO 8601 timestamp of the rating session
 - jury** Juror identifier (e.g., "jury_01")
 - product** Product identifier ("product_1before", "product_2after")
 - session** Session identifier (e.g., "session_06")
 - taste** Taste attribute or descriptor (e.g., "sourness", "bitterness")
 - value** Numerical rating value for the attribute

Source

[doi:10.1016/j.crfs.2025.101043](https://doi.org/10.1016/j.crfs.2025.101043); Zenodo <https://zenodo.org/records/14616396>

Examples

```
data(chasselas)
head(chasselas)
table(chasselas$taste)
```

cluster_nmr

*Cluster NMR***Description**

Cluster NMR

Usage

```
cluster_nmr(
  nmr_dir = "./data/10043",
  output_figure = "./data/figures/figure_nmr.pdf",
  output_groups = "./inst/extdata/groups.tsv",
  experiment_raw = "proton_00",
  experiment_ref = "proton_98",
  experiments_fractions = c("proton_17", "proton_18", "proton_19", "proton_20",
    "proton_21", "proton_22", "proton_23", "proton_24", "proton_25", "proton_26",
    "proton_27", "proton_28", "proton_29", "proton_30", "proton_31", "proton_32",
    "proton_33", "proton_34", "proton_35", "proton_36", "proton_37", "proton_38",
    "proton_39", "proton_40", "proton_41", "proton_42", "proton_43", "proton_44",
    "proton_45", "proton_46", "proton_47", "proton_48", "proton_50", "proton_51",
    "proton_52", "proton_53", "proton_54", "proton_55", "proton_56",
    "proton_57",
    "proton_58", "proton_59", "proton_60", "proton_61", "proton_62", "proton_63",
    "proton_64", "proton_65", "proton_66", "proton_67", "proton_68", "proton_69",
    "proton_70", "proton_71"),
  experiments_to_fix = c("proton_69", "proton_70", "proton_71"),
  k = 7,
  peak_width_ppm = 0.01,
  ppm_min = 0,
  ppm_max = 14.85,
  area_min = 0,
  range_exclude_area = c(2.5, 2.6),
  range_without_peaks = c(0, 0.4),
  ylim = c(0, 5e+10)
)
```

Arguments

nmr_dir	NMR directory
output_figure	Output figure
output_groups	Output groups
experiment_raw	Experiment raw
experiment_ref	Experiment ref
experiments_fractions	Experiments fractions

experiments_to_fix	Experiments to fix
k	Number of clusters
peak_width_ppm	Peak width ppm
ppm_min	Ppm min
ppm_max	Ppm max
area_min	Area min
range_exclude_area	Range exclude area
range_without_peaks	Range without peaks
ylim	Y lim

Examples

NULL	
------	--

concentration_afc	Concentration AFC (Ascending Forced Choice) Test Results
-------------------	--

Description

Results from an Ascending Forced Choice (AFC) sensory test with concentration-dependent taste evaluation. Includes AFC accuracy metrics (number correct out of total trials).

Usage

concentration_afc

Format

- A data frame with 76 rows and 6 columns:
- concentration** Concentration level of the tested compound
 - jury** Juror identifier (e.g., "jury_01")
 - taste** Taste attribute tested (e.g., "amer", "acide")
 - value** Numerical rating or intensity value
 - afc_correct** Number of correct responses in the AFC trial
 - afc_total** Total number of trials in the AFC session

Source

doi:10.1016/j.crfs.2025.101043; Zenodo <https://zenodo.org/records/14616396>

Examples

```
data(concentration_afc)
head(concentration_afc)
summary(concentration_afc)
```

correlate_ion_taste_intensities
<i>Correlate ion taste intensities</i>

Description

Correlate ion taste intensities

Usage

```
correlate_ion_taste_intensities(
  input_ions = "../data/fractions_mzmine/fractions.csv",
  input_tastes = NULL,
  output = "../data/correlations.tsv",
  min_jury = 2L,
  min_area_ion = 1L,
  imputation_factor = 0.5,
  widths = 5:9
)
```

Arguments

input_ions	Input ions
input_tastes	Input tastes
output	Output
min_jury	Min jury
min_area_ion	Min area ion
imputation_factor	Imputation factor
widths	Widths

Examples

NULL

dictionary_generic	<i>Generic Taste Terms Dictionary</i>
--------------------	---------------------------------------

Description

A reference dictionary for mapping taste descriptors to standardized terms. Includes original, translated, and simplified versions. Designed for generic/widely-used sensory attributes.

Usage

```
dictionary_generic
```

Format

A data frame with 64 rows and 3 columns:

original Original or raw descriptor (often in French)

translated English translation

translated_simple Simplified/harmonized English form

Details

Used by data harmonization functions (e.g., `harmonize_terms()`) to standardize taste terminology across panelists and sessions.

Source

doi:10.1016/j.crfs.2025.101043; Zenodo <https://zenodo.org/records/14616396>

Examples

```
data(dictionary_generic)
head(dictionary_generic)
# Example lookup
subset(dictionary_generic, original == "BOISÉ")
```

dictionary_napping	<i>Napping-Specific Taste Terms Dictionary</i>
--------------------	--

Description

A reference dictionary specifically for napping session terminology, mapping napping-specific descriptors to standardized terms.

Usage

```
dictionary_napping
```

Format

A data frame with 45 rows and 2 columns:

- original** Original napping descriptor
- translated** Standardized descriptor

Details

Napping methods often generate unique terminology or abbreviations (e.g., "_LEGER" for "light"). This dictionary maps these to harmonized forms used in broader analyses.

Source

doi:10.1016/j.crfs.2025.101043; Zenodo <https://zenodo.org/records/14616396>

Examples

```
data(dictionary_napping)
head(dictionary_napping)
# Find mappings for "LEGER"
subset(dictionary_napping, grepl("LEGER", original))
```

dictionary_specific	<i>Specific Taste Terms Dictionary</i>
---------------------	--

Description

A reference dictionary for specific or less-common taste descriptors, including mappings to English translations and simplified forms.

Usage

dictionary_specific

Format

A data frame with 196 rows and 3 columns:

- original** Original specific descriptor (often in French)
- translated** English translation
- translated_simple** Simplified/harmonized English form

Details

Larger and more specialized than dictionary_generic. Used for comprehensive harmonization of all descriptors in the dataset.

Source

doi:10.1016/j.crfs.2025.101043; Zenodo <https://zenodo.org/records/14616396>

Examples

```
data(dictionary_specific)
head(dictionary_specific)
nrow(dictionary_specific)
```

fractions

Swertia chirayita Sample Fractionation Hierarchy

Description

Hierarchical record of sample preparation and fractionation of *Swertia chirayita*, documenting parent-child relationships and mass (in milligrams) at each step. Useful for tracking sample lineage and mass balance through extraction and purification.

Usage

```
fractions
```

Format

A data frame with 94 rows and 3 columns:

parent Parent fraction identifier (NA for root material)

label Child fraction or processing step identifier

mass Mass in milligrams at this stage

Details

The hierarchy starts with raw plant material and initial solvents (EtOH, water) and traces through extraction, decanting, drying, and sub-fractionation steps leading to final analytic samples (V_01, V_02, etc.).

Source

doi:10.1016/j.crfs.2025.101043; Zenodo <https://zenodo.org/records/14616396>

Examples

```
data(fractions)
head(fractions)
# View the hierarchy structure
subset(fractions, !is.na(parent))
```

geom_sigmoid	<i>geom_sigmoid</i>
--------------	---------------------

Description

geom_sigmoid

Usage

```
geom_sigmoid(  
  mapping = NULL,  
  data = NULL,  
  geom = "line",  
  position = "identity",  
  na.rm = FALSE,  
  show.legend = NA,  
  smooth = 8,  
  direction = "x",  
  inherit.aes = TRUE,  
  ...  
)
```

Arguments

mapping	provide you own mapping. both x, xend, y and yend need to be numeric.
data	provide you own data
geom	xhange geom
position	change position
na.rm	remove missing values
show.legend	show legend in plot
smooth	how much smooth should the curve have? More means steeper curve.
direction	the character x or y depending of smoothing direction
inherit.aes	should the geom inherits aesthetics
...	other arguments to be passed to the geom

Value

ggplot layer

get_session_info	Get session info
Description	
Get session info	
Usage	
get_session_info(session)	
Arguments	
session	Session
Examples	
NULL	
groups	Sample Grouping and Cluster Assignments

Description	
Cluster or group assignments for sample fractions, including group identifiers (typically color codes used in visualizations) and corresponding sample row identifiers.	
Usage	
groups	
Format	
A data frame with 54 rows and 2 columns:	
group	Group identifier, often a color code (e.g., "#e31a1c")
rowname	Sample or fraction row identifier (e.g., "17")
Details	
Used for coloring plots and visualizations based on clustering or hierarchical analysis results. Typically derived from dendrogram or other unsupervised classification of sample sensory profiles.	
Source	
doi:10.1016/j.crfs.2025.101043; Zenodo https://zenodo.org/records/14616396	

Examples

```
data(groups)
head(groups)
unique(groups$group)
```

harmonize_terms	<i>Clean terms</i>
-----------------	--------------------

Description

Replace terms in a given string using a dictionary, with intelligent matching and case-insensitive handling.

Usage

```
harmonize_terms(dictionary, x, mode = "word", fallback = FALSE)
```

Arguments

dictionary	Path to the dictionary file, a data frame containing replacement terms, or NULL to use cached environment dictionaries
x	Character string to be cleaned
mode	Replacement mode: 'word' for word boundary matching, 'substring' for partial matching, 'fixed' for fast fixed-string matching
fallback	Logical, whether to fall back to original string if no replacements occur

Details

For better performance when calling this function repeatedly on many strings, pass a pre-loaded dictionary data frame instead of a file path. The function uses base R replacement helpers for all modes to avoid an extra dependency.

Value

Character string with terms replaced

Examples

```
## Not run:
# Using data frame dictionary (faster)
data(dictionary_generic)
harmonize_terms(
  dictionary = dictionary_generic,
  x = "Some text with specific terms"
)

# Using file path (legacy, slower)
```

```
harmonize_terms(  
  dictionary = "path/to/dictionary.csv",  
  x = "Some text with specific terms"  
)  
  
## End(Not run)
```

harmonize_terms_df	<i>Harmonize terms in a data frame</i>
--------------------	--

Description

Apply harmonization to multiple taste terms in a data frame using provided dictionaries.

Usage

```
harmonize_terms_df(  
  df,  
  dictionary_generic_path,  
  dictionary_napping_path,  
  dictionary_specific_path  
)
```

Arguments

df	Data frame containing taste/descriptor columns
dictionary_generic_path	Path to generic dict or data frame object
dictionary_napping_path	Path to napping dict or data frame object
dictionary_specific_path	Path to specific dict or data frame object

Value

Data frame with harmonized taste terms

Examples

```
NULL
```

`load_consistent_profiles`*Load consistent profiles*

Description

Filter profiles to keep only taste descriptors that are consistently used across multiple panelists (minimum jury threshold).

Usage

```
load_consistent_profiles(input, min_jury = 2L)
```

Arguments

<code>input</code>	Input data frame or file path to profiles table
<code>min_jury</code>	Minimum number of jurors who must use a taste descriptor for it to be kept. Default: 2.

Details

This function filters sensory data to keep only taste descriptors that appear consistently across at least `min_jury` panelists. It removes descriptors that are used by only one panelist (potentially idiosyncratic terms).

Value

Data frame with consistent profiles and grouped statistics

Examples

```
## Not run:  
data(profiles)  
consistent <- load_consistent_profiles(profiles, min_jury = 2)  
nrow(consistent)  
  
## End(Not run)
```

load_session	<i>Load session</i>
--------------	---------------------

Description

Load and prepare data from a single sensory panel session. Uses base R string operations.

Usage

```
load_session(input_dir, session_info, tab)
```

Arguments

- input_dir Input directory path
- session_info Session metadata (date, cluster, product_name)
- tab Table type to load ('chasselas', 'napping_coord', 'napping_words', 'profiles')

Value

Data frame with loaded and processed session data

Examples

```
NULL
```

napping_coordinates	<i>Napping Session Coordinates</i>
---------------------	------------------------------------

Description

Coordinates assigned to sample fractions during napping sessions, recorded by individual jurors (panelists). Each juror positioned samples on a 2D plane; coordinates represent their sensory perception of similarity.

Usage

```
napping_coordinates
```

Format

A data frame with 64 rows and multiple columns:

- fraction** Fraction identifier (e.g., "fraction_17")
- session** Session identifier (e.g., "session_01")
- J01X, J01Y, J02X, J02Y, J03X, J03Y, J04X, J04Y, J05X, J05Y, J06X, J06Y, J07X, J07Y, J08X, J08Y, J09X, J09Y, J10X, J10Y** Napping coordinates for each juror; column pairs (JxxX, JxxY) represent X and Y positions assigned by juror Jxx

Details

Napping is a projective sensory method where panelists position samples on a 2D plane based on perceived similarity. This produces a sensory map that can be analyzed via correspondence analysis or PCA.

Source

doi:10.1016/j.crfs.2025.101043; Zenodo <https://zenodo.org/records/14616396>

Examples

```
data(napping_coordinates)
head(napping_coordinates)
colnames(napping_coordinates)
```

napping_descriptors	<i>Napping Session Descriptors</i>
---------------------	------------------------------------

Description

Taste descriptors and attributes assigned to sample fractions during napping sessions. Each descriptor was provided by a juror for a specific fraction, with original, intermediate, and harmonized forms.

Usage

```
napping_descriptors
```

Format

- A data frame with 1,146 rows and 6 columns:
- fraction** Fraction identifier (e.g., "fraction_17")
 - session** Session identifier (e.g., "session_01")
 - jury** Juror identifier (e.g., "J1")
 - taste_original** Original descriptor as written by juror
 - taste_intermediate** Intermediate standardization step
 - taste_harmonized** Final harmonized descriptor

Details

Napping descriptors complement napping coordinates. While coordinates represent spatial positioning of similarities, descriptors provide explicit sensory attribute labels that jurors associate with each sample.

Source

doi:10.1016/j.crfs.2025.101043; Zenodo <https://zenodo.org/records/14616396>

Examples

```
data(napping_descriptors)
head(napping_descriptors)
unique(napping_descriptors$taste_harmonized)
```

plot_chasselas_modulation

Plot chasselas modulation

Description

Plot chasselas modulation

Usage

```
plot_chasselas_modulation(
  input = NULL,
  output = "./data/figures/figure_modulation.pdf"
)
```

Arguments

input	Input
output	Output

Examples

```
NULL
```

plot_chasselas_variation

Plot chasselas variation

Description

Plot chasselas variation

Usage

```
plot_chasselas_variation(
  input = NULL,
  output_jury = "./data/figures/figure_variation_jury.pdf",
  output_session = "./data/figures/figure_variation_session.pdf"
)
```

Arguments

input	Input
output_jury	Output jury
output_session	Output session

Examples

NULL

plot_chromatograms	<i>Plot chromatograms</i>
--------------------	---------------------------

Description

Plot chromatograms

Usage

```
plot_chromatograms(  
  input_dir = "./data/20210619",  
  input_groups = NULL,  
  raw_index = 7,  
  fractions_indices = 12:65,  
  xlim = c(2, 25),  
  output = "./data/figures/figure_chromatograms.pdf"  
)
```

Arguments

input_dir	Input dir
input_groups	Input groups
raw_index	Raw index
fractions_indices	Fractions indices
xlim	Xlim
output	Output

Examples

NULL

plot_concentration	<i>Plot concentration</i>
--------------------	---------------------------

Description

Plot concentration

Usage

```
plot_concentration(  
  input = NULL,  
  output = "../data/figures/figure_raw_extract.pdf"  
)
```

Arguments

input	Input
output	Output

Examples

```
NULL
```

plot_correlations	<i>Plot correlations</i>
-------------------	--------------------------

Description

Plot correlations

Usage

```
plot_correlations(  
  input_correlations = "../data/correlations.tsv",  
  input_ions = "../data/fractions_mzmine/fractions.csv",  
  input_tastes = NULL,  
  output_1 = "../data/figures/figure_correlations_1.pdf",  
  output_2 = "../data/figures/figure_correlations_2.pdf",  
  output_3 = "../data/figures/figure_correlations_3.pdf",  
  annotation_path_fractions =  
    "../data/processed/241217_130815_fractions/fractions_results.tsv",  
  imputation_factor = 0.5,  
  max_pval = 0.05,  
  min_area_ion = 1L,  
  min_confidence = 0.45,
```

```

    min_consistency = 0.9,
    min_corr = 0.95,
    min_intense_ions_ratio = 0.8,
    min_jury = 2L,
    mode = "pos",
    widths = 5:9
)

```

Arguments

<code>input_correlations</code>	Input correlations
<code>input_ions</code>	Input ions
<code>input_tastes</code>	Input tastes
<code>output_1</code>	Output 1
<code>output_2</code>	Output 2
<code>output_3</code>	Output 3
<code>annotation_path_fractions</code>	Annotation path fractions
<code>imputation_factor</code>	Imputation factor
<code>max_pval</code>	Max p value
<code>min_area_ion</code>	Minimal area ion
<code>min_confidence</code>	Minimal confidence
<code>min_consistency</code>	Minimal consistency
<code>min_corr</code>	Minimal correlation
<code>min_intense_ions_ratio</code>	Minimal intense ions ratio
<code>min_jury</code>	Minimal number of jury
<code>mode</code>	Mode
<code>widths</code>	Widths (of the taste-ion windows)

Examples

```
NULL
```

plot_descriptors_network	<i>Plot descriptors network</i>
--------------------------	---------------------------------

Description

Plot descriptors network

Usage

```
plot_descriptors_network(  
  input = NULL,  
  output = "../data/figures/figure_network.pdf"  
)
```

Arguments

input	Input
output	Output

Examples

```
NULL
```

plot_matrices	<i>Plot matrices</i>
---------------	----------------------

Description

Plot matrices

Usage

```
plot_matrices(input = NULL, output = "../data/figures/figure_matrices.pdf")
```

Arguments

input	Input
output	Output

Examples

```
NULL
```

plot_napping	<i>Plot napping</i>
Description Plot napping	
Usage <pre>plot_napping(input_coordinates = system.file("extdata", "napping_coordinates.tsv", package = "sapid"), input_descriptors = system.file("extdata", "napping_descriptors.tsv", package = "sapid"), sessions = 2)</pre>	
Arguments input_coordinates Input coordinates input_descriptors Input descriptors sessions Sessions	
Examples NULL	
plot_profiles	<i>Plot profiles</i>

Description

Plot profiles

Usage

```
plot_profiles(  
  input = NULL,  
  output = "../data/figures/figure_profiles.pdf",  
  annotation_path_extract =  
    "~/Git/sapid/data/processed/241026_103144_extract/extract_results.tsv",  
  annotation_path_fractions =  
    "~/Git/sapid/data/processed/241217_130815_fractions/fractions_results.tsv",
```

```
features_path_extract = "~/Git/sapid/data/extract_mzmine/extract.csv",
features_path_fractions = "~/Git/sapid/data/fractions_mzmine/fractions.csv",
peaks_dir_extract = "~/Git/sapid/data/interim/peaks/extract",
peaks_dir_fractions = "~/Git/sapid/data/interim/peaks/fractions",
min_confidence = 0.45,
min_similarity_prefilter = 0.6,
min_similarity_filter = 0.8,
mode = "pos",
detector = "ms",
type = "analysis"
)
```

Arguments

input	Input
output	Output
annotation_path_extract	Annotation path extract
annotation_path_fractions	Annotation path fractions
features_path_extract	Features path extract
features_path_fractions	Annotation path fractions
peaks_dir_extract	Peaks dir extract
peaks_dir_fractions	Peaks dir fractions
min_confidence	Minimal confidence
min_similarity_prefilter	Minimal similarity prefilter
min_similarity_filter	Minimal similarity filter
mode	Mode
detector	Detector
type	Type

Examples

NULL

prepare_chasselas	<i>Prepare chasselas</i>
-------------------	--------------------------

Description

Prepare chasselas

Usage

```
prepare_chasselas(  
  input_dir = "~/switchdrive/SAPERRE/02_raw-data/inhouse/02_sensory",  
  sessions = seq(1, 7),  
  output = "inst/extdata/chasselas.tsv"  
)
```

Arguments

input_dir	Input dir
sessions	Sessions
output	output

Examples

NULL

prepare_chemical_profiles	<i>Prepare chemical profiles</i>
---------------------------	----------------------------------

Description

Prepare chemical profiles

Usage

```
prepare_chemical_profiles(  
  mzmls_dir = "./data/20210619",  
  peaks_dir_extract = "./data/interim/peaks/extract",  
  peaks_dir_fractions = "./data/interim/peaks/fractions",  
  features_path_extract = "./data/extract_mzmine/extract.csv",  
  features_path_fractions = "./data/fractions_mzmine/fractions.csv",  
  min_intensity = 1e+05,  
  start_extract = 5L,  
  end_extract = 7L,  
  start_fractions = 12L,  
  end_fractions = 65L  
)
```

Arguments

mzmls_dir	mzML files directory
peaks_dir_extract	Peaks dir extract
peaks_dir_fractions	Peaks dir fractions
features_path_extract	Features path extract
features_path_fractions	Features path fractions
min_intensity	Minimal intensity
start_extract	Start extract
end_extract	End extract
start_fractions	Start fractions
end_fractions	End fractions

Examples

NULL

prepare_concentration *Prepare concentration*

Description

Prepare concentration

Usage

```
prepare_concentration(  
  input_xlsx = paste0("~/switchdrive/SAPERE/02_raw-data/inhouse/02_sensory/",  
    "20210329_raw-extract/03_files/20210329_raw-extract.xlsx"),  
  output = "inst/extdata/concentration_afc.tsv"  
)
```

Arguments

input_xlsx	Input xlsx
output	Output

Examples

NULL

prepare_napping	<i>Prepare napping</i>
-----------------	------------------------

Description

Prepare napping

Usage

```
prepare_napping(  
  input_dir = "~/switchdrive/SAPERRE/02_raw-data/inhouse/02_sensory",  
  dictionary_generic_path = system.file("extdata", "dictionary_generic.tsv", package =  
    "sapid"),  
  dictionary_napping_path = system.file("extdata", "dictionary_napping.tsv", package =  
    "sapid"),  
  dictionary_specific_path = system.file("extdata", "dictionary_specific.tsv", package =  
    "sapid"),  
  sessions = seq(1, 8),  
  output_coordinates = "inst/extdata/napping_coordinates.tsv",  
  output_descriptors = "inst/extdata/napping_descriptors.tsv"  
)
```

Arguments

input_dir	Input directory
dictionary_generic_path	Dictionary generic path
dictionary_napping_path	Dictionary napping path
dictionary_specific_path	Dictionary specific path
sessions	Sessions
output_coordinates	Output coordinates
output_descriptors	Output descriptors

Examples

NULL

prepare_profiles	<i>Prepare profiles</i>
------------------	-------------------------

Description

Prepare profiles

Usage

```
prepare_profiles(  
  input_dir = "~/switchdrive/SAPERRE/02_raw-data/inhouse/02_sensory",  
  dictionary_generic_path = system.file("extdata", "dictionary_generic.tsv", package =  
    "sapid"),  
  dictionary_specific_path = system.file("extdata", "dictionary_specific.tsv", package =  
    "sapid"),  
  sessions = seq(1, 7),  
  output = "inst/extdata/profiles.tsv"  
)
```

Arguments

input_dir	Input dir
dictionary_generic_path	Dictionary generic path
dictionary_specific_path	Dictionary specific path
sessions	Sessions
output	output

Examples

NULL

profiles	<i>Sensory Profiles with Harmonized Taste Descriptors</i>
----------	---

Description

The primary sensory panel dataset containing taste profile ratings for Swertia chirayita sample fractions. Includes original (as entered by panelists) and harmonized (standardized) taste descriptor terms. This is the main table for sensory-MS correlation analyses.

Usage

profiles

Format

A data frame with 1,665 rows and 6 columns:

fraction Fraction identifier (e.g., "fraction_17")

session Session identifier (e.g., "session_01")

jury Juror identifier (e.g., "jury_03")

taste_original Original taste descriptor as entered by panelist

taste_harmonized Standardized/harmonized taste descriptor

value Numerical rating or intensity score

Details

Missing or empty harmonized values indicate descriptors that could not be consistently mapped across panelists or were deemed non-informative (e.g., "RIEN" meaning "nothing"). Use `!is.na(taste_harmonized) & taste_harmonized != ""` to filter for valid descriptors in analyses.

Source

doi:10.1016/j.crfs.2025.101043; Zenodo <https://zenodo.org/records/14616396>

Examples

```
data(profiles)
head(profiles)
table(profiles$taste_harmonized)[table(profiles$taste_harmonized) > 10]
```

sapid-data

Sapid Package Datasets

Description

This page documents the built-in datasets in the `sapid` package, which provide sensory panel data, sample metadata, and reference dictionaries for taste-characterization workflows. All datasets originate from or are derived from the *Swertia chirayita* case study published in [doi:10.1016/j.crfs.2025.101043](https://doi.org/10.1016/j.crfs.2025.101043), with raw data archived at <https://zenodo.org/records/14616396>.

`sigmoid`

*sigmoid***Description**

Creates a longer dataframe with coordinates for a smoothed line.

Usage

```
sigmoid(x_from, x_to, y_from, y_to, smooth = 5, n = 100, direction = "x")
```

Arguments

<code>x_from</code>	start x value
<code>x_to</code>	end x value
<code>y_from</code>	start y value
<code>y_to</code>	end y values
<code>smooth</code>	smooth parameter. Higher means less smoothing
<code>n</code>	number of point that should be smoothed
<code>direction</code>	the character x or y depending on direction of smoothing

Value

a data frame

Index

* datasets

- chasselas, [3](#)
- concentration_afc, [5](#)
- dictionary_generic, [7](#)
- dictionary_napping, [7](#)
- dictionary_specific, [8](#)
- fractions, [9](#)
- groups, [11](#)
- napping_coordinates, [15](#)
- napping_descriptors, [16](#)
- profiles, [27](#)

chasselas, [3](#)

cluster_nmr, [4](#)

concentration_afc, [5](#)

correlate_ion_taste_intensities, [6](#)

dictionary_generic, [7](#)

dictionary_napping, [7](#)

dictionary_specific, [8](#)

fractions, [9](#)

geom_sigmoid, [10](#)

get_session_info, [11](#)

groups, [11](#)

harmonize_terms, [12](#)

harmonize_terms_df, [13](#)

load_consistent_profiles, [14](#)

load_session, [15](#)

napping_coordinates, [15](#)

napping_descriptors, [16](#)

plot_chasselas_modulation, [17](#)

plot_chasselas_variation, [17](#)

plot_chromatograms, [18](#)

plot_concentration, [19](#)

plot_correlations, [19](#)

plot_descriptors_network, [21](#)

plot_matrices, [21](#)

plot_napping, [22](#)

plot_profiles, [22](#)

prepare_chasselas, [24](#)

prepare_chemical_profiles, [24](#)

prepare_concentration, [25](#)

prepare_napping, [26](#)

prepare_profiles, [27](#)

profiles, [27](#)

sapid-data, [28](#)

sigmoid, [29](#)